SIMAX, VOL. 16, PP. 388–413, NO. 2, APR. 1995.

A LOOK-AHEAD BLOCK SCHUR ALGORITHM FOR TOEPLITZ-LIKE MATRICES *

ALI H. SAYED† AND THOMAS KAILATH‡

Abstract. We derive a look-ahead recursive algorithm for the block triangular factorization of Toeplitz-like matrices. The derivation is based on combining the block Schur/Gauss reduction procedure with displacement structure and leads to an efficient block-Schur complementation algorithm. For an $n \times n$ Toeplitz-like matrix, the overall computational complexity of the algorithm is $O(rn^2 + \frac{n^3}{t})$ operations, where r is the matrix displacement rank and t is the number of diagonal blocks. These blocks can be of any desirable sizes. They may, for example, correspond to the smallest nonsingular leading submatrices or, alternatively, to numerically well-conditioned blocks.

Key words. Toeplitz-like matrices, block Schur algorithm, block triangular factorization, linear equations, singular minors, look-ahead algorithm.

AMS subject classifications. 65F05, 65F30, 15A23, 15A06

1. Introduction.

The triangular factorization of a matrix is a useful tool for many problems. Such a factorization is guaranteed to exist whenever the matrices are strongly regular, *i.e.,* all leading principal minors are nonzero [11]. The standard Gaussian elimination technique (also known as Schur reduction) may then be used to compute the triangular factors of the matrix. Also, in many applications, one is often faced with matrices that exhibit some structure, *e.g.,* Toeplitz, Hankel, close-to-Toeplitz, close-to-Hankel, and related matrices. Such structure is nicely captured by introducing the concept of displacement structure [18, 20]. In this context, an $n \times n$ structured matrix R is characterized by an $n \times r$ matrix G (called a generator of R) with $r \ll n$ usually. The minimum column dimension of G is called the displacement rank of R. The triangular factorization of such strongly regular R can be computed efficiently and recursively in $O(rn^2)$ operations (additions and multiplications) [19, 23, 32]. This is achieved by appropriately combining Gaussian elimination with displacement structure. The resulting algorithm can then be regarded as a far reaching generalization of an algorithm of Schur [1, 36], which was chiefly concerned with the apparently very different problem of checking whether a power series is analytic and bounded in the unit disc; hence the name generalized Schur algorithm. The reader may consult the recent survey paper [21] for detailed discussions on the topic of displacement structure.

Now most fast factorization algorithms that have been derived so far in the literature assume that the involved structured matrices are strongly regular. In several instances, however, it might be more appropriate to perform *block* Schur complementation steps. This happens, for example, when the assumption of strong regularity is dropped, which then requires the use of the smallest nonsingular leading minor, or

* This work was supported in part by the Air Force Office of Scientific Research, Air Force Systems Command under Contract AFOSR91-0060 and by the Army Research Office under contract DAAL03-89-K-0109.

† This work was performed while Dr. Sayed was a Research Associate in the Information Systems Laboratory, Stanford University, and on leave from Escola Politécnica da Universidade de São Paulo, Brazil. His work was also supported by a fellowship from FAPESP, Brazil. He is an Assistant Professor in the Dept. of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106-9560.

‡Dr. Kailath is with the Information Systems Laboratory, Stanford University, CA 94305.

a numerically well-conditioned leading minor of appropriate dimensions, in order to proceed with a block Schur reduction step.

Indeed, many authors have worked on the problem of extending the fast algorithms to nonstrongly regular matrices, where the sizes of the block Schur complementation steps were determined by the sizes of the smallest nonsingular leading minors. Among these we may mention the works of Heinig and Rost [15], Delsarte, Genin, and Kamp [9], and Gover and Barnett [13] who generalized the classical Levinson algorithm for solving Toeplitz systems of linear equations (or equivalently factoring the inverse of the Toeplitz coefficient matrix); a so-called split-Levinson algorithm was later considered by Ciliz and Krishna [7]. Pombra, Lev-Ari, and Kailath [28] also derived both Levinson and Schur type algorithms for Toeplitz matrices by generalizing the three-term recursion for polynomials orthogonal on the unit circle. The case of nonstrongly regular Hankel matrices arises in many applications as well, such as the partial realization problem and decoding of BCH codes [4, 8]. Algorithms for computing the triangular factorization and/or inversion of arbitrary Hankel matrices have been derived by Berlekamp [3], Massey [25], Kung [22], Citron [8], etc.. More recently, Zarowski [37] used the algorithms of Heinig and Rost [15] and Delsarte et al. [9] to induce Schur type algorithms for Hermitian Toeplitz and Hankel matrices with singular minors.

All these algorithms are applicable to Toeplitz and Hankel matrices only. Recently, Pal and Kailath [26, 27] derived recursive algorithms that are applicable to a larger class of matrices called quasi-Toeplitz and quasi-Hankel. These are congruent to Toeplitz and Hankel matrices in a certain sense. The derivation exploits this fact and, among other results, shows that the determination of the size of the smallest nonsingular minor is reduced to counting the number of repeated zeros at the origin of a certain polynomial.

But Toeplitz, Hankel, quasi-Toeplitz, and quasi-Hankel matrices are all structured matrices with displacement rank $r = 2$. In many applications however, such as system identification, image processing, and multichannel filtering, block structured matrices arise that have displacement ranks larger than 2. In these cases, the previous algorithms are not applicable. Moreover, in the above varied approaches, the sizes of the block Schur complements were set equal to the sizes of the smallest nonsingular minors, which thus requires the verification of the occurrence of exact singularities. This may pose considerable difficulties from a numerical point of view.

Alternatively, one can determine the sizes of the block steps by looking for numerically well-conditioned blocks. This has recently been studied by several authors trying to devise effective numerical algorithms for general Toeplitz systems of equations. An early paper was the one of Chan and Hansen [6]. Among many others we mention Gutknecht [14] and Freund [10], which give extensive references.

In this paper, we provide a new fast look-ahead (block-Schur) algorithm for matrices with very general displacement structure, which include the Toeplitz case as a special instance. We study arbitrary Hermitian Toeplitz-like matrices and derive an algorithm that leads to a factorization of the form $R = LDL^*$, where L is a lower triangular matrix and D is a block diagonal matrix whose block entries are easily invertible. The overall computational complexity of the algorithm is $O(rn^2 + n^3/t)$ elementary operations (addition and multiplication), where t is the number of diagonal blocks in D. In the strongly regular case we have $t = n$ and the complexity reduces to the usual $O(rn^2)$ figure. The diagonal blocks in D can be of any desirable sizes. They can be chosen, for example, as the smallest nonsingular minors, or as the sizes of numerically well-conditioned blocks, etc.. For this reason, our development will consist of two independent steps. We shall first derive the block Schur algorithm assuming arbitrary choices for the sizes of the blocks, thus leading to a general-purpose fast Schur complementation procedure that does not depend on the specific choices for the sizes of these blocks. We shall then focus in Section 5 on the particular choices that correspond to the smallest (exactly) nonsingular leading blocks. This is done here because, apart from numerical possibilities, the fast block-Schur complementation algorithm also has several theoretically interesting features as well. For example, the explicit formulas for the block diagonal matrix in the block triangular factorization can give simple rules for computing the inertia of general structured matrices, with important applications in root distribution of polynomials.

The paper is organized as follows. In Section 2 we review the class of structured matrices and describe the Schur/Gauss reduction procedure for the triangular factorization of strongly regular matrices. In Section 3 we combine the Schur reduction procedure with displacement structure and derive the generalized block Schur algorithm. In Section 4 we separately consider the special cases of strongly regular and block steps along with the corresponding computational complexities. In Section 5 we address the issue of determining the sizes of the smallest (exact) nonsingular minors. In Section 6 we show how to exploit the matrix structure in order to efficiently compute the QR factors of the blocks of D. In Section 7 we give a system (and state-space) interpretation of the derived recursions and we conclude with Section 8.

2. Structured Matrices. The concept of displacement structure and structured matrices can be briefly motivated by considering the much-studied special case of a Hermitian Toeplitz matrix, $T = [c_{i-j}]_{i,j=0}^{n-1}, c_k = c_{-k}^*$. Since T depends only on n parameters rather than n^2, it may not be surprising that matrix problems involving T (such as triangular factorization, orthogonalization, inversion) have complexity $O(n^2)$ rather than $O(n^3)$. But what about the complexity of such problems for inverses, products, and related combinations of Toeplitz matrices such as $T^{-1}, T_1T_2, T_1 - T_2T_3^{-1}T_4, (T_1T_2)^{-1}T_3, \ldots$? Though these are not Toeplitz, they are certainly structured and the complexity of inversion and factorization are not expected to be very different from that for a pure Toeplitz matrix, T. It turns out that the appropriate common property of all these matrices is not their "Toeplitzness", but the fact that they all have low *displacement rank.* The displacement of an $n \times n$ Hermitian matrix R was originally defined by Kailath, Kung, and Morf [20] as

$$\nabla R \equiv R - ZRZ^* \quad , \tag{2.1}$$

where the symbol $*$ stands for Hermitian conjugate transpose of a matrix (complex conjugation for scalars), and Z is the $n \times n$ lower shift matrix with ones on the first subdiagonal and zeros elsewhere; ZRZ^* corresponds to shifting R downwards along the main diagonal by one position, explaining the name *displacement* for ∇R. If ∇R has low rank, say r, independent of n, then R is said to be *structured* with respect to the displacement defined by (2.1), and r is referred to as the displacement rank of R. In this case, we can (nonuniquely) factor ∇R as $\nabla R = GJG^*$, where $J = J^*$ is a signature matrix that specifies the *displacement inertia* of R: it has as many $\pm 1's$ on the diagonal as ∇R has positive and negative eigenvalues, $J = (I_p \oplus -I_q)$, $p + q = r$, and G is an $n \times r$ matrix. Here, I_p denotes the $p \times p$ identity matrix. The pair $\{G, J\}$ is called a *generator* of R. For a Hermitian Toeplitz matrix $T = [c_{i-j}]_{i,j=0}^{n-1}, c_k = c_{-k}^*$, with $c_0 = 1$, it is straightforward to verify that (2.1) leads to a compact description of T. Indeed, if we subtract ZTZ^* from T we get

$$T - ZTZ^* = \begin{bmatrix} 1 & 0 \\ c_1 & c_1 \\ \vdots & \vdots \\ c_{n-1} & c_{n-1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ c_1 & c_1 \\ \vdots & \vdots \\ c_{n-1} & c_{n-1} \end{bmatrix}^* , \tag{2.2}$$

which shows that $T - ZTZ^*$ has rank 2, or equivalently, T has displacement rank 2, independent of n.

To motivate more general structures, and to clarify the importance of direct factorization problems as opposed to inversion problems, we consider a simple example that shows the need for more general structures such as $R - FRF^*$, with lower triangular F.

Consider again the case of an $n \times n$ Hermitian Toeplitz matrix T for which $T - Z_n T Z_n^*$ has rank 2 (Z_n now denotes the $n \times n$ lower shift matrix), and assume we are interested in factoring T^{-1}. If we form the block matrix (see [19] for more examples and discussion)

$$M = \begin{bmatrix} -T & I \\ I & \mathbf{0} \end{bmatrix} ,$$

it is then straightforward to check that the displacement rank of M with respect to $M - Z_{2n} M Z_{2n}^*$ is equal to 4. However, we can get a lower displacement rank by using a different definition, viz.,

$$\nabla M = M - \begin{bmatrix} Z_n & \mathbf{0} \\ \mathbf{0} & Z_n \end{bmatrix} M \begin{bmatrix} Z_n & \mathbf{0} \\ \mathbf{0} & Z_n \end{bmatrix}^* ,$$

which corresponds to choosing $F = Z_n \oplus Z_n$ in the definition $R - FRF^*$ (rather than $F = Z_{2n}$, the $2n \times 2n$ lower shift matrix).

The question is then how to exploit the structure of M in order to obtain fast factorization of T^{-1}. The answer is that the (generalized) Schur algorithm operates as follows: it starts with a generator matrix G of a structured matrix (say the generator of M), and it recursively computes generator matrices of the successive Schur complements of the matrix. So the first step of the algorithm gives us G_1, which is a generator of the Schur complement of M with respect to its $(0,0)$ entry. The next step gives us G_2, which is a generator of the Schur complement of M with respect to its 2×2 leading submatrix, and so on. After n such steps, we obtain a generator of the n^{th} Schur complement, which is T^{-1}. This procedure can be shown to provide the triangular factorization of T^{-1} (see, e.g., [19, 21]).

Hence, by performing the *direct* factorization of the extended matrix M we also obtain the factors of the inverse matrix T^{-1}; this is an alternative to the use of the Levinson algorithm for this problem. Applications with more general matrices F (such as diagonal or in Jordan form) include interpolation problems [5, 33, 34] and adaptive filtering [35].

