Linear Algebra and Its Applications, vol. 219, pp. 49–78, Apr. 1995.

Fast Algorithms for Generalized Displacement Structures and Lossless Systems*

Ali H. Sayed† and Thomas Kailath

Abstract

We derive an efficient recursive procedure for the triangular factorization of strongly regular matrices with a generalized displacement structure that includes, as special cases, a variety of previously studied classes such as Toeplitz-like and Hankel-like matrices. The derivation is based on combining a simple Gaussian elimination procedure with displacement structure, and leads to a transmission-line interpretation in terms of two cascades of first-order sections. We further derive state-space realizations for each section and for the entire cascades, and show that these realizations satisfy a generalized embedding result and a generalized notion of J-losslessness. The cascades turn out to have intrinsic blocking properties, which can be shown to be equivalent to interpolation constraints.

*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.

†The work of this author was supported by a fellowship from Fundação de Amparo à Pesquisa do Estado de São Paulo while on leave from the Department of Electrical Engineering, Escola Politécnica da Universidade de São Paulo, Brazil. He is currently with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106.

1 Introduction

Matrices that exhibit certain structure, such as Toeplitz, Hankel, close-to-Toeplitz, or close-to-Hankel, often arise in application problems. The structure of these and other related classes of matrices is nicely captured by introducing the concept of displacement structure [1, 2, 3, 4]. In this context, an $n \times n$ non-Hermitian structured matrix R is compactly described by a pair of $n \times r$ matrices $\{G, B\}$ (called a generator pair of R) with $r \ll n$, and the column dimension of G or B is called the displacement rank of R. The triangular factorization of such strongly regular R (*i.e.,* a matrix with nonzero leading minors) can be computed efficiently and recursively in $O(rn^2)$ operations (additions and multiplications) [6, 7, 9]. This is achieved by appropriately combining Gaussian elimination with displacement structure. The resulting algorithm can be regarded as a far reaching generalization of an algorithm of Schur [10], which was chiefly concerned with the apparently very different problem of checking that a power series is analytic and bounded in the unit disc; hence the name generalized Schur algorithm.

The concept of displacement structure and structured matrices can be motivated by considering the much-studied special case of a symmetric Toeplitz matrix, $T = \left[c_{|i-j|}\right]_{i,j=0}^{n-1}$. 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 may be expected to be not much 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 et al. [2, 3] as

$$\nabla R \equiv R - ZRZ^* \quad , \tag{1}$$

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 (1), and r is referred to as the *displacement rank* of R. In this case, we can (nonuniquely) factor ∇R as

$$\nabla R = R - ZRZ^* = GJG^* \quad , \tag{2}$$

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. The pair $\{G, J\}$ is called a *generator* of R, since it contains all the information on R. In fact, we can write down an explicit and interesting representation for R in terms of the columns of its generator matrix G. Using the fact that Z is nilpotent, viz., $Z^n = \mathbf{0}$, we can readily conclude that the unique solution of (2) for a given $\{G, J\}$ is

$$R = \sum_{i=0}^{n-1} Z^i GJG^* Z^{*i}. \tag{3}$$

For a symmetric Toeplitz matrix $T = \left[c_{|i-j|}\right]_{i,j=0}^{n-1}$ with $c_0 = 1$, it is straightforward to verify that if we subtract ZTZ^* from T then we obtain

$$T - ZTZ^* = \begin{bmatrix} 1 & c_1 & \ldots & c_{n-1} \\ c_1 & & & \\ \vdots & & O & \\ c_{n-1} & & & \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ c_1 & c_1 \\ \vdots & \vdots \\ c_{n-1} & c_{n-1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ c_1 & c_1 \\ \vdots & \vdots \\ c_{n-1} & c_{n-1} \end{bmatrix}^* , \qquad (4)$$

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

The structure in (1) is convenient for Toeplitz and related matrices. References [2, 3] also noted the possibility of other definitions of displacement appropriate for Hankel-type matrices, periodic matrices, etc. Independently, Heinig and Rost [11] studied the properties and applications of definitions such as $\nabla R = ZR - RZ^*$, and more generally $\nabla R = UR - RV$ for suitable matrices $\{U, V\}$. This definition is especially appropriate for Hankel (and Hankel-derived) matrices. For example, if $H = [h_{i+j}]_{i,j=0}^{n-1}$ is a Hankel matrix, we can see that

$$\nabla H = ZH - HZ^* = \begin{bmatrix} 0 & -h_0 & -h_1 & \ldots & -h_{n-2} \\ h_0 & & & & \\ h_1 & & O & & \\ \vdots & & & & \\ h_{n-2} & & & & \end{bmatrix} \text{ has rank 2.} \qquad (5)$$

Note however that H cannot be recovered from its displacement ∇H, because the entries h_{n-1}, $\ldots$, h_{2n-2} do not appear in ∇H; this "difficulty" can be fixed in various ways (see, *e.g.*, [8, 11, 12] and the brief discussion in Section 6.2 ahead).

Because of the famous Levinson algorithm [13] for solving Toeplitz (or normal) equations (or equivalently making a triangular factorization of the inverse of the Toeplitz matrix), many later studies of fast factorization algorithms focused on the derivation of Levinson-like recursions for solving

$$Rx = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}^T ,$$

for a column vector x, and for more general matrices R. The Levinson algorithm was also used by Bareiss [14], LeRoux-Gueguen [15], Morf [16], and Rissanen [17] to find fast algorithms for the triangular factorization of the Toeplitz matrix itself (rather than its inverse). It was noted by Dewilde, Vieira, and Kailath [18] that an algorithm of Schur [10] was in fact the first to *efficiently* solve the direct factorization problem for Toeplitz matrices. Lev-Ari and Kailath [7] later noted that the Schur algorithm could be extended to solve the factorization problem for a larger class of Hermitian matrices R that have displacement structure of the form

$$\nabla R = \sum_{i,j=0}^{\infty} d_{ij} Z^i R Z^{*j} ,$$

if, and only, if the bivariate polynomial

$$d(z, w) = \sum_{i,j=0}^{\infty} d_{ij} z^i w^{*j}$$

admits the representation

$$d(z,w) = \alpha(z)\alpha^*(w) - \beta(z)\beta^*(w) \ ,$$

for certain polynomials $\{\alpha(z), \beta(z)\}$. This generalization includes as special cases the Toeplitz structure, which corresponds to $d(z,w) = (1 - zw^*)$, the Hankel structure, which corresponds to $d(z,w) = j(z - w^*)$ (where $j^2 = -1$), and Toeplitz-plus Hankel structure, which corresponds to $d(z,w) = j(z - w^*)(1 - zw^*)$ (see, *e.g.,* [7, 19]).

Later Chun and Kailath [8, 9, 20] pointed out that the Schur algorithm could be used to solve not only direct factorization problems but also inversion problems and QR factorization problems, by studying suitably defined block matrices and using more general definitions of displacement structure:

$$\nabla R = R - FRF^* \ , \ \ \nabla R = R - FRA^* \ ,$$

$$\nabla R = FR - RF^* \ , \ \ \nabla R = FR - RA^* \ ,$$

with

$$\{F, A\} \ \ \textit{strictly} \text{ lower triangular.}$$

However, the case of lower triangular (or even diagonal) matrices F and A is important in many applications, such as interpolation problems [24, 25, 26]. Lev-Ari and Kailath [5, 27] studied the lower triangular case. In [27] they described a recursive state-space procedure for the factorization of Hermitian positive-definite matrices with a Toeplitz-like (or close-to-Toeplitz) structure of the form

$$\nabla R = R - FRF^* \equiv GJG^* \ ,$$

where F is a stable (*i.e.,* all eigenvalues are strictly inside the unit disc) lower triangular matrix, G is an $n \times r$ matrix that is called the generator of R, and J is a signature matrix of the form $J =$ diagonal $\{I_p, \ -I_q\}$ $(p + q = r)$. The recursive procedure was derived by embedding $\{F, G\}$ into a J-lossless state-space system matrix

$$\begin{bmatrix} F & G \\ H & K \end{bmatrix}$$

that satisfies the (discrete-time) embedding or dilation relation

$$\begin{bmatrix} F & G \\ H & K \end{bmatrix} \begin{bmatrix} R & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} \begin{bmatrix} F & G \\ H & K \end{bmatrix}^* = \begin{bmatrix} R & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} \ , \tag{6}$$

and then defining a recursive cascade decomposition of this system matrix into a product of first-order J-lossless system matrices.

In this paper we simplify this general line of argument to obtain a unified derivation for the study of the previous classes of structured matrices and extensions thereof. In particular, we develop a recursive matrix-based derivation of a generalized factorization algorithm, which applies equally well to strongly regular and to non Hermitian structured matrices. For this purpose, we consider strongly regular matrices R with a generalized displacement structure of the form

$$\nabla R = \Omega R \Delta^* - F R A^* \ ,$$

where $\{\Omega, \Delta, F, A\}$ are lower triangular matrices. R is said to be structured if ∇R has low rank, independent of n. We then derive a generalized Schur algorithm that efficiently computes the triangular factorization of R. The derivation also leads to an extension of the embedding relation (6) to the non Hermitian case – see (22) and (33) ahead.