In this paper we study $n \times n$ Hermitian matrices R with Toeplitz-like displacement structure of the form

$$R - FRF^* = GJG^* , \tag{2.3}$$

where F is an $n \times n$ lower triangular matrix with diagonal elements $\{f_0, f_1, \ldots, f_{n-1}\}$, G is an $n \times r$ so-called generator matrix (with $r \leq n$), and J is an $r \times r$ Hermitian signature matrix satisfying $J^2 = I$, such as $J = (I_p \oplus -I_q)$, $p+q=r$, or some other convenient form (as will be the case in Section 5.5). We further assume that R is invertible but *not necessarily strongly regular*, and that $1 - f_i f_j^* \neq 0$ for every i, j. The latter condition guarantees the existence of a unique solution R of (2.3) (but it can be relaxed as discussed in [21]). We say that R has a Toeplitz-like structure with respect to F and $\{G, J\}$ is called a generator pair of R.

2.1. The Block Schur/Gauss Reduction Procedure. The Gaussian elimination (or Schur reduction) procedure is a recursive algorithm that computes the triangular factors of a matrix. To clarify this, consider a Hermitian and invertible (but not necessarily strongly regular) matrix R, and let η_0 denote the desired size of the leading (invertible) block, D_0, with respect to which a Schur complementation step is to be performed. The η_0 may stand for the size of the smallest nonsingular minor of R or, alternatively, for the size of a numerically well-conditioned block (as in [6, 10], for example), or for some other convenient choice. If L_0 represents the first η_0 columns of R then

$$R - L_0 D_0^{-1} L_0^* = \begin{bmatrix} \mathbf{0}_{\eta_0 \times \eta_0} & \mathbf{0} \\ \mathbf{0} & R_1 \end{bmatrix} \equiv \tilde{R}_1 \ ,$$

where R_1 is an $(n-\eta_0) \times (n-\eta_0)$ matrix that is called the Schur complement of D_0 in R. Also, L_0 is an $n \times \eta_0$ matrix whose leading $\eta_0 \times \eta_0$ block is equal to D_0. We shall say that $\tilde{R}_1$ has one (block) zero row and one (block) zero column (the size of the block being η_0). If we further let η_1 denote the desired size of the leading (invertible) block of R_1 (denoted by D_1) and consider the corresponding first η_1 columns of R_1 (denoted by L_1), then we also have

$$R_1 - L_1 D_1^{-1} L_1^* = \begin{bmatrix} \mathbf{0}_{\eta_1 \times \eta_1} & \mathbf{0} \\ \mathbf{0} & R_2 \end{bmatrix} \equiv \tilde{R}_2 \ ,$$

where R_2 is now an $(n-\eta_0-\eta_1) \times (n-\eta_0-\eta_1)$ matrix that is the Schur complement of D_1 in R_1. Repeating this recursive procedure, viz,

$$\begin{bmatrix} \mathbf{0}_{\eta_i \times \eta_i} & \mathbf{0} \\ \mathbf{0} & R_{i+1} \end{bmatrix} = R_i - L_i D_i^{-1} L_i^* \ , \quad i \geq 0 \ , \tag{2.4}$$

we clearly get, say after t steps,

$$R = L D_B^{-1} L^* =$$

$$L_0 D_0^{-1} L_0^* + \begin{bmatrix} \mathbf{0}_{\eta_0 \times \eta_1} \\ L_1 \end{bmatrix} D_1^{-1} \begin{bmatrix} \mathbf{0}_{\eta_0 \times \eta_1} \\ L_1 \end{bmatrix}^* + \begin{bmatrix} \mathbf{0}_{\eta_0 \times \eta_2} \\ \mathbf{0}_{\eta_1 \times \eta_2} \\ L_2 \end{bmatrix} D_2^{-1} \begin{bmatrix} \mathbf{0}_{\eta_0 \times \eta_2} \\ \mathbf{0}_{\eta_1 \times \eta_2} \\ L_2 \end{bmatrix}^* + \ldots \ ,$$

where $D_B = (D_0 \oplus D_1 \oplus \ldots \oplus D_{t-1})$ is *block* diagonal, and the (nonzero parts of the) columns of the *block* lower triangular matrix L are $\{L_0, \ldots, L_{t-1}\}$. Here t is the number of reduction steps, *i.e.*, $n = \sum_{i=0}^{t-1} \eta_i$. We also define, for later reference, $\alpha_j = \sum_{i=0}^{j-1} \eta_i, \alpha_0 = 0$. The computational complexity of the above procedure is $O(n^3)$ elementary operations and it leads to a block triangular factorization of R.

It is clear at this point that the following are among the major issues and/or questions that arise during the block triangular factorization procedure: (i) How to efficiently exploit any Toeplitz-like structure of R; (ii) How to efficiently compute the triangular factors L_i and D_i; (iii) How to compute (or avoid) the inversion of the diagonal blocks D_i; (iv) How to determine an alternative triangular factorization of the form $R = \hat{L}\hat{D}_B^{-1}\hat{L}^*$, with $\hat{L}$ lower (*not block*) triangular and with a block-diagonal matrix $\hat{D}_B$ whose block entries are easily invertible; (v) How to determine the sizes of the block steps, η_i.

We address the first four questions in the next two sections and postpone the discussion of the last question to Section 5, where we focus on a particular choice for the η_i that is determined by the sizes of the smallest nonsingular minors of R. It will be clear from the derivation that follows that, in order to increase the computational efficiency of the resulting algorithm, these questions should be answered by essentially restricting ourselves to the use of the entries of the generator matrix of R, without the need to explicitly form its successive block Schur complements, R_i.

3. Block Schur Algorithm for Toeplitz-Like Matrices. We now exploit the fact that R is a structured (Toeplitz-like) matrix. That is, we show that the successive computation of the Schur complements of R in (2.4) can be carried out in a computationally efficient recursive procedure by exploiting the structure implied by (2.3). To begin with, we define F_i to be the submatrix obtained by ignoring the first α_i columns and rows (or the first i block columns and rows) of F (recall that $\alpha_i = \eta_0 + \ldots + \eta_{i-1}$). This means that F_{i+1} is a submatrix of F_i, viz.,

$$F_i = \begin{bmatrix} \hat{F}_i & \mathbf{0} \\ ? & F_{i+1} \end{bmatrix}, \quad F_0 = F \ ,$$

where ? denotes irrelevant entries and $\hat{F}_i$ is the $\eta_i \times \eta_i$ leading submatrix of F_i. In other words, F_{i+1} is obtained by deleting the first η_i rows and columns of F_i. The following theorem, first stated in general terms, shows that the successive Schur complements of a Toeplitz-like matrix inherit its structure and thus satisfy a displacement equation similar to (2.3).

THEOREM 3.1. *The i^{th} Schur complement R_i of a Toeplitz-like matrix R, as in (2.3) and (2.4), is also Toeplitz-like with respect to F_i, viz., R_i satisfies a displacement equation of the form $R_i - F_i R_i F_i^* = G_i J G_i^*$, where the generator matrix G_i satisfies the following recursive construction: start with $G_0 = G, F_0 = F$, and repeat for $i = 0, 1, \ldots, t-1$:*

1. *At step i we have F_i and G_i. Let $\hat{G}_i$ denote the top η_i rows of G_i.*
2. *The i^{th} triangular factors L_i and D_i are the solutions of the equations*

$$D_i = \hat{F}_i D_i \hat{F}_i^* + \hat{G}_i J \hat{G}_i^* \ , \quad L_i = F_i L_i \hat{F}_i^* + G_i J \hat{G}_i^*. \tag{3.1a}$$

3. *Choose arbitrary $r \times \eta_i$ and $r \times r$ matrices $\hat{H}_i$ and $\hat{K}_i$, respectively, so as to satisfy the embedding relation*

$$\begin{bmatrix} \hat{F}_i & \hat{G}_i \\ \hat{H}_i & \hat{K}_i \end{bmatrix} \begin{bmatrix} D_i & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} \begin{bmatrix} \hat{F}_i & \hat{G}_i \\ \hat{H}_i & \hat{K}_i \end{bmatrix}^* = \begin{bmatrix} D_i & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix}. \tag{3.1b}$$

4. *A generator for R_{i+1} is then given by*

$$\begin{bmatrix} \mathbf{0}_{\eta_i \times r} \\ G_{i+1} \end{bmatrix} = F_i L_i \hat{H}_i^* J + G_i J \hat{K}_i^* J. \tag{3.1c}$$

Proof. We prove the result for G_1. The same argument applies to $\{G_i,\ i > 1\}$. It follows from (2.3) that the leading submatrix D_0 and the corresponding η_0 columns L_0 are solutions of the equations: $D_0 = \hat{F}_0 D_0 \hat{F}_0^* + \hat{G}_0 J \hat{G}_0^*$ and $L_0 = F L_0 \hat{F}_0^* + G J \hat{G}_0^*$. Substituting these expressions into the definition of $\tilde{R}_1$ in (2.4) and computing the difference $\tilde{R}_1 - F\tilde{R}_1 F^*$ we get

$$\begin{aligned}\tilde{R}_1 - F\tilde{R}_1 F^* = GJ\left\{J - \hat{G}_0^* D_0^{-1} \hat{G}_0\right\} JG^* + \\ FL_0\left[D_0^{-1} - \hat{F}_0^* D_0^{-1}\hat{F}_0\right] L_0^* F^* - \\ FL_0 \hat{F}_0^* D_0^{-1} \hat{G}_0 J G^* - GJ\hat{G}_0^* D_0^{-1} \hat{F}_0 L_0^* F^* \ .\end{aligned} \tag{3.2}$$

We now verify that the right-hand side of the above expression can be put into the form of a *perfect square* by introducing some auxiliary quantities. Consider an $r \times \eta_0$ matrix $\hat{H}_0$ and an $r \times r$ matrix $\hat{K}_0$ that are defined to satisfy the following relations (in terms of the quantities that appear on the right-hand side of the above expression. We shall see very soon that this is always possible.):

$$\hat{H}_0^* J \hat{H}_0 = D_0^{-1} - \hat{F}_0^* D_0^{-1} \hat{F}_0, \quad \hat{K}_0^* J \hat{K}_0 = J - \hat{G}_0^* D_0^{-1} \hat{G}_0, \quad \hat{K}_0^* J \hat{H}_0 = -\hat{G}_0^* D_0^{-1} \hat{F}_0.$$

Using $(\hat{H}_0, \hat{K}_0)$ we can factor the right-hand side of (3.2) as $\tilde{G}_1 J \tilde{G}_1^*$, where $\tilde{G}_1 = FL_0\hat{H}_0^* J + GJ\hat{K}_0^* J$. But the first block row and block column of $\tilde{R}_1$ are zero. Hence, $\tilde{G}_1$ is of the form $\tilde{G}_1 = \begin{bmatrix} \mathbf{0}_{r\times\eta_0} & G_1^{\mathbf{T}} \end{bmatrix}^{\mathbf{T}}$. Moreover, it follows from the above expressions for $(\hat{H}_0, \hat{K}_0)$ that $\hat{F}_0$, $\hat{G}_0$, $\hat{H}_0$, and $\hat{K}_0$ satisfy the relation

$$\begin{bmatrix} \hat{F}_0 & \hat{G}_0 \\ \hat{H}_0 & \hat{K}_0 \end{bmatrix}^* \begin{bmatrix} D_0^{-1} & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} \begin{bmatrix} \hat{F}_0 & \hat{G}_0 \\ \hat{H}_0 & \hat{K}_0 \end{bmatrix} = \begin{bmatrix} D_0^{-1} & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} \ ,$$

which is equivalent to (3.1b) for $i = 0$. □

We still need to show how to choose matrices $(\hat{H}_i, \hat{K}_i)$ so as to satisfy the embedding relation (3.1b). Following an argument similar to that in [24] we get the following result.

LEMMA 3.2. *All choices of $\hat{H}_i$ and $\hat{K}_i$ that satisfy (3.1b) can be expressed in terms of $\hat{F}_i$, $\hat{G}_i$, and D_i as follows :*

$$\begin{aligned}\hat{H}_i &= \Theta_i^{-1} J \hat{G}_i^* \left[I_{\eta_i} - \tau_i \hat{F}_i^*\right]^{-1} D_i^{-1} (\tau_i I_{\eta_i} - \hat{F}_i) \ , \\ \hat{K}_i &= \Theta_i^{-1} \left\{I_r - J\hat{G}_i^* \left[I_{\eta_i} - \tau_i \hat{F}_i^*\right]^{-1} D_i^{-1} \hat{G}_i\right\} \ ,\end{aligned} \tag{3.3}$$

where Θ_i is an arbitrary $J-$unitary matrix ($\Theta_i J \Theta_i^* = J$) and τ_i is an arbitrary unit-modulus scalar ($|\tau_i| = 1$).

Substituting expression (3.3) for $\hat{H}_i$ and $\hat{K}_i$ into the generator recursion (3.1c) we obtain the following algorithm, which we shall refer to as the *generalized block Schur algorithm.* This algorithm allows us to compute generator matrices for the successive (block) Schur complements of R, viz., $G \to G_1 \to G_2 \to \ldots$, which can then be used to solve for the triangular factors via (3.1a).

ALGORITHM 3.3 (Block Schur Algorithm). *The generators G_i of the successive Schur complements R_i satisfy the recursion*

$$\begin{bmatrix} \mathbf{0}_{\eta_i \times r} \\ G_{i+1} \end{bmatrix} = \left\{G_i + (\tau_i^* F_i - I_{n-\alpha_i}) L_i D_i^{-1} (I_{\eta_i} - \tau_i^* \hat{F}_i)^{-1} \hat{G}_i\right\} \Theta_i \ , \tag{3.4}$$

where Θ_i is an arbitrary $J-$unitary matrix and τ_i is an arbitrary unit-modulus scalar. The i^{th} triangular factors L_i and D_i are found by solving (3.1a).

4. Computational Issues and Simplifications. The point to stress here is that we have so far shown the following: the triangular factors L_i and D_i can be found by solving equations (3.1a), which are completely specified in terms of F_i and G_i and without the need to explicitly form R_i, since the $G_i's$ can be recursively computed via (3.4). To further stress this point we now take a closer look at recursion (3.4) and equation (3.1a).

4.1. Strongly Regular Steps. We first consider the special case that corresponds to $\eta_i = 1$, and which we shall refer to as a *strongly regular* step. In this case, it is possible to further simplify the generator recursion (3.4). To this effect, we notice that the triangular factor L_i is now a column vector, which we denote by the lowercase letter l_i, the diagonal factor D_i is a scalar, denoted by d_i, the first η_i rows of G_i collapse to a single row vector, denoted by g_i, and the quantity $\hat{F}_i$ is also a scalar equal to f_{α_i} (we are using lower case letters to refer to quantities in a strongly regular step). A direct consequence of these facts is that we can now explicitly solve for d_i and l_i in (3.1a). More specifically, we get

(4.1a) $$d_i = \frac{g_i J g_i^*}{1 - |f_{\alpha_i}|^2} \ , \ \ l_i = (I_{n-\alpha_i} - f_{\alpha_i}^* F_i)^{-1} G_i J g_i^* .$$

Substituting these expressions into the generator recursion (3.4) we readily verify that it simplifies to

(4.1b) $$\begin{bmatrix} \mathbf{0}_{1\times r} \\ G_{i+1} \end{bmatrix} = \left\{ G_i + (\Phi_i - I_{n-\alpha_i}) G_i \frac{J g_i^* g_i}{g_i J g_i^*} \right\} \Theta_i \ ,$$

where Φ_i is a “Blaschke matrix” or “Blaschke-Potapov” factor (see [29]) of the form

(4.1c) $$\Phi_i = \frac{1 - \tau_i f_{\alpha_i}^*}{\tau_i - f_{\alpha_i}} \, (F_i - f_{\alpha_i} I_{n-\alpha_i})(I_{n-\alpha_i} - f_{\alpha_i}^* F_i)^{-1} .$$

The difference between (4.1b) and the general form (3.4) is that recursion (4.1b) is written in terms of F_i and G_i only, while expression (3.4) still involves L_i and D_i^{-1}.

We now move a step further and show that (4.1b) can be further simplified by conveniently choosing the free parameters Θ_i and τ_i. An obvious choice is $\Theta_i = I_r$ and $\tau_i = \frac{1+f_{\alpha_i}}{1+f_{\alpha_i}^*}$ (this choice for τ_i leads to $\Phi_i = (F_i - f_{\alpha_i} I_{n-\alpha_i})(I_{n-\alpha_i} - f_{\alpha_i}^* F_i)^{-1}$). There are other convenient choices for Θ_i as well, such as the one we describe next: a strongly regular step clearly implies that $d_i \neq 0$ and consequently $g_i J g_i^* \neq 0$. That is, g_i has nonzero $J-$norm. Hence, we can always find a $J-$unitary rotation Θ_i that reduces g_i to the form

(4.2) $$g_i \Theta_i = \begin{bmatrix} 0 & \ldots & 0 & x_i^{(j)} & 0 & \ldots & 0 \end{bmatrix} ,$$

with a nonzero entry in a single (convenient) column, say the j^{th} column. So assume we use this choice for Θ_i, which can be implemented in a variety of ways : we may use elementary rotations such as Givens or hyperbolic [12], Householder transformations [12, 30], etc.. Using the above choice leads to the following algorithm in the strongly regular case.

ALGORITHM 4.1 (Strongly-Regular Step). *The generator recursion for a strongly regular step is given by*

$$\text{(4.3)} \quad \begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = G_i\Theta_i \begin{bmatrix} I_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{r-j-1} \end{bmatrix} + \Phi_i G_i \Theta_i \begin{bmatrix} \mathbf{0}_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{r-j-1} \end{bmatrix} ,$$

where $\Phi_i = (F_i - f_{\alpha_i} I_{n-\alpha_i})(I_{n-\alpha_i} - f^*_{\alpha_i} F_i)^{-1}$. *That is,* G_{i+1} *is obtained as follows: choose a convenient* $J-$*unitary rotation that reduces the first row of* G_i *to the form (4.2); multiply the* j^{th} *column of* $G_i\Theta_i$ *by* Φ_i *and keep all other columns unchanged; these steps result in a generator* G_{i+1}. *The triangular factors are given by* $d_i = g_i J g_i^* / (1 - |f_{\alpha_i}|^2)$ *and* $l_i = (I_{n-\alpha_i} - f^*_{\alpha_i} F_i)^{-1} G_i \Theta_i J \begin{bmatrix} \mathbf{0} & x_i^{*(j)} & \mathbf{0} \end{bmatrix}^*$.

An alternative form for the generator recursion that corresponds to using $\Theta_i = I_r$, *instead of (4.2), is given by*

$$\begin{bmatrix} \mathbf{0}_{1\times r} \\ G_{i+1} \end{bmatrix} = G_i + (\Phi_i - I_{n-\alpha_i}) G_i \frac{J g_i^* g_i}{g_i J g_i^*} .$$

In this case, we compute l_i *via* $l_i = (I_{n-\alpha_i} - f^*_{\alpha_i} F_i)^{-1} G_i J g_i^*$, *and* d_i *is the leading entry of* l_i.

We shall assume throughout that F is a sparse matrix in the sense that computing Fx, for any $n \times 1$ column vector x, requires $O(n)$ operations. It can then be checked that each step of recursion (4.3) requires $O(r(n - \alpha_i))$ operations. Furthermore, we may not need to explicitly compute the inverse matrix $(I_{n-\alpha_i} - f^*_{\alpha_i} F_i)^{-1}$ that appears in the expressions for Φ_i and l_i. We can instead, in the case of l_i for example, solve a *triangular* system of linear equations of the form $(I_{n-\alpha_i} - f^*_{\alpha_i} F_i)x = G_i \Theta_i J \begin{bmatrix} \mathbf{0} & 1 & \mathbf{0} \end{bmatrix}^{\mathbf{T}}$. Moreover, in many applications the matrix F has zero diagonal entries (*i.e.*, $f_{\alpha_i} = 0$), in which case computing l_i and Φ_i is trivialized since the inverse term disappears.

As remarked above, a strongly regular step corresponds to $d_i \neq 0$, or equivalently, $g_i J g_i^* \neq 0$. There is however a trivial special case with $d_i = 0$, which can still be incorporated into a strongly regular step. This happens when g_i is itself a zero row vector. That is, G_i is of the form

$$G_i = \begin{bmatrix} \mathbf{0} \\ \bar{G}_i \end{bmatrix} .$$

This implies that the first row and column of R_i are zero. Going back to the description of the Schur reduction procedure in Section 2.1 we see that we can proceed in this special case by choosing $l_i = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}^{\mathbf{T}}$ and by setting $G_{i+1} = \bar{G}_i$ and F_{i+1} equal to the submatrix obtained by deleting the first row and column of F_i.

4.2. Block Steps. We now consider the case $\eta_i > 1$, and which we refer to as a *block* step. In this case, the triangular factors L_i and D_i are block matrices and it is not possible, in general, to solve for L_i and D_i and write down simple explicit expressions in terms of F_i and G_i only, as in the strongly regular case (see (4.1a)).

We can however proceed with (3.4) and use the simple choices $\Theta_i = I_r$ and $\tau_i = 1$. Under these conditions, we can rewrite the generator recursion (3.4) in the following form.

ALGORITHM 4.2 (Block Step). *The generator recursion for a block step can be expressed as*

$$\begin{bmatrix} \mathbf{0}_{\eta_i \times r} \\ G_{i+1} \end{bmatrix} = G_i + X_i \ , \tag{4.4a}$$

where $X_i = (F_i - I_{n-\alpha_i})L_i D_i^{-1}(I_{\eta_i} - \hat{F}_i)^{-1}\hat{G}_i$. *The triangular factor* L_i *is obtained by solving the equation (the leading* $\eta_i \times \eta_i$ *submatrix of* L_i *provides* D_i*)*

$$L_i = F_i L_i \hat{F}_i^* + G_i J \hat{G}_i^*. \tag{4.4b}$$

Expression (4.4a) shows that G_{i+1} is obtained by adding the last $(n - \alpha_{i+1})$ rows of X_i and G_i, while the top rows of X_i should cancel the top rows of G_i.

4.2.1. Computing $\mathbf{L_i}$. Solving for L_i in (4.4b) is not a major problem in most applications such as linear prediction, inverse scattering, solution of (structured) linear systems, least-squares problems, interpolation problems, etc.., because the matrix F arises in sparse forms, *e.g.*, $F = Z$, $F = Z + \lambda I$, $F =$ diagonal $\{f_0, f_1, \ldots, f_{n-1}\}$, $F = (Z+\lambda_0 I)\oplus(Z+\lambda_1 I)\oplus\ldots$, $F = Z+$diagonal $\{f_0, \ldots, f_{n-1}\}$. Consider, for instance, this last bidiagonal form. Denote the η_i columns of L_i by $L_i = \begin{bmatrix} l_{i0} & l_{i1} & \ldots & l_{i,\eta_i-1} \end{bmatrix}$, and the η_i rows of $\hat{G}_i$ by $\{g_{i0}, g_{i1}, \ldots, g_{i,\eta_i-1}\}$ $(g_i = g_{i0})$. It is then straightforward to check, using (4.4b), that the columns of L_i can be recursively computed as follows:

$$\begin{aligned} l_{i0} &= (I_{n-\alpha_i} - f_{\alpha_i}^* F_i)^{-1} G_i J g_{i0}^* \\ l_{ij} &= (I_{n-\alpha_i} - f_{\alpha_i+j}^* F_i)^{-1}\left[G_i J g_{ij}^* + F_i l_{i,j-1}\right], \quad \text{for } j = 1, \ldots, \eta_i - 1. \end{aligned}$$

Once again, the inversion $(I_{n-\alpha_i} - f_{\alpha_i+j}^* F_i)^{-1}$ can be avoided by solving a sparse triangular system of linear equations. The computational complexity needed in computing L_i is $O(r\eta_i(n-\alpha_i))$.

4.2.2. Computing $\mathbf{X_i}$. We now consider the operation count for one possibility for computing X_i (other possibilities clearly exist). Recall that L_i has D_i as its leading block. To show this explicitly we partition L_i as follows: $L_i = \begin{bmatrix} D_i^{\mathbf{T}} & W_i^{\mathbf{T}} \end{bmatrix}^{\mathbf{T}}$. Then $L_i D_i^{-1} = \begin{bmatrix} I_{\eta_i} & (W_i D_i^{-1})^{\mathbf{T}} \end{bmatrix}^{\mathbf{T}}$. At this stage we introduce the QR decomposition of D_i, viz., $D_i = Q_i P_i$, where Q_i is an $\eta_i \times \eta_i$ unitary matrix $(Q_i Q_i^* = I_{\eta_i})$ and P_i is an $\eta_i \times \eta_i$ nonsingular upper triangular matrix. Invoking the fact that D_i is Hermitian (*i.e.*, $Q_i P_i = P_i^* Q_i^*$) we conclude that $D_i^{-1} = Q_i P_i^{-*}$. The point is that we shall show later in Section 6 that Q_i and P_i can be efficiently computed with $O(\eta_i^2)$ operations by using only *strongly regular steps* (this is despite the fact that the leading minors of D_i may be singular). Assume, for the moment, that this is indeed the case. We can then rewrite X_i in the form

$$X_i = (F_i - I_{n-\alpha_i})\begin{bmatrix} P_i^* \\ W_i Q_i \end{bmatrix} P_i^{-*}(I_{\eta_i} - \hat{F}_i)^{-1}\hat{G}_i. \tag{4.5}$$

We now evaluate the operation count needed in computing X_i. The term $Y_1 = (I_{\eta_i} - \hat{F}_i)^{-1}\hat{G}_i$ can be evaluated in $O(r\eta_i)$ operations (by solving r lower triangular linear systems, for instance). The product $Y_2 = P_i^{-*}Y_1$ can also be reduced to the solution of r triangular linear systems, viz., $P_i^* Y_2 = Y_1$, and thus requires $O(r\eta_i^2)$ operations. The term $Y_3 = W_i Q_i Y_2$ requires $O((\eta_i^2 + r\eta_i)(n - \alpha_{i+1}))$ operations.