The factorization algorithm is derived by combining a simple Gaussian elimination (or Jacobi reduction) procedure with the generalized (displacement) structure. To further emphasize this point, we should mention here that the Gaussian elimination procedure factors an arbitrary matrix, but with $O(n^3)$ elementary computations for an $n \times n$ matrix. The fact that R has further (displacement) structure allows us to show that the Gaussian procedure can be reduced to a pair of generalized (Schur) recursions. The resulting algorithm works only with the entries of the generator matrices $\{G, B\}$ and the defining matrices $\{\Omega, \Delta, F, A\}$. It does not use the entries of R, which is in contrast to the Gaussian procedure. In some applications, for instance, such as interpolation problems (see, *e.g.,* [24, 26, 28]), we do not even know the matrix itself. All we know is that R satisfies a special (displacement) equation and all we are given are the generator matrices as well as the displacement form. In such cases, computing the triangular factors of R via the Gaussian elimination procedure would require us to first solve for R. On the other hand, the recursive algorithm presented here avoids this step and uses only the generator matrices and the quantities that define the displacement of R; the matrix itself is not needed.

The paper is organized as follows. In Section 2, we introduce the class of matrices with generalized displacement structure. In Section 3 we describe the classical Gaussian (often also called Jacobi or Schur) reduction procedure for the triangular factorization of strongly regular matrices. In Section 4, we exploit the underlying structure and show that the Gauss reduction collapses to an efficient generator recursion. We also remark the appearance of a generalized embedding relation. In Section 5, we discuss some of the implications of this general embedding result. We show that the generator recursion leads to a cascade of first-order sections that satisfy a (non-Hermitian) J-losslessness relation. We may mention that these cascades turn out to have intrinsic interpolation properties, which can be advantageously used in the solution of several types of unconstrained interpolation problems such as Padé approximation, Lagrange interpolation, and others (see, *e.g.,* [28]). In Section 6, we derive the generalized Schur algorithm by further simplifying the generator recursions of Section 4, and elaborate on some computational and uniqueness issues. In Section 7, we derive a state-space realization for the entire cascade of first-order sections, and show that it also satisfies a general embedding result; Section 8 has some concluding remarks.

1.1 Direct Factorization Problems

Before proceeding further, we would like to stress the fact that direct factorization is, in many respects, more fundamental than the inverse problem. To illustrate this point we consider several examples that motivate the need for more general structures such as $R - FRF^*$ rather than $R - ZRZ^*$. The arguments that follow are based on an embedding technique of Chun and Kailath [8, 9], which shows how to exploit the freedom in choosing F to great effect.

Consider again the case of an $n \times n$ symmetric Toeplitz matrix T for which $T - Z_n T Z_n^*$ has rank 2 (Z_n denotes the $n \times n$ lower shift matrix). If we form the block matrix

$$M = \begin{bmatrix} -T & I \\ I & \mathbf{0} \end{bmatrix} \ , \tag{7}$$

then it is 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,

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

which corresponds to choosing $F = Z \oplus Z$ 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 the generator matrix G of a structured matrix (say the generator of M) and at each step it provides us with the generator of the successive Schur complements of the matrix. So the first step of the algorithm gives us G_1, which is the generator of the Schur complement of M with respect to its $(0,0)$ entry. The next step gives us G_2, which is the generator of the Schur complement of M with respect to its 2×2 leading submatrix, and so on. After n such steps, we clearly obtain the generator of the n^{th} Schur complement, which is T^{-1}. The generator of T^{-1} can then be used to efficiently determine the triangular factorization of the inverse matrix.

Hence, by performing the *direct* factorization of the extended matrix M in (7) 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. As a second example, we consider the product of two symmetric Toeplitz matrices $T_1 T_2$,

$$T_1 = \left[c_{|i-j|} \right]_{i,j=0}^{n-1}, \quad T_2 = \left[e_{|i-j|} \right]_{i,j=0}^{n-1}, \quad \text{with } c_0 = e_0 = 1 ,$$

and note that $T_1 T_2$ is the Schur complement of the $(0,0)$ block entry in the extended matrix M defined by

$$M = \begin{bmatrix} -I & T_2 \\ T_1 & \mathbf{0} \end{bmatrix} .$$

It is now easy to verify that

$$\nabla M = M - \begin{bmatrix} Z & \mathbf{0} \\ \mathbf{0} & Z \end{bmatrix} M \begin{bmatrix} Z & \mathbf{0} \\ \mathbf{0} & Z \end{bmatrix}^* = \left[\begin{array}{cccccccc} -1 & 0 & \cdots & 0 & 1 & e_1 & \cdots & e_{n-1} \\ 0 & & & & e_1 & & & \\ \vdots & & \text{O} & & \vdots & & \text{O} & \\ 0 & & & & e_{n-1} & & & \\ 1 & c_1 & \cdots & c_{n-1} & 0 & 0 & \cdots & 0 \\ c_1 & & & & 0 & & & \\ \vdots & & \text{O} & & \vdots & & \text{O} & \\ c_{n-1} & & & & 0 & & & \end{array} \right] ,$$

which shows that M has displacement rank 4 with respect to the displacement operation $M - (Z \oplus Z) M (Z \oplus Z)^*$. The (generalized) Schur algorithm can then be used to determine a generator for $T_1 T_2$ and consequently, it triangular factorization. Several other examples can be found in [8, 9]. Applications with more general matrices F (*i.e.,* not necessarily strictly lower triangular) include interpolation problems [24, 26, 28] and adaptive filtering [21, 26].

2 Generalized Displacement Structure

We consider an arbitrary $n \times n$ strongly regular (*i.e.,* all leading principal minors are nonzero) matrix R that satisfies a generalized displacement equation of the form

$$\Omega R \Delta^* - FRA^* = GJB^* \ , \tag{8}$$

where Ω, Δ, F, and A are $n \times n$ lower triangular matrices whose diagonal entries will be denoted by $\{\omega_i\}_{i=0}^{n-1}$, $\{\delta_i\}_{i=0}^{n-1}$, $\{f_i\}_{i=0}^{n-1}$, and $\{a_i\}_{i=0}^{n-1}$, respectively, G and B are $n \times r$ so-called generator matrices (with $r \leq n$), and J is an $r \times r$ signature matrix that satisfies $J^2 = I$, such as $J = I$,

$$J = \begin{bmatrix} I_p & \mathbf{0} \\ \mathbf{0} & -I_q \end{bmatrix}, \quad p + q = r \ ,$$

or some other possible form. We assume that

$$\omega_j \delta_i^* - f_j a_i^* \neq 0 \quad \text{for all} \quad i, j. \tag{9}$$

This guarantees the existence of a unique solution R of the displacement equation (8) since we can then solve directly for the entries $\{r_{mj}\}$ of R. For example, the entries of the first row of R satisfy

$$\omega_0 \begin{bmatrix} r_{00} & r_{01} & \ldots & r_{0,n-1} \end{bmatrix} \Delta^* - f_0 \begin{bmatrix} r_{00} & r_{01} & \ldots & r_{0,n-1} \end{bmatrix} A^* = g_0 J B^* \ ,$$

where g_0 denotes the first row of G, and hence can be determined uniquely. Once the entries of the first row of R are determined, we can now repeat the argument for the second row, and so on[1]. Alternatively, it will be clear later that the the above conditions allow us to uniquely determine the triangular factors of R, and consequently R itself. We further note that algorithms often need to be developed in circumstances where the nonuniqueness condition does not hold; procedures for such cases are briefly addressed in Section 6.2.

We shall say that R has a generalized displacement structure with respect to $\{\Omega, \Delta, F, A\}$, and $\{G, B\}$ will be called a generator pair of R. The motivation for calling R a structured matrix stems from the fact that Toeplitz and Hankel matrices satisfy special cases of (8), as remarked in the introductory section. Another interesting example is the case of a symmetric Toeplitz plus Hankel matrix [19], viz.,

$$R = T + H \equiv \begin{bmatrix} c_{|i-j|} + h_{i+j} \end{bmatrix}_{i,j=0}^{\infty} \ , \quad \text{with} \quad c_0 = 1.$$

It is straightforward to verify that the difference

$$ZR(I + Z^2)^* - (I + Z^2)RZ^*$$

has rank 4, viz.,

$$ZR(I + Z^2)^* - (I + Z^2)RZ^* = GJG^* \ , \tag{10}$$

where

$$J = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad G = \begin{bmatrix} 1 & 0 & \frac{1}{2} & 0 \\ 0 & 1 & c_1 & \frac{1}{2} + h_0 \\ 0 & 0 & c_2 + h_0 & c_1 + h_1 \\ 0 & 0 & c_3 + h_1 & c_2 + h_2 \\ 0 & 0 & c_4 + h_2 & c_3 + h_3 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} .$$

[1] The authors would like to thank a referee for pointing out corrections to our original argument, and Prof. G. W. Stewart for suggesting this alternative proof.

Therefore, a symmetric Toeplitz plus Hankel matrix has displacement rank 4, and (10) is a special case of (8) with

$$\Omega = A = Z, \quad \Delta = F = I + Z^2, \quad r = 4, \quad G \text{ and } J \text{ as above.}$$

The displacement operators $\{Z, I + Z^2, I + Z^2, Z\}$ do not satisfy the uniqueness condition (9). However, this can be properly handled as discussed in [19] (see also the discussion in Section 6.3).

We further remark that (8) includes more general structured matrices such as Toeplitz-like (*e.g.,* quasi-Toeplitz, block-Toeplitz, Toeplitz-block) and Hankel-like (*e.g.,* quasi-Hankel, block-Hankel, Hankel-block, Vandermonde): Toeplitz-like matrices correspond to $\Omega = \Delta = I$, while Hankel-like matrices correspond to $\Delta = F = I$.