Finally computing the last $(n-\alpha_{i+1})$ rows of $(F_i - I_{n-\alpha_i})\begin{bmatrix} Y_1 \\ Y_3 \end{bmatrix}$ requires $O(\eta_i(n-\alpha_{i+1}))$ operations.

It is not necessary to perform these computations in the above specified order. Other orders are possible and may be more suitable depending on the problem at hand. We may even ignore the QR factorization of D_i altogether and simply invert D_i. But we opted here for introducing the QR representation of D_i simply because, as we shall show in a later section, this factorization can be computed rather efficiently due to the Toeplitz-like structure of R and, moreover, it will also lead to an alternative convenient factorization for R itself, as shown in the next section.

But for now we note that the computational cost involved in computing G_{i+1} and L_i in the block case is $O((n-\alpha_{i+1})(\eta_i^2 + \eta_i + 2r\eta_i) + r\eta_i^2 + r\eta_i + r\eta_i\eta_{i-1})$ operations. To get an idea of the overall computational complexity, *i.e.*, for $i = 0, 1, \ldots, t-1$, we assume that the $\eta_i' s$ are equal, viz., $\eta_0 = \eta_1 = \ldots = \eta_{t-1} = n/t$. It is then straightforward to verify that the above operation count reduces to $O(rn^2 + \frac{n^3}{t})$. (In the strongly regular case we have $t = n$ and $\eta_i = 1$, in which case the complexity reduces to the usual $O(rn^2)$ figure).

4.3. An Alternative Triangular Factorization. The factors L_i and D_i lead to a triangular factorization of the form $R = LD_B^{-1}L^*$, as discussed in Section 2.1, where D_B is block diagonal with entries equal to D_i and L_i is block lower triangular. We can instead use the QR factors of D_i to write an alternative factorization for R, where L is replaced by a *lower* triangular matrix $\hat{L}$, and D_B is replaced by a block diagonal matrix $\hat{D}_B$ with *unitary* and *triangular* blocks. To clarify this, we introduce the block-diagonal unitary matrix $Q = Q_0 \oplus Q_1 \oplus \ldots \oplus Q_{t-1}$ and the block diagonal matrix $P = P_0^{-*} \oplus P_1^{-*} \oplus \ldots \oplus P_{t-1}^{-*}$, where the diagonal blocks P_i^{-*} are lower triangular. Then $LD_B^{-1}L^* = LQQ^*D_B^{-1}QQ^*L^*$. If we define $\hat{L} = LQ$ then we obtain the alternative factorization

$$R = \hat{L}\underbrace{PQ}_{\hat{D}_B^{-1}}\hat{L}^*,$$

where $\hat{L}$ is lower triangular. In fact, the (nonzero part of the) i^{th} block column of $\hat{L}$ has the form $\begin{bmatrix} P_i^* \\ W_iQ_i \end{bmatrix}$, where P_i^* is lower triangular and the term W_iQ_i has already been computed in the generator recursion. We further remark that the inverses P_i^{-*} in P may not be needed explicitly since using the factorization $R = \hat{L}\hat{D}_B^{-1}\hat{L}^*$ to solve a linear system of equations, for example, requires knowledge of the $P_i' s$ only. In summary, we get the following algorithm.

ALGORITHM 4.3 (Fast Block Triangular Factorization). *Consider a Hermitian invertible and Toeplitz-like matrix R, viz., R satisfies $R - FRF^* = GJG^*$. A triangular factorization for R can be recursively computed in $O(rn^2 + \frac{n^3}{t})$ operations as follows: start with $G_0 = G, F_0 = F$, and repeat for $i \geq 0$:*

1. *At step i we have F_i and G_i.*
2. *Choose the size η_i of block Schur complementation step.*
3. *If $\eta_i = 1$ then update G_i to G_{i+1} using Algorithm 4.1 and compute the corresponding $l_i = \begin{bmatrix} d_i & w_i^{\mathbf{T}} \end{bmatrix}^{\mathbf{T}}$. A QR factorization for d_i can be trivially chosen as $q_i = 1$ and $p_i = d_i$.*

4. *If $\eta_i > 1$ then compute $L_i = \begin{bmatrix} D_i^{\mathbf{T}} & W_i^{\mathbf{T}} \end{bmatrix}^{\mathbf{T}}$ as described in Section 4.2.1 and determine the QR factors of D_i, viz., $D_i = Q_i P_i$ as described in Section 6. Also update G_i to G_{i+1} using Algorithm 4.2.*

5. *Construct the (nonzero parts of the) columns of $\hat{L}$ via $\begin{bmatrix} p_i^* \\ w_i q_i \end{bmatrix}$ or $\begin{bmatrix} P_i^* \\ W_i Q_i \end{bmatrix}$.*

This leads to a triangular factorization of the form $R = \hat{L} P Q \hat{L}^*$ where $Q = Q_0 \oplus Q_1 \oplus \ldots \oplus Q_{t-1}$ and $P = P_0^{-*} \oplus P_1^{-*} \oplus \ldots \oplus P_{t-1}^{-*}$.

The standard block triangular factorization, $R = L D_B^{-1} L^*$, can also be obtained by simply ignoring the QR factorizations specified above and directly using the L_i and D_i.

5. One Possibility for Choosing the Block Sizes η_i: The Exact Case.

As mentioned earlier, the sizes of the block steps (η_i) can be determined in different ways. They may denote the smallest (exact) nonsingular minors, or the sizes of numerically well-conditioned blocks, or some other convenient choices. In this section we shall focus, however, on the first choice in order to highlight some theoretically interesting features that arise in the *exactly* singular case. But we hasten to add that the block factorization algorithm of the previous section is equally applicable to other choices for the η_i.

5.1. Checking for $\eta_i = 1, 2, 3$.

We first remark that for a Toeplitz-like matrix R as in (2.3), determining whether $\eta_i = 1, 2$, or 3 in the exactly singular case is a simple task. To clarify this, recall from Theorem 3.1 that the successive Schur complements of R are also Toeplitz-like, viz., they satisfy displacement equations of the form

$$R_i - F_i R_i F_i^* = G_i J G_i^* \quad , \tag{5.1}$$

where F_i is lower triangular with diagonal entries equal to $\{f_{\alpha_i}, f_{\alpha_i+1}, \ldots, f_{n-1}\}$. It thus follows that the top-left corner element of R_i is given by (where g_{i0} denotes the first row of G_i) $d_i = g_{i0} J g_{i0}^* / (1 - f_{\alpha_i} f_{\alpha_i}^*)$. If $d_i \neq 0$, or equivalently, $g_{i0} J g_{i0}^* \neq 0$, then $\eta_i = 1$. If this is not the case, then we have to check for the nonsingularity of the 2×2 leading submatrix of R_i, which has to be of the form

$$\begin{bmatrix} 0 & r_{01}^{(i)} \\ r_{01}^{*(i)} & r_{11}^{(i)} \end{bmatrix}.$$

Using (5.1) it is easy to verify that $r_{01}^{(i)} = g_{i0} J g_{i1}^* / (1 - f_{\alpha_i} f_{\alpha_i+1}^*)$, which implies that $\eta_i = 2$ if, and only if, $g_{i0} J g_{i0}^* = 0$ and $g_{i0} J g_{i1}^* \neq 0$. If this test fails then we proceed to check for the leading 3×3 submatrix of R_i, viz.,

$$\begin{bmatrix} 0 & 0 & r_{02}^{(i)} \\ 0 & r_{11}^{(i)} & r_{12}^{(i)} \\ r_{02}^{*(i)} & r_{12}^{*(i)} & r_{22}^{(i)} \end{bmatrix} , \tag{5.2}$$

where, using (5.1) again, $r_{02}^{(i)} = g_{i0} J g_{i2}^* / (1 - f_{\alpha_i} f_{\alpha_i+2}^*)$, $r_{11}^{(i)} = g_{i1} J g_{i1}^* / (1 - f_{\alpha_i+1} f_{\alpha_i+1}^*)$, $r_{12}^{(i)} = g_{i1} J g_{i2}^* / (1 - f_{\alpha_i+1} f_{\alpha_i+2}^*)$, and $r_{22}^{(i)} = g_{i2} J g_{i2}^* / (1 - f_{\alpha_i+2} f_{\alpha_i+2}^*)$. Hence, for $\eta_i = 3$ we need $g_{i0} J g_{i2}^* \neq 0$ and $g_{i1} J g_{i1}^* \neq 0$. In summary we have the following.

LEMMA 5.1. *The following are simple tests for $\eta_i = 1, 2$, or 3 in the exactly singular case:*

If $g_{i0}Jg_{i0}^* \neq 0$ *then* $\eta_i = 1$
else if $g_{i0}Jg_{i1}^* \neq 0$ *then* $\eta_i = 2$
else if $g_{i0}Jg_{i2}^* \neq 0$ *and* $g_{i1}Jg_{i1}^* \neq 0$ *then* $\eta_i = 3$
else $\eta_i \geq 4$.

Observe that for $\eta_i \leq 3$ the leading nonsingular submatrix of R_i has a reversed lower triangular form. The inversion or QR factorization of these submatrices can be easily evaluated. For example, the QR decomposition of (5.2) is

$$D_i = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} r_{02}^{*(i)} & r_{12}^{*(i)} & r_{22}^{(i)} \\ 0 & r_{11}^{(i)} & r_{12}^{(i)} \\ 0 & 0 & r_{02}^{(i)} \end{bmatrix}.$$

Moreover, $\hat{G}_iJ\hat{G}_i^*$ also has the same reversed lower triangular form for $\eta_i = 1, 2, 3$ ($\hat{G}_i$ being the first η_i rows of G_i). For example, the conditions for $\eta_i = 3$ mean that $\hat{G}_iJ\hat{G}_i^*$ has to be of the form

$$\hat{G}_iJ\hat{G}_i^* = \begin{bmatrix} 0 & 0 & x \\ 0 & x & x \\ x & x & x \end{bmatrix}.$$

The above discussion suggests the following result.

LEMMA 5.2. *For some k, the leading $k \times k$ submatrix of $G_iJG_i^*$ has a nonsingular reversed lower triangular form with antidiagonal entries*

$$\{m_{0,k-1}, m_{1,k-2}, \ldots, m_{k-1,0}\},$$

if, and only if, $\eta_i = k$ and the leading nonsingular submatrix D_i has the same reversed lower triangular form.

Proof. The claim is certainly sufficient and necessary for $k = 1, 2, 3$, as discussed prior to the statement of the lemma. To verify the claim for larger values of k we consider a general $k \times k$ matrix E in reversed lower triangular form with antidiagonal entries $\{e_{0,k-1}, e_{1,k-2}, \ldots, e_{k-1,0}\}$, and let $\hat{F}_i$ denote the leading $k \times k$ submatrix of F_i. It is then easy to check that we can find a matrix E of this form that solves the equation

$$E - \hat{F}_iE\hat{F}_i^* = \begin{bmatrix} \text{O} & & m_{0,k-1} \\ & \cdot & \\ m_{k-1,0} & & \text{X} \end{bmatrix}.$$

In fact, we can write down explicit formulas for the desired entries of E in terms of the known entries on the right-hand side of the above equality. For example, the diagonal entries of E are given by

$$e_{0,k-1} = \frac{m_{0,k-1}}{1 - f_{\alpha_i}f_{\alpha_i+k-1}^*}, \quad e_{1,k-2} = \frac{m_{1,k-2}}{1 - f_{\alpha_i+1}f_{\alpha_i+k-2}^*}, \; \ldots \;,$$

which shows that we can always find an invertible solution E. But the leading $k \times k$ minor of R_i satisfies the same equation as E. It follows from the uniqueness condition $(1 - f_if_j^* \neq 0$, for all $i, j)$ that we must have $D_i = E$. Conversely, assume that D_i has the suggested reversed lower triangular form then it readily follows that $D_i - \hat{F}_i D_i \hat{F}_i^*$ is nonsingular and has the same reversed lower triangular form. □

We should stress that the lemma does *not* state that the nonsingular submatrices D_i always have a reversed lower triangular form. It only states that if D_i happens to have this form then $\hat{G}_i J \hat{G}_i^*$ also has the same form (and vice-versa). In fact, the triangular structure of D_i is not necessarily valid for higher sizes η_i as can be easily checked. For example, a nonsingular 4×4 leading submatrix of R_i may have one of the following forms

$$\begin{bmatrix} 0 & 0 & 0 & x \\ 0 & x & x & x \\ 0 & x & x & x \\ x & x & x & x \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 & 0 & x \\ 0 & 0 & x & x \\ 0 & x & x & x \\ x & x & x & x \end{bmatrix}, \quad \text{or} \quad \begin{bmatrix} 0 & 0 & x & x \\ 0 & 0 & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix}.$$

We can, however, give a stronger statement in the important special case of displacement rank $r = 2$.

5.2. Displacement Rank r = 2. We now consider the special case of structured matrices R as in (2.3) but with displacement rank $r = 2$, *i.e.,* G has two columns. We further assume that $J = (1 \oplus -1)$ and that F is a stable matrix, or equivalently, that its diagonal entries have less than unit-modulus magnitude,

$$R - FRF^* = \begin{bmatrix} \mathbf{u}_0 & \mathbf{v}_0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 & \mathbf{v}_0 \end{bmatrix}^*. \tag{5.3}$$

Our purpose is to show that for this class of structured matrices we can derive an explicit test for all $\eta_i's$ in the exactly singular case. Special cases of (5.3) were studied earlier in [16, 27]. Iohvidov [16] studied the special case of Toeplitz matrices, which corresponds to the special choice $F = Z$ and a special generator matrix of the form (recall expression (2.2))

$$G = \begin{bmatrix} 1 & c_1 & \dots & c_{n-1} \\ 0 & c_1 & \dots & c_{n-1} \end{bmatrix}^{\mathbf{T}}.$$

Pal and Kailath [26, 27] considered the wider class of so-called quasi-Toeplitz matrices, which still corresponds to $F = Z$ but one where the columns $\mathbf{u}_0$ and $\mathbf{v}_0$ of G are arbitrary and not as restricted as in the Toeplitz case above. Such matrices can be shown to be congruent to Toeplitz matrices in a certain sense, hence the name quasi-Toeplitz. The derivation in [26, 27] exploits this fact and, among other results, shows that the determination of the size of the smallest nonsingular minor is reduced to counting the number of repeated zeros at the origin of a certain polynomial.

We provide here a general statement that goes beyond the $F = Z$ case. We follow a matrix-based argument that also reveals under what conditions on F the derived test is not applicable. (see also [2] for generalizations of the Iohvidov laws using the theory of reproducing kernel Hilbert spaces).

We start again with the displacement equation of the i^{th} Schur complement, viz.,

$$R_i - F_i R_i F_i^* = G_i J G_i^* \ , \tag{5.4}$$

and denote the entries of the now two-column generator G_i by

$$G_i = \begin{bmatrix} u_{ii} & u_{i+1,i} & u_{i+2,i} & \dots \\ v_{ii} & v_{i+1,i} & v_{i+2,i} & \dots \end{bmatrix}^{\mathbf{T}} = \begin{bmatrix} \mathbf{u}_i & \mathbf{v}_i \end{bmatrix}.$$

Assume we encounter a singularity $d_i = 0$, or equivalently, $g_i J g_i^* = |u_{ii}|^2 - |v_{ii}|^2 = 0$. Then either of the following two cases could have happened: g_i is a zero row, which corresponds to the trivial case discussed at the end of Section 4.1, or g_i is a nonzero row, which corresponds to a *block step* that we now discuss in more details.

5.3. A Preliminary Result and Definitions. Before proving the main theorem we first state an easily verifiable result that follows from the following type of argument: an equality such as $g_i J g_i^* = 0$ clearly implies that v_{ii} and u_{ii} are related via $v_{ii} = u_{ii} e^{j\xi}$ for some phase angle $\xi \in [0, 2\pi]$. More generally, we have the following.

LEMMA 5.3. *The entries of the first k rows of G_i satisfy*

$$v_{l+i,i} = u_{l+i,i} e^{j\xi} , \quad l = 0, 1, \ldots, k-1,$$

for some phase angle $\xi \in [0, 2\pi]$, if, and only if, the leading $2k \times 2k$ submatrix of $G_i J G_i^*$ has the form

$$\begin{bmatrix} \mathbf{0}_{k\times k} & M_{k\times k} \\ M^*_{k\times k} & X \end{bmatrix} , \tag{5.5}$$

where M is a rank 1 matrix and X denotes irrelevant entries. That is, $G_i J G_i^*$ has a $k \times k$ leading zero block.

For a column vector $\mathbf{x}$ and a square matrix A, we let $K^m(A, \mathbf{x})$ denote the Krylov matrix $K^m(A, \mathbf{x}) = \begin{bmatrix} \mathbf{x} & A\mathbf{x} & \ldots & A^{m-1}\mathbf{x} \end{bmatrix}$. We further define some auxiliary quantities that will be used in the statement and proof of the next theorem: for a positive number k, we define the column vectors $\{\mathbf{a}, \mathbf{b}, \mathbf{x}, \mathbf{y}\}$ as follows:

$$\begin{bmatrix} \mathbf{a} & \mathbf{b} \end{bmatrix} = \begin{bmatrix} u_{ii} & v_{ii} \\ u_{i+1,i} & v_{i+1,i} \\ \vdots & \vdots \\ u_{i+k-1,i} & v_{i+k-1,i} \end{bmatrix} , \quad \begin{bmatrix} \mathbf{x} & \mathbf{y} \end{bmatrix} = \begin{bmatrix} u_{i+k,i} & v_{i+k,i} \\ u_{i+k+1,i} & v_{i+k+1,i} \\ \vdots & \vdots \\ u_{i+2k-1,i} & v_{i+2k-1,i} \end{bmatrix} . \tag{5.6a}$$

That is, $\{\mathbf{a}, \mathbf{b}\}$ contain the entries of the first k rows of G_i, while $\{\mathbf{x}, \mathbf{y}\}$ contain the entries of the next k rows of G_i. Recall that g_i is a nonzero row vector with zero $J-$norm. Consequently, both u_{ii} and v_{ii} must be nonzero since if one of them is zero then the other one must be zero, due to the relation $v_{ii} = u_{ii} e^{j\xi}$. We also define the column vectors

$$\rho = \frac{\mathbf{a} + e^{-j\xi}\mathbf{b}}{\sqrt{2}} , \quad \nu = \frac{\mathbf{x} - e^{-j\xi}\mathbf{y}}{\sqrt{2}} , \quad \text{for a given } \xi , \tag{5.6b}$$

and partition F_i as follows

$$F_i = \begin{bmatrix} \hat{F}_i & & \mathrm{O} \\ ? & \hat{A}_i & \\ ? & ? & ? \end{bmatrix} , \tag{5.6c}$$

where $\hat{F}_i$ and $\hat{A}_i$ are $k \times k$ lower triangular matrices.

5.4. Main Result for Displacement Rank $\mathbf{r} = 2$. The next result gives an explicit test for the determination of the sizes of the nonsingular minors for the class of structured matrices as in (5.3), with extra conditions on the entries of F. This extends earlier results in [16, 27].

THEOREM 5.4. *The size of the smallest nonsingular leading submatrix of R_i is $2k$ and has the block form*

$$\begin{bmatrix} \mathbf{0}_{k\times k} & N_{k\times k} \\ N^*_{k\times k} & C_{k\times k} \end{bmatrix} , \tag{5.7a}$$

where N is invertible if, and only if, the $k\times k$ matrix $K^\infty(\hat{F}_i,\rho)K^{*\infty}(\hat{A}_i,\nu)$ is invertible and the entries of the first k rows of G_i satisfy

$$v_{l+i,i} = u_{l+i,i}e^{j\xi} \ , \ \ l = 0, 1, \ldots, k-1 , \tag{5.7b}$$

for some phase angle $\xi \in [0, 2\pi]$.

Proof. If $u_{l+i,i}$ and $v_{l+i,i}$ satisfy (5.7b) then it is straightforward to verify that the leading $2k \times 2k$ submatrix of R_i has a leading zero block as in (5.7a) (similar to the argument in Lemma 5.3). The converse is also true. If the leading $2k \times 2k$ submatrix of R_i has a leading zero block as in (5.7a) then $u_{l+i,i}$ and $v_{l+i,i}$ satisfy (5.7b). We still need to prove that (5.7a) is the smallest nonsingular minor. For this purpose, it is enough to verify that N is invertible.

It follows from (5.4) that N satisfies the (non-Hermitian) displacement equation

$$N - \hat{F}_i N \hat{A}_i^* = \begin{bmatrix} \mathbf{a} & \mathbf{b} \end{bmatrix} J \begin{bmatrix} \mathbf{x} & \mathbf{y} \end{bmatrix}^* \ ,$$

where $\{\mathbf{a}, \mathbf{b}, \mathbf{x}, \mathbf{y}\}$ were defined in (5.6a). But conditions (5.7b) imply that $\mathbf{b} = e^{j\xi}\mathbf{a}$. Also, the eigenvalues of the lower triangular matrices $\hat{F}_i$ and $\hat{A}_i$ are strictly less than unit magnitude. Hence, we can write

$$\begin{aligned} N &= K^\infty(\hat{F}_i,\mathbf{a})K^{*\infty}(\hat{A}_i,\mathbf{x}) - K^\infty(\hat{F}_i,\mathbf{b})K^{*\infty}(\hat{A}_i,\mathbf{y}) \\ &= \frac{1}{2}\Big\{K^\infty(\hat{F}_i,\mathbf{a}+e^{-j\xi}\mathbf{b})K^{*\infty}(\hat{A}_i,\mathbf{x}-e^{-j\xi}\mathbf{y}) \ + \\ &\qquad K^\infty(\hat{F}_i,\mathbf{a}-e^{-j\xi}\mathbf{b})K^{*\infty}(\hat{A}_i,\mathbf{x}+e^{-j\xi}\mathbf{y})\Big\} \\ &= K^\infty(\hat{F}_i,\rho)K^{*\infty}(\hat{A}_i,\nu) \ , \end{aligned}$$

where ρ and ν were defined in (5.6b). We thus conclude that N is full rank. □

5.4.1. Remarks. The last theorem states that, provided the following condition is satisfied

$$K^\infty(\hat{F}_i,\rho)K^{*\infty}(\hat{A}_i,\nu) \ \ \textit{is invertible} \ , \tag{5.8}$$

the determination of η_i reduces to checking the proportionality condition (5.7b), viz., whether the first k elements of $\mathbf{v}_i$ are unit-modulus multiples of the first k elements of $\mathbf{u}_i$. It is clear that *necessary* conditions for (5.8) to hold are

$$K^\infty(\hat{F}_i,\rho) \ \ \textit{and} \ \ K^{*\infty}(\hat{A}_i,\nu) \ \ \textit{must have full rank.}$$

For those familiar with system theory [17], the above necessary conditions are equivalent to saying that the pair $(\hat{F}_i,\rho)$ must be controllable and the pair $(\hat{A}_i^*,\nu^*)$ must be observable. For example, if $\hat{F}_i$ is similar to a Jordan structure with repeated Jordan blocks for the same eigenvalue then the pair $(\hat{F}_i, \rho)$ will not be controllable. A similar remark holds for $\hat{A}_i$.

Furthermore, condition (5.8) is automatically satisfied in the special case $F = Z$ studied in [16, 27]. Indeed, $F = Z$ implies that $K^\infty(\hat{F}_i, \rho) = K^\infty(Z, \rho) = \begin{bmatrix} \mathbf{L}(\rho) & \mathbf{0} \end{bmatrix}$ and $K^\infty(\hat{A}_i, \nu) = K^\infty(Z, \nu) = \begin{bmatrix} \mathbf{L}(\nu) & \mathbf{0} \end{bmatrix}$, where the notation $\mathbf{L}(x)$ denotes a lower triangular Toeplitz matrix whose first column is x. But $\mathbf{L}(\rho)$ and $\mathbf{L}(\nu)$ are full rank matrices since the top entries of ρ and ν are nonzero. Hence, $\mathbf{L}(\rho)\mathbf{L}^*(\nu)$ is invertible and (5.8) is satisfied. It also follows that N is strongly regular.

Moreover, using (5.7a) we get

$$D_i^{-1} = \begin{bmatrix} -N^{-*}CN^{-1} & N^{-*} \\ N^{-1} & \mathbf{0} \end{bmatrix},$$

which shows that inverting D_i essentially reduces to inverting a strongly regular matrix N, which has a non-Hermitian Toeplitz-like structure. This can be done in strongly regular (*i.e.,* scalar) steps. Following this reasoning we can show that in this case ($F = Z$), the inversion of D_i (or N) and the generator recursion (4.4a) reduce to the algorithm derived in [27], which involves only scalar operations. We shall not go into the details here mainly because the derivation (and simplifications thereof) relies heavily on the special structure in question ($r = 2$ and $F = Z$). We instead focus on the case of higher displacement ranks ($r > 2$).

5.5. A Recursive Test for Displacement Ranks r >2. A conventional rank test for determining whether an arbitrary $n \times n$ matrix is invertible or not requires $O(n^3)$ operations. This figure can be reduced to $O(rn^2)$ in the case of structured matrices as discussed in Section 5.5.1. The following lemma states that if we are given a structured matrix R (*not necessarily strongly regular*), then checking whether R is invertible or not can be achieved by using only *strongly regular* Schur steps that are applied to an appropriately defined extended generator matrix.