3 Gaussian Elimination

The assumption of strong regularity of $R = [r_{mj}]_{m,j=0}^{n-1}$ guarantees the existence of a triangular factorization of the form [29]

$$R = LD^{-1}U \ ,$$

where D is diagonal and L and U are lower- and upper-triangular matrices, respectively. The columns of L and U, and the diagonal entries of D, can be computed recursively via the Gauss (also often called Jacobi [22] and Schur [10]) reduction procedure. Let l_0, u_0^*, and d_0 denote the first column, the first row, and the $(0,0)$ entry of R, respectively:

$$d_0 = r_{00}, \quad l_0 = \begin{bmatrix} r_{00} \\ r_{10} \\ \vdots \\ r_{n-1,0} \end{bmatrix}, \quad u_0^* = \begin{bmatrix} r_{00} & r_{01} & \dots & r_{0,n-1} \end{bmatrix}.$$

Then

$$R - l_0 d_0^{-1} u_0^* = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & R_1 & \\ 0 & & & \end{bmatrix} \equiv \tilde{R}_1 \ , \tag{11}$$

where $R_1 = \left[r_{mj}^{(1)} \right]_{m,j=0}^{n-2}$ is called the Schur complement of r_{00} in R. Expression (11) represents one reduction step, and it can now be repeated in order to compute the Schur complement $R_2 = \left[r_{mj}^{(2)} \right]_{m,j=0}^{n-3}$ of $r_{00}^{(1)}$ in R_1, and so on[2]. This suggests the following recursive procedure:

$$\tilde{R}_{i+1} = \tilde{R}_i - \tilde{l}_i d_i^{-1} \tilde{u}_i^*, \quad \tilde{l}_0 = l_0, \quad \tilde{u}_0^* = u_0^*, \quad \tilde{R}_0 = R \ , \tag{12}$$

where $d_i = r_{00}^{(i)}$, and $\tilde{l}_i$ and $\tilde{u}_i^*$ denote the i^{th} column and row of $\tilde{R}_i$, respectively. Notice that the first i entries of $\tilde{l}_i$ and $\tilde{u}_i^*$ are all zero, and we shall denote the corresponding nonzero parts by l_i and u_i^*, respectively, viz.,

[2] Because of this terminology, we had dubbed this the Schur reduction procedure in [5, 7] and in later papers. G. W. Stewart reminded us that this was just Gaussian elimination; also that for nonsymmetric matrices, the *LU* decomposition should be credited to Jacobi in 1857.

$$\tilde{l}_i = \begin{bmatrix} \mathbf{0}_{i\times 1} \\ l_i \end{bmatrix}, \quad \tilde{u}_i = \begin{bmatrix} \mathbf{0}_{i\times 1} \\ u_i \end{bmatrix}.$$

That is, l_i and u_i^* are the first column and row of the i^{th} Schur complement R_i. We can also rewrite each reduction step (12) in the following alternative form

$$R_i = \begin{bmatrix} l_i & \begin{matrix} \mathbf{0} \\ I_{n-i-1} \end{matrix} \end{bmatrix} \begin{bmatrix} d_i^{-1} & \mathbf{0} \\ \mathbf{0} & R_{i+1} \end{bmatrix} \begin{bmatrix} u_i & \begin{matrix} \mathbf{0} \\ I_{n-i-1} \end{matrix} \end{bmatrix}^* . \tag{13}$$

It follows from (12) that R can be expressed as the sum of n rank 1 terms (since $R_n = 0$),

$$R = \sum_{i=0}^{n-1} \tilde{l}_i d_i^{-1} \tilde{u}_i^*.$$

Therefore, the triangular factors L, U, and D are given by

$$L = \begin{bmatrix} \tilde{l}_0 & \tilde{l}_1 & \ldots & \tilde{l}_{n-1} \end{bmatrix} \ , \ U = \begin{bmatrix} \tilde{u}_0 & \tilde{u}_1 & \ldots & \tilde{u}_{n-1} \end{bmatrix}^* \ ,$$

$$D = \text{diagonal } \{d_0, d_1, \ldots, d_{n-1}\}.$$

Observe that the reduction procedure (12) is a recursive algorithm that operates directly on the entries of the successive Schur complements R_i. This clearly requires $O(n^3)$ operations (additions and multiplications) for an arbitrary strongly-regular matrix R. However, the computational complexity can be reduced to $O(rn^2)$ in the case of structured matrices as in (8) (with some additional constraints on the displacement operators $\{\Omega, \Delta, F, A\}$, as discussed in Section 6.2). For such structured matrices, we can replace (12) with the so-called *generator recursions* that operate directly on the elements of the generator matrices G and B, which have rn elements each as compared to n^2 in R. For example, in the Toeplitz and Toeplitz-plus-Hankel cases we have $r = 2$ and $r = 4$ respectively, independent of n.

4 Generalized Generator Recursions

We now verify that the successive Schur complements R_i also exhibit generalized displacement structure. We shall assume from now on that Ω and Δ are invertible matrices (by symmetry, the same argument holds for invertible F and A). If instead Ω and A (or Δ and F) were invertible then a continuous-time analogue of the embedding technique used ahead can be invoked to derive the appropriate recursions - we shall for simplicity and brevity forgo the details here [23] and focus on the discrete-time case. Also, in Section 6.2 and in [19] we briefly illustrate with examples, how to handle cases where the nonuniqueness or invertibility conditions are violated.

We start with the first Schur complement R_1 defined by (11). It readily follows from the displacement equation (8) that the first column l_0 and the first row u_0^* of R satisfy the following relations (due to the lower triangularity of the displacement operators $\{\Omega, \Delta, F, A\}$):

$$\begin{aligned} \Omega l_0 \delta_0^* &= F l_0 a_0^* + G J b_0^* \ , \\ \omega_0 u_0^* \Delta^* &= f_0 u_0^* A^* + g_0 J B^* \ , \\ d_0 &= \frac{g_0 J b_0^*}{\omega_0 \delta_0^* - f_0 a_0^*} \ , \end{aligned} \tag{14}$$

where g_0 and b_0 denote the first rows of G and B, respectively. That is, given G, B, and the displacement operators $\{\Omega, \Delta, F, A\}$, we can write down explicit expressions for the first triangular factors l_0 and u_0^*,

$$l_0 = (\delta_0^*\Omega - a_0^*F)^{-1}GJb_0^* \ ,$$
$$u_0 = (\omega_0^*\Delta - f_0^*A)^{-1}BJg_0^*.$$

Let $\{\Omega_1, \Delta_1, F_1, A_1\}$ be the submatrices obtained after deleting the first row and column of $\{\Omega, \Delta, F, A\}$,

$$F = \begin{bmatrix} f_0 & \mathbf{0} \\ ? & F_1 \end{bmatrix} \ , \quad A = \begin{bmatrix} a_0 & \mathbf{0} \\ ? & A_1 \end{bmatrix} \ ,$$

$$\Omega = \begin{bmatrix} \omega_0 & \mathbf{0} \\ ? & \Omega_1 \end{bmatrix} \ , \quad \Delta = \begin{bmatrix} \delta_0 & \mathbf{0} \\ ? & \Delta_1 \end{bmatrix} \ ,$$

where ? denotes irrelevant entries. Using (11) and (14) we can prove the following result.

Fact 1 (Structure of R_1) *The first Schur complement R_1 is also structured with generator matrices G_1 and B_1, viz.,*

$$\Omega_1 R_1 \Delta_1^* - F_1 R_1 A_1^* = G_1 J B_1^* \ ,$$

where G_1 and B_1 are $(n-1) \times r$ matrices that are computed from G and B as follows:

$$\begin{aligned} \begin{bmatrix} \mathbf{0} \\ G_1 \end{bmatrix} &= F l_0 c_0^* J + G J s_0^* J \ , \\ \begin{bmatrix} \mathbf{0} \\ B_1 \end{bmatrix} &= A u_0 h_0^* J + B J k_0^* J \ , \end{aligned} \tag{15}$$

where c_0 and h_0 are arbitrary $r \times 1$ column vectors, and s_0 and k_0 are arbitrary $r \times r$ matrices chosen so as to satisfy the generalized embedding relation

$$\begin{bmatrix} f_0 & g_0 \\ h_0 & k_0 \end{bmatrix} \begin{bmatrix} d_0 & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} \begin{bmatrix} a_0 & b_0 \\ c_0 & s_0 \end{bmatrix}^* = \begin{bmatrix} \omega_0 d_0 \delta_0^* & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix}. \tag{16}$$

Proof: The proof can be obtained via direct manipulations of (11) and (14). Using (11) we write

$$\begin{aligned} \Omega \tilde{R}_1 \Delta^* - F \tilde{R}_1 A^* \ = \ & GJ \left\{ J - \frac{b_0^* g_0}{\omega_0 d_0 \delta_0^*} \right\} JB^* - \\ & GJ \ \frac{f_0 b_0^*}{\omega_0 d_0 \delta_0^*} \ u_0^* A^* - F l_0 \ \frac{a_0^* g_0}{\omega_0 d_0 \delta_0^*} \ JB^* + \\ & F l_0 \ \frac{g_0 J b_0^*}{\omega_0 d_0^2 \delta_0^*} \ u_0^* A^*. \end{aligned} \tag{17}$$

We now verify that the right-hand side can be put into the form of a *perfect square* by introducing some auxiliary quantities. Consider $r \times r$ matrices k_0 and s_0 and $r \times 1$ column vectors h_0 and c_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)

$$\begin{array}{ll} s_0^* J k_0 = J - \frac{b_0^* g_0}{\omega_0 d_0 \delta_0^*} \quad , & s_0^* J h_0 = - \frac{f_0 b_0^*}{\omega_0 d_0 \delta_0^*} \quad , \\ & \\ c_0^* J k_0 = - \frac{a_0^* g_0}{\omega_0 d_0 \delta_0^*} \quad , & c_0^* J h_0 = \frac{g_0 J b_0^*}{\omega_0 d_0^2 \delta_0^*} \ . \end{array} \tag{18}$$