LEMMA 5.5. *Let T be any $n \times n$ positive-definite matrix. Then an $n \times n$ Hermitian matrix R (not necessarily strongly regular) is invertible if, and only if, the extended $2n \times 2n$ matrix $\hat{R}$,*

$$\hat{R} = \begin{bmatrix} -T & R \\ R & \mathbf{0} \end{bmatrix},$$

is strongly regular.

Proof. The leading $n \times n$ submatrix of $\hat{R}$ is strongly regular since T is positive definite ($T > 0$). The Schur complement with respect to the leading $n \times n$ block is $RT^{-1}R$. The claim now follows by observing that $RT^{-1}R$ is positive-definite if, and only if, R is invertible. □

In other words, if we apply the generalized Schur algorithm to a generator of $\hat{R}$ and a singularity is (not) encountered then we conclude that the original R is (not) singular. But we first need to check whether the extended matrix $\hat{R}$ is structured. For this purpose, recall that R is Toeplitz-like, viz., $R - FRF^* = GJG^*$. It then follows that

$$\hat{R} - \begin{bmatrix} F & \mathbf{0} \\ \mathbf{0} & F \end{bmatrix} \hat{R} \begin{bmatrix} F & \mathbf{0} \\ \mathbf{0} & F \end{bmatrix}^* = \begin{bmatrix} FTF^* - T & GJG^* \\ GJG^* & \mathbf{0} \end{bmatrix}, \tag{5.9}$$

which shows that $\hat{R}$ has a Toeplitz-like structure if $(FTF^* - T)$ has low rank, say β. So assume that this is the case. Then we can (nonuniquely) factor $(FTF^* - T)$ as follows: $FTF^* - T = VJ_\beta V^*$, where V is an $n \times \beta$ generator matrix and J_β is a $\beta \times \beta$ signature matrix with $\beta \ll n$. This means that we need to choose a positive-definite matrix T that has low displacement rank with respect to F. We shall show later in this section how such choices (of T and, consequently, of V and J_β) can be made. Then we can factor the right-hand side of (5.9) as follows

$$\begin{bmatrix} FTF^* - T & GJG^* \\ GJG^* & \mathbf{0} \end{bmatrix} = \begin{bmatrix} V & \mathbf{0} & G \\ \mathbf{0} & G & \mathbf{0} \end{bmatrix} \mathcal{J} \begin{bmatrix} V & \mathbf{0} & G \\ \mathbf{0} & G & \mathbf{0} \end{bmatrix}^*, \; \mathcal{J} = \begin{bmatrix} J_\beta & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & J \\ \mathbf{0} & J & \mathbf{0} \end{bmatrix},$$

where $\mathcal{J}$ satisfies $\mathcal{J}^2 = I$. We thus conclude that a possible (not necessarily minimal) $2n \times (2r + \beta)$ generator for $\hat{R}$ is

$$\hat{H} = \begin{bmatrix} V & \mathbf{0} & G \\ \mathbf{0} & G & \mathbf{0} \end{bmatrix}.$$

We can now proceed by applying Algorithm 4.1 to $\hat{R}$ with the initial conditions $G_0 = \hat{H}$, $F_0 = (F \oplus F)$, and $J = \mathcal{J}$. The first n steps of the algorithm will clearly yield negative diagonal entries $\{d_i,\ i = 0, 1, \ldots, n-1, d_i < 0\}$ since $-T$ is negative definite. The n^{th} generator, G_n, will be a generator of the Schur complement $RT^{-1}R$ of $\hat{R}$ with respect to its leading $n \times n$ submatrix $(-T)$. If in the subsequent generator steps $(i = n, n+1, \ldots, 2n-1)$ we obtain a zero d_i, (*i.e.,* a row vector g_i with a zero $\mathcal{J}-$norm), then we stop and conclude that the original matrix R is singular. Otherwise, R is nonsingular. This test requires at most $O((2r + \beta)n^2)$ operations (which is the computational effort due to applying the strongly regular Schur algorithm to $\hat{R}$). This should be compared with a conventional $O(n^3)$ rank test applied to R. A computational advantage results when $(2r + \beta) \ll n$.

5.5.1. Specializing to the η_i's. We now show how to recursively use the above procedure in order to compute the $\eta_i's$. Recall that the successive Schur complements R_i of the Toeplitz-like matrix R satisfy displacement equations of the form (5.1), and our objective is to determine the size η_i of the smallest nonsingular submatrix of R_i. We already know how to check whether $\eta_i \leq 3$ (as discussed in Section 5.1). For higher values of η_i we can proceed as suggested by the result of Lemma 5.5.

For this purpose, assume we have already chosen a positive-definite matrix T_i that has low displacement rank with respect to F_i (as described ahead), and introduce the factorization

$$F_i T_i F_i^* - T_i = V_i J_\beta V_i^*.$$

We further define E_k, T_k, $\hat{F}_k, \hat{G}_k$, and V_k to denote the leading $k \times k$, $k \times k$, $k \times k$, $k \times r$, and $k \times \beta$ submatrices of R_i, T_i, F_i, G_i, and V_i, respectively. It follows from (5.1) that E_k is also a Toeplitz-like matrix since $E_k - \hat{F}_k E_k \hat{F}_k^* = \hat{G}_k J \hat{G}_k^*$. We can now use the result of Lemma 5.5 to check whether E_k is nonsingular by forming the corresponding extended matrix $\hat{E}_k$,

$$\hat{E}_k = \begin{bmatrix} -T_k & E_k \\ E_k & \mathbf{0} \end{bmatrix},$$

and checking for its strong regularity. If E_k turns out to be invertible then we set $\eta_i = k$, otherwise we check for the next submatrix E_{k+1}, and so on. A generator for

$\hat{E}_k$ is given by

$$\hat{H}_k = \begin{bmatrix} V_k & \mathbf{0} & \hat{G}_k \\ \mathbf{0} & \hat{G}_k & \mathbf{0} \end{bmatrix}, \quad \mathcal{J} = \begin{bmatrix} J_\beta & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & J \\ \mathbf{0} & J & \mathbf{0} \end{bmatrix}, \tag{5.10a}$$

and we thus apply the generator recursion of Algorithm 4.1 with the initial conditions $G_0 = \hat{H}_k, F_0 = (\hat{F}_k \oplus \hat{F}_k),\ J = \mathcal{J}$. More precisely, we can rewrite recursion (4.3) for the present case as follows: start with $\hat{H}_{k,0} = \hat{H}_k$ and repeat for $i = 0, 1, \ldots, 2k-1$,

$$\begin{bmatrix} \mathbf{0}_{1\times(2r+\beta)} \\ \\ \hat{H}_{k,i+1} \end{bmatrix} = \hat{H}_{k,i}\Theta_i \begin{bmatrix} I_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{0} & 0 & I \end{bmatrix} + \Phi_i \hat{H}_{k,i}\Theta_i \begin{bmatrix} \mathbf{0}_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{5.10b}$$

where Φ_i is as defined in Algorithm 4.1 with $F_0 = (\hat{F}_k \oplus \hat{F}_k)$, and Θ_i is a $\mathcal{J}-$unitary rotation that reduces the first row of $\hat{H}_{k,i}$ (denoted by $h_{k,i}$) to the form $h_{k,i}\Theta_i = \begin{bmatrix} \mathbf{0} & \kappa_{k,j}^{(i)} & \mathbf{0} \end{bmatrix}$, where $\kappa_{k,j}^{(i)}$ is a scalar at a convenient j^{th} column position.

The test starts by applying the above recursion to $\hat{H}_k$. Schematically, we form (Θ_0, Φ_0) and apply the recursion in order to obtain $\hat{H}_{k,1}$. We then form (Θ_1, Φ_1) and apply the recursion again to obtain $\hat{H}_{k,2}$, and so on. Each such step corresponds to a transformation determined by the pair (Θ_i, Φ_i). After the first k transformations $(i = 0, 1, \ldots, k-1)$, we obtain $\hat{H}_{k,k}$, which is a generator for $E_k T_k^{-1} E_k$. We then proceed by applying (at most) k more steps of the recursion. E_k will then be declared singular if, at any of the steps $i = k, \ldots, 2k-1$ we encounter a row $h_{k,i}$ with a zero $\mathcal{J}-$norm, viz., $h_{k,i}\mathcal{J}h_{k,i}^* = |\kappa_{k,j}^{(i)}|^2 = 0$, for some $i \geq k$.

If the procedure ends without encountering a singularity then $\eta_i = k$, otherwise we have to check for the next leading submatrix E_{k+1}. Now, the generators of $\hat{E}_{k+1}$ and $\hat{E}_k$ are closely related since V_k and $\hat{G}_k$ are submatrices of V_{k+1} and $\hat{G}_{k+1}$, respectively. That is,

$$V_{k+1} = \begin{bmatrix} V_k \\ b_k \end{bmatrix}, \quad \hat{G}_{k+1} = \begin{bmatrix} \hat{G}_k \\ a_k \end{bmatrix},$$

for some row vectors a_k and b_k. Hence, $\hat{H}_{k+1}$ and $\hat{H}_k$ differ only at rows $(k+1)$ and $2(k+1)$, viz.,

$$\hat{H}_{k+1} = \begin{bmatrix} V_k & \mathbf{0} & \hat{G}_k \\ b_k & \mathbf{0} & a_k \\ \mathbf{0} & \hat{G}_k & \mathbf{0} \\ \mathbf{0} & a_k & \mathbf{0} \end{bmatrix} = \begin{bmatrix} V_{k+1} & \mathbf{0} & \hat{G}_{k+1} \\ \mathbf{0} & \hat{G}_{k+1} & \mathbf{0} \end{bmatrix}.$$

Therefore, $\hat{H}_k$ and $\hat{H}_{k+1}$ share the same first k Schur reduction steps. This means that in order to obtain a generator for $E_{k+1}T_{k+1}^{-1}E_{k+1}$, we first apply the *last* $(k+2)$ rows of $\hat{H}_{k+1}$, viz.,

$$\begin{bmatrix} b_k & \mathbf{0} & a_k \\ \mathbf{0} & \hat{G}_k & \mathbf{0} \\ \mathbf{0} & a_k & \mathbf{0} \end{bmatrix},$$

through the first k transformations $\{(\Theta_i, \Phi_i),\ \ i = 0, \ldots, k-1\}$ that were applied to $\hat{H}_k$. This leads to $\hat{H}_{k+1,k}$. We now apply one more transformation (Θ_k, Φ_k) to $\hat{H}_{k+1,k}$

in order to get $\hat{H}_{k+1,k+1}$, which is a generator for $E_{k+1}T_{k+1}^{-1}E_{k+1}$. We then proceed by applying at most $(k+1)$ steps in order to check for the positive-definiteness of $E_{k+1}T_{k+1}^{-1}E_{k+1}$, and so on. The size η_i is determined when, for some k, we are able to complete the whole recursive procedure without encountering a singularity. In this case, we get $k = \eta_i$ and hence $E_k = E_{\eta_i} = D_i$. The η_i transformations $\{\Theta_i, \Phi_i,\ i = 0, \ldots, \eta_i - 1\}$ used in this last test will be relevant in Section 6 while computing the QR factors of D_i.

It can be verified that $O(k^2(r+\beta))$ operations are needed for each k. This should be compared with the following alternative procedure: For each k, compute the leading $k \times k$ submatrix and check whether it is singular using a conventional rank test. This requires $O(k^3)$ operations and does not exploit the underlying (displacement) structure. A computational advantage results when $(r+\beta)$ is smaller than k.

ALGORITHM 5.6. *To check whether the $k \times k$ leading submatrix of R_i is nonsingular we proceed as follows:*

1. *Form a generator pair $(\hat{H}_k, \mathcal{J})$ as in (5.10a).*
2. *Apply k steps of recursion (5.10b) starting with $\hat{H}_{k,0} = \hat{H}_k, F_0 = (\hat{F}_k \oplus \hat{F}_k)$, and $J = \mathcal{J}$. This leads to $\hat{H}_{k,k}$.*
3. *Apply more steps of recursion (5.10b) to $\hat{H}_{k,k}$. If $h_{k,j}$ is found to have zero $\mathcal{J}-$norm, for some $k \leq j \leq 2k-1$, then E_k is declared singular ($\eta_i > k$). Otherwise $\eta_i = k$.*
4. *To check for the higher order $(k+1) \times (k+1)$ submatrix we essentially repeat the same procedure, except that we exploit the fact that $\hat{H}_k$ and $\hat{H}_{k+1}$ differ only in two rows as follows:*

(4a) Apply the last $(k+2)$ rows of $\hat{H}_{k+1}$ through the k transformations

$$\{(\Theta_i, \Phi_i),\ i = 0, \ldots, k-1\}$$

that were applied to $\hat{H}_k$. This leads to $\hat{H}_{k+1,k}$.

(4b) Apply one more step to get $\hat{H}_{k+1,k+1}$.

(4c) Go back to step 3 and repeat.