Using $\{c_0, s_0, h_0, k_0\}$ we can rewrite the right-hand side of (17) in the form

$$GJs_0^*Jk_0JB^* + GJs_0^*Jh_0u_0^*A^* + Fl_0c_0^*Jk_0JB^* + Fl_0c_0^*Jh_0u_0^*A^*,$$

which can be clearly factored as $\tilde{G}_1 J \tilde{B}_1^*$ where

$$\tilde{G}_1 = Fl_0c_0^*J + GJs_0^*J \quad \text{and} \quad \tilde{B}_1 = Au_0h_0^*J + BJk_0^*J.$$

Recall that the first row and column of $\tilde{R}_1$ are zero. Hence, the first rows of $\tilde{G}_1$ and $\tilde{B}_1$ are zero

$$\tilde{G}_1 = \begin{bmatrix} \mathbf{0} \\ G_1 \end{bmatrix} \quad , \quad \tilde{B}_1 = \begin{bmatrix} \mathbf{0} \\ B_1 \end{bmatrix} .$$

Finally, it follows from the definitions in (18) (and from the expression for d_0) that $\{c_0, s_0, h_0, k_0\}$ satisfy

$$\begin{bmatrix} a_0 & b_0 \\ c_0 & s_0 \end{bmatrix}^* \begin{bmatrix} (\omega_0 d_0 \delta_0^*)^{-1} & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} \begin{bmatrix} f_0 & g_0 \\ h_0 & k_0 \end{bmatrix} = \begin{bmatrix} d_0^{-1} & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} ,$$

which leads to (16).

■

We shall later give explicit expressions for $\{c_0, s_0, h_0, k_0\}$ that satisfy (18), in terms of the known quantities $\{f_0, a_0, \delta_0, \omega_0, g_0, b_0\}$. Meanwhile, observe that the generator recursions (15) and the expressions for the triangular factors in (14) can be combined and rewritten compactly as follows

$$\begin{aligned} \begin{bmatrix} \delta_0^* \Omega l_0 & \mathbf{0} \\ & G_1 \end{bmatrix} &= \begin{bmatrix} Fl_0 & G \end{bmatrix} \begin{bmatrix} a_0^* & c_0^* J \\ Jb_0^* & Js_0^* J \end{bmatrix} , \\ \begin{bmatrix} \omega_0^* \Delta u_0 & \mathbf{0} \\ & B_1 \end{bmatrix} &= \begin{bmatrix} Au_0 & B \end{bmatrix} \begin{bmatrix} f_0^* & h_0^* J \\ Jg_0^* & Jk_0^* J \end{bmatrix} . \end{aligned} \tag{19}$$

It is clear that the previous discussion can be repeated for the higher order Schur complements R_i $(i \geq 2)$, leading to the following theorem.

Theorem 1 (Structure of the Schur Complements) *The Schur complements R_i satisfy the displacement equation*

$$\Omega_i R_i \Delta_i^* - F_i R_i A_i^* = G_i J B_i^* \ , \tag{20}$$

where G_i and B_i are $(n-i) \times r$ generator matrices that satisfy, along with the triangular factors l_i and u_i, the following recursions

$$\begin{aligned} \begin{bmatrix} \delta_i^* \Omega_i l_i & \mathbf{0} \\ & G_{i+1} \end{bmatrix} &= \begin{bmatrix} F_i l_i & G_i \end{bmatrix} \begin{bmatrix} a_i^* & c_i^* J \\ Jb_i^* & Js_i^* J \end{bmatrix} \ , \ G_0 = G \ , \\ \begin{bmatrix} \omega_i^* \Delta_i u_i & \mathbf{0} \\ & B_{i+1} \end{bmatrix} &= \begin{bmatrix} A_i u_i & B_i \end{bmatrix} \begin{bmatrix} f_i^* & h_i^* J \\ Jg_i^* & Jk_i^* J \end{bmatrix} \ , \ B_0 = B \ , \end{aligned} \tag{21}$$

where c_i and h_i are arbitrary $r \times 1$ column vectors, and s_i and k_i are arbitrary $r \times r$ matrices chosen so as to satisfy the generalized embedding relation

$$\begin{bmatrix} f_i & g_i \\ h_i & k_i \end{bmatrix} \begin{bmatrix} d_i & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} \begin{bmatrix} a_i & b_i \\ c_i & s_i \end{bmatrix}^* = \begin{bmatrix} \omega_i d_i \delta_i^* & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} , \tag{22}$$

with g_i and b_i denoting the first rows of G_i and B_i, respectively,

$$d_i = r_{00}^{(i)} = \frac{g_i J b_i^*}{\omega_i \delta_i^* - f_i a_i^*} ,$$

and $\{\Omega_i, \Delta_i, F_i, A_i\}$ are the submatrices obtained after deleting the first row and column of the corresponding matrices $\{\Omega_{i-1}, \Delta_{i-1}, F_{i-1}, A_{i-1}\}$.

■

5 Generalized Embedding Relations

Observe that each generator recursion involves two first-order discrete-time systems (in state-space form) that appear on the right-hand side of (21). We now verify that these systems satisfy a generalized losslessness relation. We remark that (22) is a generalization and an extension to the non-Hermitian case, of the now well-known Hermitian embedding relation for $J-$lossless systems (see, *e.g.,* [27, 31] and expression (6) in the introductory section).

Consider the generalized embedding relation (22), and introduce the two first-order discrete-time state-space models that arise in the generator recursion (21), viz.,

$$\begin{bmatrix} \delta_i^* \mathbf{x}_{j+1}^{(1)} & \mathbf{y}_j^{(1)} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_j^{(1)} & \mathbf{w}_j^{(1)} \end{bmatrix} \begin{bmatrix} a_i^* & c_i^* J \\ J b_i^* & J s_i^* J \end{bmatrix} , \tag{23}$$

$$\begin{bmatrix} \omega_i^* \mathbf{x}_{j+1}^{(2)} & \mathbf{y}_j^{(2)} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_j^{(2)} & \mathbf{w}_j^{(2)} \end{bmatrix} \begin{bmatrix} f_i^* & h_i^* J \\ J g_i^* & J k_i^* J \end{bmatrix} , \tag{24}$$

where $\mathbf{x}_j^{(1,2)}$ denote the so-called state variables, and $\mathbf{w}_j^{(1,2)}$ denote $1 \times r$ row input vectors at time j (that is, the input vectors are from the left). Let $\Theta_i(z)$ and $\Gamma_i(z)$ denote the corresponding $r \times r$ (generalized) transfer matrices, viz.,

$$\begin{aligned} \Theta_i(z) &= J s_i^* J + J b_i^* \left[z^{-1} \delta_i^* - a_i^* \right]^{-1} c_i^* J \ , \\ \Gamma_i(z) &= J k_i^* J + J g_i^* \left[z^{-1} \omega_i^* - f_i^* \right]^{-1} h_i^* J. \end{aligned} \tag{25}$$

Using the embedding relation (22) we readily conclude that the first-order sections $\Theta_i(z)$ and $\Gamma_i(z)$ satisfy the generalized $J-$losslessness (or normalization) relation

$$\Gamma_i(z) J \Theta_i^*(w) = J \ \text{ on } \ zw^* = 1. \tag{26}$$

5.1 First-Order Sections

The above discussion clearly shows that we can associate with each embedding relation (22) two transfer matrices $\Gamma_i(z)$ and $\Theta_i(z)$ that satisfy (26). We now verify that (22) completely specify $\{c_i, s_i, h_i, k_i\}$ in terms of the known quantities.

Lemma 1 ($\{h_i, k_i, c_i, s_i\}$) *All choices of h_i, k_i, c_i, and s_i are given by*

$$h_i = \Theta_i^{-1}\left\{\frac{1}{d_i\omega_i}\frac{\mu_i\omega_i - f_i}{(\delta_i^* - \mu_i a_i^*)}Jb_i^*\right\} \quad , \quad k_i = \Theta_i^{-1}\left\{I_r - \frac{1}{d_i\omega_i}\frac{Jb_i^* g_i}{(\delta_i^* - \mu_i a_i^*)}\right\} \quad ,$$

$$c_i = \Gamma_i^{-1}\left\{\frac{1}{d_i^*\delta_i}\frac{\tau_i\delta_i - a_i}{(\omega_i^* - \tau_i f_i^*)}Jg_i^*\right\} \quad , \quad s_i = \Gamma_i^{-1}\left\{I_r - \frac{1}{d_i^*\delta_i}\frac{Jg_i^* b_i}{(\omega_i^* - \tau_i f_i^*)}\right\} \quad ,$$

for arbitrary constant matrices Θ_i and Γ_i satisfying $\Theta_i J\Gamma_i^* = J$, and arbitrary scalars τ_i and μ_i such that $\tau_i\mu_i^* = 1$.