5.5.2. Choosing T. We now show how to choose a positive-definite matrix T that has low displacement rank with respect to an F. This choice is rather trivial in some special (though frequent) cases such as $F = Z$ or $F = Z \oplus Z \oplus \ldots \oplus Z$. For these cases, a simple choice is $T = I$. For example, choosing $T = I$ in the $F = Z$ case leads to $\beta = 1,\ J_\beta = -1$, and $V = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}^{\mathbf{T}}$, viz.,

$$ZZ^* - I = -\begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}^*.$$

On the other hand, for a diagonal or bidiagonal matrix F with distinct diagonal entries, the choice $T = I$ would usually lead to a full displacement rank $\beta = n$, *i.e.,* $FF^* - I$ would generally have rank n, which substantially increases the computational cost of the recursive tests. However, for such cases, it is still possible to choose a positive-definite matrix T that leads to a low displacement rank β. For this purpose, we exploit connections with analytic interpolation theory.

Assume, for instance, that we have an $n \times n$ diagonal matrix F with distinct and stable entries f_i ($|f_i| < 1$), and choose *any* scalar function $s(z)$ that is analytic and strictly bounded by unity inside the open unit disc $|z| < 1$, viz., $\sup_{|z|<1} |s(z)| < 1$. We say that $s(z)$ is a Schur-type function [1, 36]. We further introduce the matrices

V and J_β given by

$$V = \begin{bmatrix} 1 & s(f_0) \\ 1 & s(f_1) \\ \vdots & \vdots \\ 1 & s(f_{n-1}) \end{bmatrix} \ , \ J_\beta = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \ ,$$

and define T to be the solution of the following displacement equation

$$T - FTF^* = V \begin{bmatrix} 1 & \\ & -1 \end{bmatrix} V^*.$$

It is then a standard result in analytic interpolation theory (see, *e.g.,* [1, 32, 34]) that T is a positive-definite matrix since $s(z)$ is of Schur-type. So all we need to do is to choose a Schur function $s(z)$ and define V and J_β as above. We don't even need to explicitly determine the corresponding T since the recursive algorithm described in the previous section uses (V, J_β) and not T.

For a bidiagonal matrix $F = Z + \text{diag.}\{f_0, \ldots, f_{n-1}\}$ with distinct stable entries f_i ($|f_i| < 1$), we again choose a Schur function $s(z)$ and define

$$V = \begin{bmatrix} 1 & \phi_0 \\ 0 & \phi_1 \\ \vdots & \vdots \\ 0 & \phi_{n-1} \end{bmatrix} \ , \ J = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \ ,$$

where the $\phi_i's$ denote the first n Newton-series coefficients associated with $s(z)$. These coefficients can be recursively determined via the so-called *divided difference recursion* as follows: start with $s_0(z) = s(z)$ and then use

$$s_i(z) = \frac{s_{i-1}(z) - \phi_{i-1}}{z - f_{i-1}} \ , \quad \phi_i = s_i(f_i) \ .$$

It also follows that the associated matrix T is positive-definite [31, 34]. For a more general matrix F with $r_i \times r_i$ Jordan blocks, viz., $F = (Z + f_0 I) \oplus (Z + f_1 I) \oplus (Z + f_2 I) \oplus \ldots$, with f_i distinct and $|f_i| < 1$, we define [31, 34]

$$V = \begin{bmatrix} 1 & s(f_0) \\ 0 & s^{(1)}(f_0) \\ \vdots & \vdots \\ 0 & \frac{1}{(r_0-1)!} s^{(r_0-1)}(f_0) \\ 1 & s(f_1) \\ \vdots & \vdots \\ 0 & \frac{1}{(r_1-1)!} s^{(r_1-1)}(f_1) \\ \vdots & \vdots \end{bmatrix} \ , \ J = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \ ,$$

where $s^{(j)}(f_i)$ denotes the j^{th} derivative of $s(z)$ at f_i.

6. QR Factorization of the $\mathbf{D_i}$'s. Once the sizes η_i have been chosen, say as described in the above sections for the exactly singular case or as numerically well-conditioned blocks, we still need to show how to compute the QR factors of D_i, viz., $D_i = Q_i P_i$, where Q_i is an $\eta_i \times \eta_i$ unitary matrix and P_i is an $\eta_i \times \eta_i$ nonsingular upper triangular matrix. This is useful if the alternative triangular factorization of Section 4.3 is desired. The discussion that follows assumes, for brevity of argument and notation, that the η_i have been chosen as described in the above section. But it is rather immediate to see that the result is equally applicable for other choices of the η_i. The main point is simply the following: to compute the QR factors of D_i we form a $3\eta_i \times 3\eta_i$ extended block matrix and apply $2\eta_i$ steps of the (strongly-regular) Schur algorithm to it. Once this is done, the QR factors can be read out from the resulting triangular factors.

So we first assume that F is such that the matrix $T = I$ has low displacement rank with respect to it. We then consider the following $3\eta_i \times 3\eta_i$ extended matrix

$$\hat{D}_i = \begin{bmatrix} -I & D_i & \mathbf{0} \\ D_i & \mathbf{0} & D_i \\ \mathbf{0} & D_i & \mathbf{0} \end{bmatrix} ,$$

which turns out to also be Toeplitz-like with respect to $(\hat{F}_i \oplus \hat{F}_i \oplus \hat{F}_i)$ and with a generator matrix of the form

$$\begin{bmatrix} V_{\eta_i} & \mathbf{0} & \hat{G}_i \\ \mathbf{0} & \hat{G}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \hat{G}_i \end{bmatrix} , \quad \mathcal{J} = \begin{bmatrix} J_\beta & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & J \\ \mathbf{0} & J & \mathbf{0} \end{bmatrix}, \tag{6.1a}$$

where $\hat{F}_i \hat{F}_i^* - I = V_{\eta_i} J_\beta V_{\eta_i}^*$. The first two block rows of the above generator are the same block rows of the generator $\hat{H}_{\eta_i}$ of $\hat{E}_{\eta_i}$ (refer to (5.10a)), viz.,

$$\hat{H}_{\eta_i} = \begin{bmatrix} V_{\eta_i} & \mathbf{0} & \hat{G}_i \\ \mathbf{0} & \hat{G}_i & \mathbf{0} \end{bmatrix} .$$

Therefore, if we apply to the generator (6.1a) of $\hat{D}_i$ the same η_i transformations $\{(\Theta_i, \Phi_i),\ i = 0, 1, \ldots, \eta_i - 1\}$ that were applied to $\hat{H}_{\eta_i}$, we then obtain a generator matrix for the Schur complement of the leading block matrix in $\hat{D}_i$, which is equal to $\bar{D}_i$ below:

$$\bar{D}_i = \begin{bmatrix} D_i D_i & D_i \\ D_i & \mathbf{0} \end{bmatrix} . \tag{6.1b}$$

If we denote this generator of $\bar{D}_i$ by $\bar{S}_i$ then $\bar{S}_i$ is clearly of the form

$$\bar{S}_i = \begin{bmatrix} \hat{H}_{\eta_i, \eta_i} \\ \tilde{S}_i \end{bmatrix} , \tag{6.1c}$$

where $\tilde{S}_i$ results from the application of the above η_i transformations $\{\Theta_i, \Phi_i\}$ to the last block row in the generator of $\hat{D}_i$, viz., $\begin{bmatrix} \mathbf{0} & \mathbf{0} & \hat{G}_i \end{bmatrix}$. In summary, we already know how to obtain a generator for (the $2\eta_i \times 2\eta_i$ matrix) $\bar{D}_i$ in (6.1b): just update the block row $\begin{bmatrix} \mathbf{0} & \mathbf{0} & \hat{G}_i \end{bmatrix}$ via the transformations (Θ_i, Φ_i) and construct $\bar{S}_i$.

Once a generator for $\bar{D}_i$ is available, we can then use it to determine the first η_i triangular factors of $\bar{D}_i$. For this purpose, we need only apply η_i steps of the strongly regular Algorithm 4.1 starting with $G_0 = \bar{S}_i, F_0 = (\hat{F}_i \oplus \hat{F}_i)$, and $J = \mathcal{J}$. These steps however, are completely specified in terms of the same transformations $\{(\Theta_i, \Phi_i),\ i = \eta_i, \ldots, 2\eta_i - 1\}$ that were applied to $\hat{H}_{\eta_i,\eta_i}$ while checking the positive definiteness of $E_{\eta_i} E_{\eta_i}$. So we just need to update the last block row $\tilde{S}_i$ via the same transformations.

The point is that we can read out the desired QR factors Q_i and P_i from these first η_i triangular factors of $\bar{D}_i$. To see this, we denote the first η_i triangular factors by

$$L_d = \begin{bmatrix} l_{d0} & l_{d1} & \ldots & l_{d,\eta_i-1} \end{bmatrix} \ , \ D_d = \text{diagonal}\{d_{d0}, \ldots, d_{d,\eta_i-1}\}.$$

Then we can write, using the Schur reduction procedure (2.4),

$$\bar{D}_i = \bar{L}_d \bar{L}_d^* + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -I_{\eta_i} \end{bmatrix} \ , \tag{6.1d}$$

where $\bar{L}_d = L_d D_d^{-1/2}$. Comparing (6.1b) and (6.1d) we can easily conclude that $\bar{L}_d$ can be partitioned into a top lower triangular block equal to P_i^* and a lower block equal to Q_i, viz.,

$$\bar{L}_d = \begin{bmatrix} P_i^* \\ Q_i \end{bmatrix}.$$

ALGORITHM 6.1. *The QR factors Q_i and P_i^* can be computed in strongly regular steps as follows:*

1. *Apply the transformations $\{(\Theta_i, \Phi_i),\ i = 0, \ldots, \eta_i - 1\}$ that were applied to $\hat{H}_{\eta_i}$ to the block row $\begin{bmatrix} \mathbf{0} & \mathbf{0} & \hat{G}_i \end{bmatrix}$, and construct $\tilde{S}_i$ as in (6.1c).*
2. *Apply the last block row $\tilde{S}_i$ through the next η_i transformations $\{(\Theta_i, \Phi_i),\ i = \eta_i, \ldots, 2\eta_i - 1\}$ that were applied to $\hat{H}_{\eta_i,\eta_i}$ while checking the positive definiteness of $E_{\eta_i} E_{\eta_i}$. This determines the first η_i triangular factors of $\bar{D}_i$.*
3. *Partition $\bar{L}_d$ as shown above and read out Q_i and P_i^*.*

What about the choice $T_{\eta_i} \neq I$? In this case we need to consider the extended (also Toeplitz-like) matrix

$$\hat{D}_i = \begin{bmatrix} -T_{\eta_i} & D_i & \mathbf{0} \\ D_i & \mathbf{0} & D_i \\ \mathbf{0} & D_i & \mathbf{0} \end{bmatrix} \ ,$$

which still leads, after the first η_i recursive steps, to a generator for the matrix

$$\bar{D}_i = \begin{bmatrix} D_i T_{\eta_i}^{-1} D_i & D_i \\ D_i & \mathbf{0} \end{bmatrix} \ ,$$

and which is of the form

$$\bar{S}_i = \begin{bmatrix} \hat{H}_{\eta_i,\eta_i} \\ \tilde{S}_i \end{bmatrix}.$$

The point, however, is that the first η_i triangular factors of $\bar{D}_i$ now lead to a factorization of the form $D_i = Q_i P_i$, where P_i is still upper triangular but Q_i now satisfies $Q_i Q_i^* = T_{\eta_i}$. That is, Q_i is no longer a unitary matrix. But T_{η_i} is a positive-definite and structured matrix. Hence, its Cholesky factorization $T_{\eta_i} = \bar{L}_T \bar{L}_T^*$, can be efficiently evaluated in $O(\beta\eta_i^2)$ operations by using the strongly regular Algorithm 4.1. In this case, and following the argument in Section 4.3, we are instead led to a triangular factorization for R of the form $R = \hat{L}PQ\hat{L}^*$, where we now define $Q = Q_0 \oplus \ldots \oplus Q_{t-1}$, $P = P_0^{-*} \oplus \ldots \oplus P_{t-1}^{-*}$, $T = T_{\eta_0}^{-1} \oplus \ldots \oplus T_{\eta_{t-1}}^{-1}$, and $\hat{L} = LTQ$. The matrix $\hat{L}$ is still lower triangular with block columns of the form

$$\begin{bmatrix} P_i^* \\ W_i T_{\eta_i}^{-1} Q_i \end{bmatrix}.$$

The inverses $T_{\eta_i}^{-1}$ are not needed explicitly because, once we have the Cholesky factor of T_{η_i}, the products $T_{\eta_i}^{-1}Q_i$ can be computed by solving linear triangular systems. Also, the generator recursion has the same form as before (4.4a), viz.,

$$\begin{bmatrix} \mathbf{0}_{\eta_i \times r} \\ G_{i+1} \end{bmatrix} = G_i + X_i \ , \ \ X_i = (F_i - I_{n-\alpha_i})L_i D_i^{-1}(I_{\eta_i} - \hat{F}_i)^{-1}\hat{G}_i \ ,$$

and where X_i can now be rewritten as (compare with (4.5))

$$X_i = (F_i - I_{n-\alpha_i}) \begin{bmatrix} P_i^* \\ S_i T_{\eta_i}^{-1} Q_i \end{bmatrix} P_i^{-*}(I_{\eta_i} - \hat{F}_i)^{-1}\hat{G}_i.$$