Proof: The proof is patterned on one in [27]. Let τ_i and μ_i be two arbitrary scalars on the curve $\tau_i\mu_i^* = 1$. We first show how to choose pairs $(\hat{h}_i, \hat{k}_i)$ and $(\hat{c}_i, \hat{s}_i)$ such that the corresponding transfer matrices $\hat{\Gamma}_i(z)$ and $\hat{\Theta}_i(z)$ (as in (25)) satisfy $\hat{\Gamma}_i(\mu_i^{-*}) = I_r$ and $\hat{\Theta}_i(\tau_i^{-*}) = I_r$, or equivalently,

$$\hat{s}_i + \hat{c}_i(\tau_i\delta_i - a_i)^{-1}b_i = I_r \quad \text{and} \quad \hat{k}_i + \hat{h}_i(\mu_i\omega_i - f_i)^{-1}g_i = I_r.$$

But (22) implies that $\hat{c}_i d_i^* f_i^* + \hat{s}_i Jg_i^* = \mathbf{0}$ and $\hat{h}_i d_i a_i^* + \hat{k}_i Jb_i^* = \mathbf{0}$. Therefore,

$$\hat{h}_i = \frac{1}{d_i\omega_i}\frac{\mu_i\omega_i - f_i}{(\delta_i^* - \mu_i a_i^*)}Jb_i^* \quad , \quad \hat{k}_i = I_r - \frac{1}{d_i\omega_i}\frac{Jb_i^* g_i}{(\delta_i^* - \mu_i a_i^*)} \quad ,$$

$$\hat{c}_i = \frac{1}{d_i^*\delta_i}\frac{\tau_i\delta_i - a_i}{(\omega_i^* - \tau_i f_i^*)}Jg_i^* \quad , \quad \hat{s}_i = I_r - \frac{1}{d_i^*\delta_i}\frac{Jg_i^* b_i}{(\omega_i^* - \tau_i f_i^*)}.$$

The claim is that other values of $\{h_i, k_i, c_i, s_i\}$ are related to $\{\hat{h}_i, \hat{k}_i, \hat{c}_i, \hat{s}_i\}$ by

$$h_i = \Theta_i^{-1}\hat{h}_i \ , \quad k_i = \Theta_i^{-1}\hat{k}_i \ , \quad c_i = \Gamma_i^{-1}\hat{c}_i \ , \quad s_i = \Gamma_i^{-1}\hat{s}_i \ ,$$

for arbitrary constant matrices Θ_i and Γ_i that satisfy $\Theta_i J\Gamma_i^* = J$. To check this, let $\Theta_i(z)$ and $\Gamma_i(z)$ be the transfer matrices of any other possible choice (c_i, s_i) and (h_i, k_i), respectively. Clearly, $\Gamma_i(\mu_i^{-*})J\Theta_i^*(\tau_i^{-*}) = J$, since $\tau_i\mu_i^* = 1$. If we define

$$\hat{\Theta}_i(z) = \Theta_i(z)\Theta_i^{-1}(\tau_i^{-*}) \quad \text{and} \quad \hat{\Gamma}_i(z) = \Gamma_i(z)\Gamma_i^{-1}(\mu_i^{-*}).$$

Then $\hat{\Theta}_i(\tau_i^{-*}) = I_r$ and $\hat{\Gamma}_i(\mu_i^{-*}) = I_r$. Using the fact that these conditions are satisfied by $(\hat{h}_i, \hat{k}_i)$ and $(\hat{c}_i, \hat{s}_i)$, we readily conclude that

$$h_i = \Theta_i^{-1}(\tau_i^{-*})\hat{h}_i \ , \quad k_i = \Theta_i^{-1}(\tau_i^{-*})\hat{k}_i \ , \quad c_i = \Gamma_i^{-1}(\mu_i^{-*})\hat{c}_i \ , \quad s_i = \Gamma_i^{-1}(\mu_i^{-*})\hat{s}_i.$$

■

Using the just derived expressions for (h_i, k_i) and (c_i, s_i) we can verify that $\Gamma_i(z)$ and $\Theta_i(z)$ in (25) can be rewritten in the following forms

$$\Gamma_i(z) = \left\{I_r + [B_{\Gamma,i}(z) - 1]\,\frac{Jg_i^* b_i}{b_i Jg_i^*}\right\}\Gamma_i \quad ,$$

$$\Theta_i(z) = \left\{I_r + [B_{\Theta,i}(z) - 1]\,\frac{Jb_i^* g_i}{g_i Jb_i^*}\right\}\Theta_i \quad ,$$

where

$$B_{\Gamma,i}(z) = \frac{f_i^* - \mu_i^* \omega_i^*}{\delta_i - \mu_i^* a_i} \; \frac{a_i - \delta_i z}{\omega_i^* - f_i^* z} \quad \text{and} \quad B_{\Theta,i}(z) = \frac{a_i^* - \tau_i^* \delta_i^*}{\omega_i - \tau_i^* f_i} \; \frac{f_i - \omega_i z}{\delta_i^* - a_i^* z}.$$

Consider for example, the special case of non-Hermitian Toeplitz-like structured matrices, viz.,

$$R - FRA^* = GJB^*.$$

Then the above expressions reduce to

$$B_{\Gamma,i}(z) = \frac{f_i^* - \mu_i^*}{1 - \mu_i^* a_i} \; \frac{a_i - z}{1 - f_i^* z} \quad \text{and} \quad B_{\Theta,i}(z) = \frac{a_i^* - \tau_i^*}{1 - \tau_i^* f_i} \; \frac{f_i - z}{1 - a_i^* z} \quad ,$$

and it is easy to see that the first-order sections $\Theta_i(z)$ and $\Gamma_i(z)$ have an interesting *interlaced* blocking property,

$$b_i \Gamma_i(a_i) = \mathbf{0} \quad \text{and} \quad g_i \Theta_i(f_i) = \mathbf{0} \quad ,$$

which can be used as a basis for an alternative approach to the solution of (constrained and unconstrained) interpolation problems (see, *e.g.,* [24, 26, 28]).

6 Generalized Schur Algorithm

We can also substitute the expressions for (h_i, k_i) and (c_i, s_i) into the generator recursions (21). This allows us to rewrite the recursions in the following alternative form (where $\{c_i, s_i, h_i, k_i\}$ are eliminated).

Theorem 2 (Generator Recursions) *The generator recursions (21) can also be written in the form*

$$\begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = \left[G_i + (\Phi_i - I_{n-i}) G_i \tfrac{J b_i^* g_i}{g_i J b_i^*} \right] \Theta_i \quad ,$$

$$\begin{bmatrix} \mathbf{0} \\ B_{i+1} \end{bmatrix} = \left[B_i + (\Psi_i - I_{n-i}) B_i \tfrac{J g_i^* b_i}{b_i J g_i^*} \right] \Gamma_i \quad ,$$

where

$$\Phi_i = \left(\tfrac{a_i^* - \tau_i^* \delta_i^*}{\omega_i - \tau_i^* f_i} \right) (f_i \Omega_i - \omega_i F_i)(\delta_i^* \Omega_i - a_i^* F_i)^{-1} \quad ,$$

$$\Psi_i = \left(\tfrac{f_i^* - \mu_i^* \omega_i^*}{\delta_i - \mu_i^* a_i} \right) (a_i \Delta_i - \delta_i A_i)(\omega_i^* \Delta_i - f_i^* A_i)^{-1} \; .$$

■

We now further simplify the recursions of Theorem 2 by properly choosing the free parameters $(\Theta_i, \Gamma_i, \tau_i, \mu_i)$. First recall that R is assumed to be strongly regular. This guarantees that $d_i \neq 0$ for every i and hence, the term $g_i J b_i^*$ can not vanish. Therefore, we can always choose constant matrices Θ_i and Γ_i such that $\Theta_i J \Gamma_i^* = J$ and the rows g_i and b_i are reduced to the forms

$$g_i \Theta_i = \begin{bmatrix} 0 & \ldots 0 & x_i^{(j)} & 0 & \ldots & 0 \end{bmatrix} \quad \text{and} \quad b_i \Gamma_i = \begin{bmatrix} 0 & \ldots 0 & y_i^{(j)} & 0 & \ldots & 0 \end{bmatrix} \quad , \qquad (27)$$

where the nonzero entries $x_i^{(j)}$ and $y_i^{(j)}$ are in the same column position, say the j^{th} position. Observe that $(G_i\Theta_i, B_i\Gamma_i)$ is also a generator pair of R_i since

$$G_i\Theta_i J\Gamma_i^* B_i^* = G_i J B_i^*.$$

Moreover, we shall say that a given generator pair is *proper* if the first row of each generator matrix has a single non-zero entry in the same column position. Therefore, the above choice (27) of (Θ_i, Γ_i) amounts to converting the original generator pair (G_i, B_i) to a proper generator pair $(G_i\Theta_i, B_i\Gamma_i)$:

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

$$B_i\Gamma_i = \begin{bmatrix} 0 & \ldots & 0 & y_i^{(j)} & 0 & \ldots & 0 \\ x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \end{bmatrix} .$$

It is clear that we must have

$$x_i^{(j)} y_i^{*(j)} = g_i J b_i^* = d_i(\omega_i \delta_i^* - f_i a_i^*).$$

We also remark that Θ_i and Γ_i can be implemented by using suitable variations of elementary transformations such as Householder, Givens, hyperbolic, etc.. Substituting (27) into the generator recursions of Theorem 2 we obtain the following generalized Schur algorithm in (proper) array form.

Algorithm 1 (Generalized Schur Algorithm: Array Form) *The generator recursions can be rewritten in the following simplified array form*

$$\begin{aligned} \begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} &= G_i\Theta_i \begin{bmatrix} I_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{r-j-1} \end{bmatrix} + \Phi_i G_i \Theta_i \begin{bmatrix} \mathbf{0}_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{r-j-1} \end{bmatrix} , \\ \begin{bmatrix} \mathbf{0} \\ B_{i+1} \end{bmatrix} &= B_i\Gamma_i \begin{bmatrix} I_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{r-j-1} \end{bmatrix} + \Psi_i B_i \Gamma_i \begin{bmatrix} \mathbf{0}_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{r-j-1} \end{bmatrix} , \end{aligned} \tag{28}$$

where Φ_i and Ψ_i are as before. The triangular factors are given by

$$\begin{aligned} l_i &= (\delta_i^* \Omega_i - a_i^* F_i)^{-1} G_i \Theta_i J \begin{bmatrix} 0 & \ldots & 0 & y_i^{(j)} & 0 & \ldots & 0 \end{bmatrix}^* , \\ u_i &= (\omega_i^* \Delta_i - f_i^* A_i)^{-1} B_i \Gamma_i J \begin{bmatrix} 0 & \ldots & 0 & x_i^{(j)} & 0 & \ldots & 0 \end{bmatrix}^* . \end{aligned} \tag{29}$$

■

The generator recursions (28) have a simple array interpretation:

(i) Convert (G_i, B_i) to a proper generator pair $(G_i\Theta_i, B_i\Gamma_i)$ with respect to a certain column, say the j^{th} column.