7. System Interpretation. The generator recursions of Algorithms 4.1 and 4.2 have an interpretation as a cascade of linear state-space systems of orders $\{\eta_0, \eta_1, \ldots\}$. To clarify this, observe that the expressions for L_i and G_{i+1} in Theorem 3.1 can be combined together as follows

$$\begin{bmatrix} L_i & \mathbf{0} \\ & G_{i+1} \end{bmatrix} = \begin{bmatrix} F_i L_i & G_i \end{bmatrix} \begin{bmatrix} \hat{F}_i^* & \hat{H}_i^* J \\ J\hat{G}_i^* & J\hat{K}_i^* J \end{bmatrix}.$$

Hence, each recursive step involves an η_i−order discrete-time system that arises in state-space form on the right-hand side of the above expression, viz.,

$$\begin{bmatrix} \mathbf{x}_{j+1} & \mathbf{y}_j \end{bmatrix} = \begin{bmatrix} \mathbf{x}_j & \mathbf{w}_j \end{bmatrix} \begin{bmatrix} \hat{F}_i^* & \hat{H}_i^* J \\ J\hat{G}_i^* & J\hat{K}_i^* J \end{bmatrix} ,$$

where $\mathbf{x}_j$ is a $1 \times \eta_i$ state-vector and $\mathbf{w}_j$ and $\mathbf{y}_j$ are $1 \times r$ (row) input and output vectors, respectively, at time j. The above system matrix can also be regarded as a state-space realization of the inverse system

$$\begin{bmatrix} \hat{F}_i & \hat{G}_i \\ \hat{H}_i & \hat{K}_i \end{bmatrix}^{-1} ,$$

since it follows from the embedding relation (3.1b) that

$$\begin{bmatrix} \hat{F}_i & \hat{G}_i \\ \hat{H}_i & \hat{K}_i \end{bmatrix}^{-1} = \begin{bmatrix} D_i \hat{F}_i^* D_i^{-1} & D_i \hat{H}_i^* J \\ J\hat{G}_i^* D_i^{-1} & J\hat{K}_i^* J \end{bmatrix}.$$

The corresponding $r \times r$ transfer matrix $\Theta_i(z)$ is given by

$$\Theta_i(z) = J\hat{K}_i^* J + J\hat{G}_i^* \left[z^{-1} I_{\eta_i} - \hat{F}_i^* \right]^{-1} \hat{H}_i^* J.$$

It also follows from the embedding relation (3.1b) that $\Theta_i(z)$ satisfies the normalization condition $\Theta_i(z)J\Theta_i^*(z) = J$ on $|z| = 1$ and that, using (3.3), we can rewrite $\Theta_i(z)$ in the form

$$\Theta_i(z) = \left\{I - (1 - z\tau_i^*)J\hat{G}_i^*(I_{\eta_i} - z\hat{F}_i^*)^{-1}D_i^{-1}(I - \tau_i^*\hat{F}_i)^{-1}\hat{G}_i\right\}\Theta_i.$$

Therefore, t recursive steps lead to a cascade $\mathbf{\Theta}(z) = \Theta_0(z)\Theta_1(z)\ldots\Theta_{t-1}(z)$, which also satisfies $\mathbf{\Theta}(z)J\mathbf{\Theta}^*(z) = J$ on $|z| = 1$. In fact, we can further show that the cascade admits a state-space realization in terms of the original matrices F and G [24, 31].

THEOREM 7.1. *The cascade $\mathbf{\Theta}(z)$ admits an $n-$dimensional state-space description of the form*

$$\begin{bmatrix} \mathbf{x}_{j+1} & \mathbf{y}_j \end{bmatrix} = \begin{bmatrix} \mathbf{x}_j & \mathbf{w}_j \end{bmatrix} \begin{bmatrix} F^* & H^*J \\ JG^* & JK^*J \end{bmatrix},$$

where H and K are $r \times n$ and $r \times r$ matrices that satisfy the embedding relation

$$\begin{bmatrix} F & G \\ H & K \end{bmatrix} \begin{bmatrix} R & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} \begin{bmatrix} F & G \\ H & K \end{bmatrix}^* = \begin{bmatrix} R & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix}.$$

It also follows that the matrices H and K can be expressed in terms of $R, F,$ and G as follows

$$H = \mathbf{\Theta}^{-1}JG^*\left[I - \tau F^*\right]^{-1}R^{-1}(\tau I - F),$$
$$K = \mathbf{\Theta}^{-1}\left\{I - JG^*\left[I - \tau F^*\right]^{-1}R^{-1}G\right\},$$

and that $\mathbf{\Theta}(z) = \left\{I - (1 - z\tau^*)JG^*(I - zF^*)^{-1}R^{-1}(I - \tau^*F)^{-1}G\right\}\mathbf{\Theta}$, where τ is a unit-modulus scalar and $\mathbf{\Theta}$ is a $J-$unitary matrix.

8. Concluding Remarks. We derived a block Schur algorithm for the block triangular factorization of Hermitian Toeplitz-like matrices. We also provided tests for the determination of the sizes of the nonsingular minors in the exactly singular case. We also presented a system interpretation of the algorithm in terms of a cascade of elementary sections. We further remark that the results can be extended to non-Hermitian Toeplitz-like matrices, as well as Hankel-like matrices, and may be discussed elsewhere; though see [31].

Some issues deserve further consideration and may simplify the development of the algorithm. We have limited ourselves in the block case, for example, to the obvious choice $\Theta_i = I$. Other choices may be considered and could lead to an array form of the generator recursion (4.4a) in the same spirit as (4.3). Also, explicit tests for determining the sizes of the nonsingular minors in the general case of displacement ranks larger than two, along the lines of the special cases discussed in Section 5.2, deserve further investigation. These issues will be addressed elsewhere.

REFERENCES

[1] N. I. AKHIEZER, *The Classical Moment Problem and Some Related Questions in Analysis*, Hafner Publishing Company, NY, 1965.

[2] D. ALPAY AND H. DYM, *Structured invariant spaces of vector valued rational functions, Hermitian matrices, and a generalization of the Iohvidov laws*, Linear Algebra and Its Applications, 137–138 (1990), pp. 137–181.

[3] E. R. BERLEKAMP, *Algebraic Coding Theory*, McGraw-Hill, NY, 1968.
[4] R. E. BLAHUT, *Theory and Practice of Error Control Codes*, Addison Wesley Pub. Co., 1983.
[5] T. BOROS, A. H. SAYED, AND T. KAILATH, *Structured matrices and unconstrained rational interpolation*, Linear Algebra and Its Applications, 203–204 (1994), pp. 155–188.
[6] T. F. CHAN AND P. C. HANSEN, *A look-ahead Levinson algorithm for general Toeplitz systems*, IEEE Trans. on Signal Processing, 40 (1992), pp. 1079–1090.
[7] M. K. CILIZ AND H. KRISHNA, *Split Levinson algorithm for Toeplitz matrices with singular submatrices*, IEEE Transactions on Circuits and Systems, 36 (1989), pp. 922–924.
[8] T. CITRON, *Algorithms and Architectures for Error Correcting Codes*, PhD thesis, Stanford University, Stanford, CA, 1986.
[9] P. DELSARTE, Y. GENIN, AND Y. KAMP, *A generalization of the Levinson algorithm for Hermitian Toeplitz matrices with any rank profile*, IEEE Transactions on Acoustics, Speech and Signal Processing, 33 (1985), pp. 964–971.
[10] R. W. FREUND, *A look-ahead Bareiss algorithm for general Toeplitz matrices*, Numer. Math., (1994), pp. 35–69.
[11] F. R. GANTMACHER, *The Theory of Matrices*, Chelsea Publishing Company, NY, 1959.
[12] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, second ed., 1989.
[13] M. J. C. GOVER AND S. BARNETT, *Inversion of Toeplitz matrices which are not strongly nonsingular*, IMA Journal of Numerical Analysis, 5 (1985), pp. 101–110.
[14] M. GUTKNECHT, *Stable row recurrences for the Padé table and generically superfast lookahead solvers for non-Hermitian Toeplitz systems*, Linear Algebra and Its Applications, 188/189 (1993), pp. 351–422.
[15] G. HEINIG AND K. ROST, *Algebraic Methods for Toeplitz-like Matrices and Operators*, Akademie-Verlag, Berlin and Birkhäuser, 1984.
[16] I. S. IOHVIDOV, *Hankel and Toeplitz Matrices and Forms*, Birkhäuser, Boston, MA, 1982.
[17] T. KAILATH, *Linear Systems*, Prentice Hall, Englewood Cliffs, NJ, 1980.
[18] ———, *Signal processing applications of some moment problems*, in Moments in Mathematics, H. Landau, ed., American Mathematical Society, Providence, 1987, pp. 71–109.
[19] T. KAILATH AND J. CHUN, *Generalized displacement structure for block-Toeplitz, Toeplitz-block, and Toeplitz-derived matrices*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 114–128.
[20] T. KAILATH, S. Y. KUNG, AND M. MORF, *Displacement ranks of matrices and linear equations*, Journal of Mathematical Analysis and Applications, 68 (1979), pp. 395–407.
[21] T. KAILATH AND A. H. SAYED, *Displacement structure: Theory and applications*, SIAM Review. To appear.
[22] S. Y. KUNG, *Multivariable and Multidimensional Systems*, PhD thesis, Stanford University, Stanford, CA, June 1977.
[23] H. LEV-ARI AND T. KAILATH, *Triangular factorization of structured Hermitian matrices*, Operator Theory: Advances and Applications, 18 (1986), pp. 301–324.
[24] ———, *State-space approach to factorization of lossless transfer functions and structured matrices*, Linear Algebra and Its Applications, 162–164 (1992), pp. 273–295.
[25] J. L. MASSEY, *Shift-register synthesis and BCH decoding*, IEEE Transactions on Information Theory, 15 (1969), pp. 122–127.
[26] D. PAL, *Fast Algorithms for Structured Matrices with Arbitrary Rank Profile*, PhD thesis, Stanford University, Stanford, CA, May 1990.
[27] D. PAL AND T. KAILATH, *Fast triangular factorization of Hermitian Toeplitz and related matrices with arbitrary rank profile*, SIAM J. Matrix Anal. Appl. To appear.
[28] S. POMBRA, H. LEV-ARI, AND T. KAILATH, *Levinson and Schur algorithms for Toeplitz matrices with singular minors*, in Proc. IEEE Inter. Conf. on Acoustics, Speech and Signal Processing, NY, April 1988, pp. 1643–1646.
[29] Y. POTAPOV, *The multiplicative structure of J-contractive matrix functions*, Amer. Math. Soc. Transl., 15 (1960), pp. 131–244.
[30] C. M. RADER AND A. O. STEINHARDT, *Hyperbolic Householder transformations*, IEEE Transactions on Acoustics, Speech and Signal Processing, 34 (1986), pp. 1589–1602.
[31] A. H. SAYED, *Displacement Structure in Signal Processing and Mathematics*, PhD thesis, Stanford University, Stanford, CA, August 1992.
[32] A. H. SAYED, T. CONSTANTINESCU, AND T. KAILATH, *Square-Root Algorithms for Structured Matrices, Interpolation, and Completion Problems*, vol. 69 of IMA Volumes in Mathematics and Its Applications, Springer-Verlag, (1994).
[33] ———, *Time-variant displacement structure and interpolation problems*, IEEE Transactions on Automatic Control, 39 (1994), pp. 960–976.
[34] A. H. SAYED, T. KAILATH, H. LEV-ARI, AND T. CONSTANTINESCU, *Recursive solutions of ratio-*

nal interpolation problems via fast matrix factorization, Integral Equations and Operator Theory, 20 (1994), pp. 84–118.

[35] A. H. SAYED, H. LEV-ARI, AND T. KAILATH, *Time-variant displacement structure and triangular arrays*, IEEE Transactions on Signal Processing, 42 (1994), pp. 1052–1062.

[36] I. SCHUR, *Über potenzreihen die im Inneren des Einheitskreises beschränkt sind*, Journal für die Reine und Angewandte Mathematik, 147 (1917), pp. 205–232. (English translation in *Operator Theory: Advances and Applications*, vol. 18, pp. 31–88, edited by I. Gohberg, Birkhäuser, Boston, 1986).

[37] C. J. ZAROWSKI, *Schur algorithms for Hermitian Toeplitz, and Hankel matrices with singular leading principal submatrices*, IEEE Transactions on Signal Processing, 39 (1991), pp. 2464–2480.