(ii) Multiply the j^{th} column of $(G_i\Theta_i)$ by Φ_i, and the j^{th} column of $(B_i\Gamma_i)$ by Ψ_i, while keeping all other columns of $(G_i\Theta_i, B_i\Gamma_i)$ unchanged.

(iii) These steps result in a pair of non-proper generators (G_{i+1}, B_{i+1}) and we return to step (i).

Notice that the $(0,0)$ entries of Φ_i and Ψ_i are zero. Hence, the effect of step (ii) is to annihilate the non-zero entries $x_i^{(j)}$ and $y_i^{(j)}$ of the j^{th} columns of $(G_i\Theta_i, B_i\Gamma_i)$:

$$G_i = \begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \\ \vdots & \vdots & \vdots \\ x & x & x \end{bmatrix} \overset{\Theta_i}{\rightarrow} \begin{bmatrix} \mathbf{0} & x_i^{(j)} & \mathbf{0} \\ x & x & x \\ x & x & x \\ \vdots & \vdots & \vdots \\ x & x & x \end{bmatrix} \xrightarrow{\Phi_i,\, j^{th}\text{col.}} \begin{bmatrix} \mathbf{0} & 0 & \mathbf{0} \\ x & x & x \\ x & x & x \\ \vdots & \vdots & \vdots \\ x & x & x \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} ,$$

$$B_i = \begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \\ \vdots & \vdots & \vdots \\ x & x & x \end{bmatrix} \overset{\Gamma_i}{\rightarrow} \begin{bmatrix} \mathbf{0} & y_i^{(j)} & \mathbf{0} \\ x & x & x \\ x & x & x \\ \vdots & \vdots & \vdots \\ x & x & x \end{bmatrix} \xrightarrow{\Psi_i,\, j^{th}\text{col.}} \begin{bmatrix} \mathbf{0} & 0 & \mathbf{0} \\ x & x & x \\ x & x & x \\ \vdots & \vdots & \vdots \\ x & x & x \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ B_{i+1} \end{bmatrix} .$$

The triangular factors l_i and u_i can be obtained via (29). Observe however, that we do not need to explicitly form the inverses $(\delta_i^*\Omega_i - a_i^*F_i)^{-1}$ and $(\omega_i^*\Delta_i - f_i^*A_i)^{-1}$. We may alternatively determine l_i and u_i by solving the following triangular linear systems of equations

$$(\delta_i^*\Omega_i - a_i^*F_i)l_i = G_i\Theta_i J \begin{bmatrix} \mathbf{0} & y_i^{(j)} & \mathbf{0} \end{bmatrix}^* ,$$

$$(\omega_i^*\Delta_i - f_i^*A_i)u_i = B_i\Gamma_i J \begin{bmatrix} \mathbf{0} & x_i^{(j)} & \mathbf{0} \end{bmatrix}^* .$$

6.1 A Remark

Finally, the previous derivation was based on the assumption that Ω, Δ, F, and A in (8) are lower triangular matrices. If this is not so, then we can apply the generalized Schur decomposition theorem [30], which guarantees the existence of unitary matrices Q_1, Q_2, P_1, and P_2 such that

$$\Omega = Q_1^*\hat{\Omega}Q_2 \ , \quad F = Q_1^*\hat{F}Q_2 \ , \quad \Delta = P_1^*\hat{\Delta}P_2, \text{ and } A = P_1^*\hat{A}P_2 \ ,$$

where $\hat{\Omega}$, $\hat{\Delta}$, $\hat{F}$, and $\hat{A}$ are lower triangular. If we define $\hat{R} = Q_2RP_2^*$, $\hat{G} = Q_1G$ and $\hat{B} = P_1B$, then equation (8) reduces to

$$\hat{\Omega}\hat{R}\hat{\Delta}^* - \hat{F}\hat{R}\hat{A}^* = \hat{G}J\hat{B}^*.$$

We can now compute the triangular factorization $\hat{R} = \hat{L}\hat{D}\hat{U}$ using the derived recursions, and this leads to the useful factorization $R = Q_2^*\hat{L}\hat{D}\hat{U}P_2$.

6.2 Remarks on Nonuniqueness and Computational Complexity

We now discuss some points relevant to the invertibility, uniqueness conditions, and computational complexity of the generalized Schur Algorithm. We first remark that at each step we need to (at least implicitly) compute the inverses

$$(\omega_i^*\Delta_i - f_i^*A_i)^{-1} \quad \text{and} \quad (\delta_i^*\Omega_i - a_i^*F_i)^{-1}.$$

These are $(n-i)\times(n-i)$ matrices whose inversion, in general, requires $O((n-i)^3)$ operations. If we are interested in a fast $(O(n^2))$ algorithm then $\Omega, \Delta, F,$ and A have to satisfy the additional constraint that solving the linear systems for l_i and u_i requires $O(n-i)$ operations. For example, in the special case

$$\Omega = \Delta = I, \quad F \text{ and } A \text{ are strictly lower triangular},$$

the matrices $(\omega_i^*\Delta_i - f_i^*A_i)$ and $(\delta_i^*\Omega_i - a_i^*F_i)$ are both equal to the identity matrix. Moreover, in interpolation problems (see, *e.g.,* [24, 28]) one is often faced with

$$\Omega = \Delta = I, \quad F = A = \text{ a diagonal matrix },$$

in which case $(\omega_i^*\Omega_i - f_i^*F_i)$ is diagonal and hence easily invertible.

We also assumed throughout our discussion that R is unique and hence, $(\omega_j\delta_i^* - f_ja_i^*) \neq 0$ for all i, j. This condition guarantees the invertibility of the matrices $(\omega_i^*\Delta_i - f_i^*A_i)$ and $(\delta_i^*\Omega_i - a_i^*F_i)$. Suppose for instance that this is not the case, then we are led to the following expressions for l_i and u_i:

$$(\delta_i^*\Omega_i - a_i^*F_i)\, l_i = G_iJb_i^* \quad \text{and} \quad (\omega_i^*\Delta_i - f_i^*A_i)\, u_i = B_iJg_i^*,$$

which show that if either u_i or l_i is orthogonal to the nullspaces of $(\omega_i^*\Delta_i - f_i^*A_i)$ or $(\delta_i^*\Omega_i - a_i^*F_i)$, respectively, then we can determine either u_i or l_i unambiguously by using appropriate pseudo-inverse: $(\omega_i^*\Delta_i - f_i^*A_i)^\dagger$ or $(\delta_i^*\Omega_i - a_i^*F_i)^\dagger$. This approach was considered in [8] for the Hankel-like case. To illustrate this point, we include a simple example extracted from [8, page 92].

6.3 A Simple Example

Consider an $n \times n$ real Hankel matrix H, which has displacement rank 2 with respect to the displacement operation $ZH - HZ^*$ (recall (5)). We remarked earlier that H cannot be recovered from its displacement ∇H, because the entries $\{h_{n-1}, \ldots, h_{2n-2}\}$ do not appear in ∇H. One solution to this difficulty [8, 12] is to embed H into an $(n+1)\times(n+1)$ extended matrix M defined as follows

$$M = \begin{bmatrix} H & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix},$$

where H is a leading submatrix of M. The extended matrix M also has structure with respect to the displacement operation $Z_{n+1}M - MZ_{n+1}^*$, where Z_{n+1} is the $(n+1)\times(n+1)$ lower shift,

$$\nabla M = \begin{bmatrix} 0 & -h_0 & \ldots & -h_{n-2} & -h_{n-1} \\ h_0 & & & & -h_n \\ \vdots & & \text{O} & & \vdots \\ h_{n-2} & & & & -h_{2n-2} \\ h_{n-1} & h_n & \ldots & h_{2n-2} & 0 \end{bmatrix} \quad \text{has rank 4.}$$

The structure of M clearly corresponds to $\Omega = Z_{n+1},\ \Delta = I,\ F = I,\ A = Z_{n+1}$, and the nullspace of $(\delta_0^* \Omega - a_0^* F)$ $(\equiv Z_{n+1})$ is the set of all $(n+1) \times 1$ column vectors of the form

$$\left[\begin{array}{cccc} 0 & \dots & 0 & \alpha \end{array} \right]^T, \quad \text{for arbitrary scalar } \alpha.$$

The triangular factors of M (denoted by $l_i^{(M)}$) are orthogonal to the above nullspace, since the last entries of $l_i^{(M)}$ are zero for all i. We can thus determine $l_i^{(M)}$ uniquely, and by deleting the last entry we obtain the factors l_i of H.

An alternative procedure (see [7, 19]) is to extend H to a $2n \times 2n$ Hankel matrix $\bar{H}$,

$$\bar{H} = \left[\begin{array}{ccccccccc} h_0 & h_1 & \dots & h_{n-1} & h_n & h_{n+1} & \dots & h_{2n-2} & 0 \\ h_1 & h_2 & . & h_n & h_{n+1} & h_{n+1} & . & 0 & \\ h_2 & h_3 & \dots & h_{n+1} & h_{n+2} & . & & & \\ & & & & . & & & & \\ \vdots & & & . & & & & & \\ & & . & & & & & & \\ & & & & & \text{O} & & & \\ & . & & & & & & & \\ h_{2n-2} & 0 & & & & & & & \\ 0 & & & & & & & & \end{array} \right],$$

and to use the displacement representation of $\bar{H}$ with respect to $Z_{2n}\bar{H} - \bar{H}Z_{2n}^*$. Notice that all the entries $\{h_0, \dots, h_{2n-1}\}$ that define the original matrix H appear in the displacement $\nabla \bar{H}$.

7 Generalized State-Space Realizations

We showed in Section 5 that each generator recursion gives rise to two first-order discrete-time systems $\Theta_i(z)$ and $\Gamma_i(z)$ as in (25) (or alternatively (23) and (24)). Therefore, after n steps we obtain two cascades that we denote by $\mathbf{\Theta}(z)$ and $\mathbf{\Gamma}(z)$, viz.,

$$\mathbf{\Theta}(z) = \Theta_0(z)\Theta_1(z) \dots \Theta_{n-1}(z) \ ,$$

$$\mathbf{\Gamma}(z) = \Gamma_0(z)\Gamma_1(z) \dots \Gamma_{n-1}(z).$$

It is clear that $\mathbf{\Theta}(z)$ and $\mathbf{\Gamma}(z)$ also satisfy the generalized J-losslessness relation

$$\mathbf{\Gamma}(z) J \mathbf{\Theta}^*(w) = J \ \text{ on } \ zw^* = 1.$$

We shall use the following notation to represent the transfer matrices $\Theta_i(z)$ and $\Gamma_i(z)$ (recall (23) and (24))

$$\Gamma_i(z) \sim \left[\begin{array}{cc} f_i^* \omega_i^{-*} & h_i^* J \\ J g_i^* \omega_i^{-*} & J k_i^* J \end{array} \right] \quad \text{and} \quad \Theta_i(z) \sim \left[\begin{array}{cc} a_i^* \delta_i^{-*} & c_i^* J \\ J b_i^* \delta_i^{-*} & J s_i^* J \end{array} \right].$$

We first use the last two sections of each of the cascades and compute their state-space realizations defined by

$$W_{n-2}(z) \sim \Theta_{n-2}(z)\Theta_{n-1}(z) \quad \text{and} \quad T_{n-2}(z) \sim \Gamma_{n-2}(z)\Gamma_{n-1}(z) \ ,$$

We then use $\Theta_{n-3}(z)$ and $\Gamma_{n-3}(z)$ to compute

$$W_{n-3}(z) \sim \Theta_{n-3}(z)W_{n-2}(z) \quad \text{and} \quad T_{n-3}(z) \sim \Gamma_{n-3}(z)T_{n-3}(z) \ ,$$

and so on. Clearly, $T_0(z)$ and $W_0(z)$ represent the entire cascades $\mathbf{\Gamma}(z)$ and $\mathbf{\Theta}(z)$, respectively. In matrix notation, the cascades $\Gamma_{n-2}(z)\Gamma_{n-1}(z)$ and $\Theta_{n-2}(z)\Theta_{n-1}(z)$ are given by

$$T_{n-2}(z) \sim \begin{bmatrix} f_{n-2}^* \omega_{n-2}^{-*} & \mathbf{0} & h_{n-2}^* J \\ 0 & 1 & \mathbf{0} \\ J g_{n-2}^* \omega_{n-2}^{-*} & \mathbf{0} & J k_{n-2}^* J \end{bmatrix} \begin{bmatrix} 1 & 0 & \mathbf{0} \\ 0 & f_{n-1}^* \omega_{n-1}^{-*} & h_{n-1}^* J \\ \mathbf{0} & J g_{n-1}^* \omega_{n-1}^{-*} & J k_{n-1}^* J \end{bmatrix} ,$$

$$W_{n-2}(z) \sim \begin{bmatrix} a_{n-2}^* \delta_{n-2}^{-*} & \mathbf{0} & c_{n-2}^* J \\ 0 & 1 & \mathbf{0} \\ J b_{n-2}^* \delta_{n-2}^{-*} & \mathbf{0} & J s_{n-2}^* J \end{bmatrix} \begin{bmatrix} 1 & 0 & \mathbf{0} \\ 0 & a_{n-1}^* \delta_{n-1}^{-*} & c_{n-1}^* J \\ \mathbf{0} & J b_{n-1}^* \delta_{n-1}^{-*} & J s_{n-1}^* J \end{bmatrix} .$$

We are interested in evaluating the right-hand side of the last two expressions. We first remark that (22) leads to

$$\begin{bmatrix} f_i & g_i \\ h_i & k_i \end{bmatrix}^{-1} = \begin{bmatrix} a_i^* (\omega_i \delta_i^*)^{-1} & d_i c_i^* J \\ (d_i \omega_i \delta_i^*)^{-1} J b_i^* & J s_i^* J \end{bmatrix} ,$$

$$\begin{bmatrix} a_i & b_i \\ c_i & s_i \end{bmatrix}^{-1} = \begin{bmatrix} f_i^* (\omega_i^* \delta_i)^{-1} & d_i^* h_i^* J \\ (d_i^* \omega_i^* \delta_i)^{-1} J g_i^* & J k_i^* J \end{bmatrix} ,$$

and, for notational simplicity, we introduce the matrices L_i and U_i defined by

$$L_i = \begin{bmatrix} \frac{l_i}{d_i} & 0 & \mathbf{0} \\ & I_{n-i-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_r \end{bmatrix} \quad \text{and} \quad U_i = \begin{bmatrix} \frac{u_i}{d_i^*} & 0 & \mathbf{0} \\ & I_{n-i-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_r \end{bmatrix} .$$

Using the expressions for G_{n-1}, B_{n-1}, and the triangular factors l_{n-2} and u_{n-2}, it can be seen that we can write

$$\begin{bmatrix} \frac{l_{n-2}}{d_{n-2}} & \begin{matrix} 0 \\ 1 \end{matrix} \end{bmatrix} \begin{bmatrix} \omega_{n-2}^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \omega_{n-1}^{-1} f_{n-1} & \omega_{n-1}^{-1} g_{n-1} \end{bmatrix} \begin{bmatrix} f_{n-2} & 0 & g_{n-2} \\ 0 & 1 & 0 \\ h_{n-2} & 0 & k_{n-2} \end{bmatrix} L_{n-2}^{-1} = \begin{bmatrix} \Omega_{n-2}^{-1} F_{n-2} & \Omega_{n-2}^{-1} G_{n-2} \end{bmatrix} \tag{30}$$

$$\begin{bmatrix} \frac{u_{n-2}}{d_{n-2}^*} & \begin{matrix} 0 \\ 1 \end{matrix} \end{bmatrix} \begin{bmatrix} \delta_{n-2}^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \delta_{n-1}^{-1} a_{n-1} & \delta_{n-1}^{-1} b_{n-1} \end{bmatrix} \begin{bmatrix} a_{n-2} & 0 & b_{n-2} \\ 0 & 1 & 0 \\ c_{n-2} & 0 & s_{n-2} \end{bmatrix} U_{n-2}^{-1} = \begin{bmatrix} \Delta_{n-2}^{-1} A_{n-2} & \Delta_{n-2}^{-1} B_{n-2} \end{bmatrix} . \tag{31}$$

If we now introduce the quantities $H_{n-2}, K_{n-2}, C_{n-2}$, and S_{n-2}, defined by

$$\begin{bmatrix} H_{n-2} & K_{n-2} \end{bmatrix} = \begin{bmatrix} 0 & h_{n-1} & k_{n-1} \end{bmatrix} \begin{bmatrix} \omega_{n-2}^{-1} f_{n-2} & 0 & \omega_{n-2}^{-1} g_{n-2} \\ 0 & 1 & 0 \\ h_{n-2} & \mathbf{0} & k_{n-2} \end{bmatrix} L_{n-2}^{-1} ,$$

$$\begin{bmatrix} C_{n-2} & S_{n-2} \end{bmatrix} = \begin{bmatrix} 0 & c_{n-1} & s_{n-1} \end{bmatrix} \begin{bmatrix} \delta_{n-2}^{-1} a_{n-2} & 0 & \delta_{n-2}^{-1} b_{n-2} \\ 0 & 1 & 0 \\ c_{n-2} & \mathbf{0} & s_{n-2} \end{bmatrix} U_{n-2}^{-1} ,$$

then (30) and (31) imply that

$$T_{n-2} \sim \begin{bmatrix} F_{n-2}^{*}\Omega_{n-2}^{-*} & H_{n-2}^{*}J \\ & \\ JG_{n-2}^{*}\Omega_{n-2}^{-*} & JK_{n-2}^{*}J \end{bmatrix} \quad \text{and} \quad W_{n-2} \sim \begin{bmatrix} A_{n-2}^{*}\Delta_{n-2}^{-*} & C_{n-2}^{*}J \\ & \\ JB_{n-2}^{*}\Delta_{n-2}^{-*} & JS_{n-2}^{*}J \end{bmatrix}.$$

Moreover, it follows from (13) and (22) that

$$\begin{bmatrix} F_{n-2} & G_{n-2} \\ H_{n-2} & K_{n-2} \end{bmatrix} \begin{bmatrix} R_{n-2} & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} \begin{bmatrix} A_{n-2} & B_{n-2} \\ C_{n-2} & S_{n-2} \end{bmatrix}^{*} = \begin{bmatrix} \Omega_{n-2}R_{n-2}\Delta_{n-2}^{*} & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix}.$$

The same argument can now be used to compute $T_{n-3}(z)$ and $W_{n-3}(z)$ and so on. Each further step would correspond to a relation of the form

$$\begin{bmatrix} F_i & G_i \\ H_i & K_i \end{bmatrix} \begin{bmatrix} R_i & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} \begin{bmatrix} A_i & B_i \\ C_i & S_i \end{bmatrix}^{*} = \begin{bmatrix} \Omega_i R_i \Delta_i^{*} & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} , \tag{32}$$

and

$$\begin{bmatrix} H_i & K_i \end{bmatrix} = \begin{bmatrix} \mathbf{0} & H_{i+1} & K_{i+1} \end{bmatrix} \begin{bmatrix} \omega_i^{-1} f_i & \mathbf{0} & \omega_i^{-1} g_i \\ \mathbf{0} & I_{n-i-1} & \mathbf{0} \\ h_i & \mathbf{0} & k_i \end{bmatrix} L_i^{-1} ,$$

$$\begin{bmatrix} C_i & S_i \end{bmatrix} = \begin{bmatrix} \mathbf{0} & C_{i+1} & S_{i+1} \end{bmatrix} \begin{bmatrix} \delta_i^{-1} a_i & \mathbf{0} & \delta_i^{-1} b_i \\ \mathbf{0} & I_{n-i-1} & \mathbf{0} \\ c_i & \mathbf{0} & s_i \end{bmatrix} U_i^{-1}.$$

Theorem 3 (State-Space Realizations) *The cascades $\mathbf{\Theta}(z)$ and $\mathbf{\Gamma}(z)$ have the following $n-$dimensional state-space realizations*

$$\begin{aligned} \Delta^{*}\mathbf{x}_{j+1}^{(1)} &= \mathbf{x}_j^{(1)} A^{*} + \mathbf{w}_k^{(1)} JB^{*} \\ \mathbf{y}_j^{(1)} &= \mathbf{x}_j^{(1)} C^{*}J + \mathbf{w}_k^{(1)} JS^{*}J \ , \end{aligned}$$

and

$$\begin{aligned} \Omega^{*}\mathbf{x}_{j+1}^{(2)} &= \mathbf{x}_j^{(2)} F^{*} + \mathbf{w}_k^{(2)} JG^{*} \\ \mathbf{y}_j^{(2)} &= \mathbf{x}_j^{(2)} H^{*}J + \mathbf{w}_k^{(2)} JK^{*}J \ , \end{aligned}$$

respectively, and satisfy the generalized embedding relation

$$\begin{bmatrix} F & G \\ H & K \end{bmatrix} \begin{bmatrix} R & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} \begin{bmatrix} A & B \\ C & S \end{bmatrix}^{*} = \begin{bmatrix} \Omega R \Delta^{*} & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix}. \tag{33}$$

Moreover, it also follows that $\mathbf{\Gamma}(z)$ and $\mathbf{\Theta}(z)$ admit the representations:

$$\mathbf{\Gamma}(z) = \{I - (z^{-1} - \mu^{*})JG^{*}(z^{-1}\Omega^{*} - F^{*})^{-1}R^{-*}(\Delta - \mu^{*}A)^{-1}B\}\,\mathbf{\Gamma} \ ,$$

$$,$$

$$\mathbf{\Theta}(z) = \{I - (z^{-1} - \tau^{*})JB^{*}(z^{-1}\Delta^{*} - A^{*})^{-1}R^{-1}(\Omega - \tau^{*}F)^{-1}G\}\,\mathbf{\Theta}$$

where τ and μ satisfy $\tau\mu^{*} = 1$, and the constant matrices $\mathbf{\Theta}$ and $\mathbf{\Gamma}$ satisfy $\mathbf{\Theta}J\mathbf{\Gamma}^{*} = J$ (the values of $\{\Theta, \Gamma, \tau, \mu\}$ depend on the choices $\{\Theta_i, \Gamma_i, \tau_i, \mu_i\}$).

■

8 Concluding Remarks

We introduced a generalized definition of displacement structure and showed how to exploit it to derive fast triangular factorization algorithms for such matrices. We combined a simple Gaussian elimination procedure with displacement structure and derived the corresponding generator recursions in a convenient array form. It was also verified that each step of the algorithm provides two first-order sections that satisfy a general embedding relation and a generalized notion of J-losslessness. We also derived a state-space realization for the cascade in terms of the matrices that describe the matrix (displacement)structure. An application of the generalized recursions derived in this paper to unconstrained interpolation problems is discussed in [28].

References

[1] T. Kailath. Signal processing applications of some moment problems. In H. Landau, editor, *Moments in Mathematics*, vol. 37, pages 71–109. AMS, Providence, 1987.

[2] T. Kailath, S. Y. Kung, and M. Morf. Displacement ranks of a matrix. *Bulletin of the American Mathematical Society*, 1(5):769–773, September 1979.

[3] T. Kailath, S. Y. Kung, and M. Morf. Displacement ranks of matrices and linear equations. *Journal of Mathematical Analysis and Applications*, 68:395–407, 1979.

[4] T. Kailath. Remarks on the origin of the displacement-rank concept. *Applied Mathematics and Computation*, 45:193–206, 1991.

[5] H. Lev-Ari. *Nonstationary Lattice-Filter Modeling.* PhD thesis, Stanford University, Stanford, CA, December 1983.

[6] H. Lev-Ari and T. Kailath. Lattice filter parametrization and modeling of nonstationary processes. *IEEE Transactions on Information Theory*, 30(1):2–16, January 1984.

[7] H. Lev-Ari and T. Kailath. Triangular factorization of structured Hermitian matrices. *Operator Theory: Advances and Applications*, edited by I. Gohberg, Birkhäuser, Boston, vol. 18:301–324, 1986.

[8] J. Chun. *Fast Array Algorithms for Structured Matrices.* PhD thesis, Stanford University, Stanford, CA, 1989.

[9] T. Kailath and J. Chun. Generalized displacement structure for block-Toeplitz, Toeplitz-block, and Toeplitz-derived matrices. To appear in *SIAM J. Matrix Anal. Appl.*.

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

[11] G. Heinig and K. Rost. *Algebraic Methods for Toeplitz-like Matrices and Operators.* Akademie-Verlag, Berlin and Birkhäuser, 1984.

[12] J. Chun and T. Kailath. Displacement structure for Hankel- and Vandermonde- like matrices. In L. Auslander, T. Kailath, and S. Mitter, editors, *Signal Processing, Part I: Signal Processing Theory*, pages 37–58. Springer–Verlag, NY, 1990.

[13] N. Levinson. The Wiener r.m.s. (root-mean-square) error criterion in filter design and prediction. *J. Math. Phys.*, 25:261–278, January 1947.

[14] E. H. Bareiss. Numerical solution of linear equations with Toeplitz and vector Toeplitz matrices. *Numer. Math.*, 13:404–424, 1969.

[15] J. LeRoux and C. Gueguen. A fixed-point computation of parcor coefficients. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 25(3):257–259, June 1977.

[16] M. Morf. *Fast Algorithms for Multivariable Systems.* PhD thesis, Stanford University, Stanford, CA, 1974.

[17] J. Rissanen. Algorithms for triangular decomposition of block Hankel and Toeplitz matrices with application to factoring positive matrix polynomials. *Mathematics of Computation*, 27(121):147–154, January 1973.

[18] P. Dewilde, A. Vieira, and T. Kailath. On a generalized Szegö-Levinson realization algorithm for optimal linear predictions on a network synthesis approach. *IEEE Transactions on Circuits and Systems*, 25:663–675, 1978.

[19] A. H. Sayed, H. Lev-Ari, and T. Kailath. Fast triangular factorization of the sum of quasi-Toeplitz and quasi-Hankel matrices. To appear in *Linear Algebra and Its Applications.*

[20] J. Chun, T. Kailath, and H. Lev-Ari. Fast parallel algorithms for QR and triangular factorization. *SIAM J. Sci. Stat. Comput.*, 8(6):899–913, November 1987.

[21] A. H. Sayed, H. Lev-Ari, and T. Kailath. Time-variant displacement structure and triangular arrays. To appear in *IEEE Transactions on Signal Processing.*

[22] C. G. J. Jacobi. Über einen algebraischen Fundamentalsatz und seine Anwendungen. *Journal für die reine und angewandte Mathematik*, vol. 53, pp. 275–280, 1857, posthumous.

[23] A. H. Sayed. Unpublished notes. 1991.

[24] A. H. Sayed and T. Kailath. Recursive solutions to rational interpolation problems. In *Proc. IEEE International Symposium on Circuits and Systems*, vol. 5, pp. 2376–2379, San Diego, CA, May 1992.

[25] J. A. Ball, I. Gohberg, and L. Rodman. *Interpolation of Rational Matrix Functions*, volume 45 of *Operator Theory: Advances and Applications*, Birkhäuser, Basel, 1990.

[26] A. H. Sayed. *Displacement Structure in Signal Processing and Mathematics.* PhD thesis, Stanford University, Stanford, CA, August 1992.

[27] H. Lev-Ari and T. Kailath. State-space approach to factorization of lossless transfer functions and structured matrices. *Linear Algebra and Its Applications*, 162–164:273–295, February 1992.

[28] T. Boros, A. H. Sayed, and T. Kailath. Structured matrices and unconstrained rational interpolation. To appear in *Linear Algebra and Its Applications.*

[29] F. R. Gantmacher. *The Theory of Matrices.* Chelsea Publishing Company, NY, 1959.

[30] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, second edition, 1989.

[31] Y. Genin, P. V. Dooren, T. Kailath, J. Delosme, and M. Morf. On $\Sigma-$lossless transfer functions and related questions. *Linear Algebra and Its Applications*, 50:251–275, 1983.

[32] T. Kailath. *Linear Systems.* Prentice Hall, Englewood Cliffs, NJ, 1980.