SIAM REVIEW, VOL. 37, NO.3, PP. 297–386, SEP. 95

DISPLACEMENT STRUCTURE: THEORY AND APPLICATIONS *

THOMAS KAILATH† AND ALI H. SAYED‡

Abstract. In this survey paper, we describe how strands of work important in two different fields, matrix theory and complex function theory, have come together in some work on fast computational algorithms for matrices with what we have called displacement structure. In particular, a fast triangularization procedure can be developed for such matrices, generalizing in a striking way an algorithm presented by Schur (1917) in a paper on checking when a power series is bounded in the unit disc. This factorization algorithm has a surprisingly wide range of significant applications going far beyond numerical linear algebra. We mention, among others, inverse scattering, analytic and unconstrained rational interpolation theory, digital filter design, adaptive filtering, and state-space least-squares estimation.

Key words. Displacement structure, structured matrices, generalized Schur algorithm, triangular matrix factorization, interpolation theory, time-variant structures, state-space models, Kalman filtering, adaptive filtering.

AMS subject classifications. 15A23, 15A06, 30C15, 30D50, 30E05, 47A57, 60G35, 93A20, 93C50, 93C55, 93E11, 93E24.

Abbreviated title. Displacement Structure.

1. Introduction.

We describe how two strands of work from matrix theory and function theory have come together in interesting ways in some work on fast computational algorithms for matrices with what we have called displacement structure. This structure can be identified in a surprising variety of applications in engineering, mathematics, and physics. However, we may mention that, as so often happens, we came to the matrix problems quite indirectly, starting with work in 1972-1973 [106, 107] on certain nonlinear matrix Riccati and Chandrasekhar differential equations [45], going on to scattering theory in radiative transfer and transmission lines [116, 140, 164, 195], and then to Fredholm- and Wiener-Hopf-types of linear integral equations [117]. Two recent surveys [110, 111] elaborate on these early developments.

The discrete-time analogues of these results were developed somewhat in parallel (see [118, 122, 144, 146, 147, 186]), but with more initial effort, and it took some time to focus on the purely algebraic problems of structured matrices. Here we had available various famous results on Toeplitz matrices, especially the Levinson algorithm [137, 196], the Szegő polynomials orthogonal on the unit circle [85, 192], the Gohberg-Semencul formulas for inverses of Toeplitz matrices [86, 92], and a difference form of these formulas, discovered earlier by Trench [193]; these results were discussed, along with their somewhat lesser-known continuous-time analogues, in the survey paper [121]. It became clear through the above work that the Toeplitz structure was encompassed by a more general concept of displacement structure, though going from continuous-time to the right definitions in discrete-time took a while [114, 115].

* This work was supported in part by the Army Research Office under contract DAAL03-89-K-0109, and also by a grant from the National Science Foundation under award no. MIP-9409319.

† Information Systems Laboratory, Dept. of Elec. Eng., Stanford University, CA 94305.

‡ The work of Dr. Sayed was also supported by a fellowship from Fundação de Amparo à Pesquisa do Estado de São Paulo, Brazil, while on leave from the Department of Electrical Engineering, Escola Politécnica da Universidade de São Paulo, Brazil. He is currently an Assistant Professor in the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106.

Moreover on the algorithmic side, the initial efforts focused on generalizations of the Levinson algorithm [66, 79, 122] and the equivalent problem of fast triangular factorization of the *inverses* of matrices with displacement structure (see also [63, 78, 79, 144]). The direct factorization problem, even for Toeplitz matrices, was less studied, though algorithms for the Toeplitz case existed, *e.g.,* [22, 144]; the references [129, 166] gave derivations for the Toeplitz case that started with the Levinson algorithm and [144] did the same for more general matrices. In 1976-1977, our work on scattering theory [110] led to a collaboration with P. Dewilde, whose background in circuit theory led him to recognize that an algorithm of I. Schur (1917) [185] on a problem in complex function theory was relevant to these problems - see the paper [70]. Dewilde, Fokkema, and Widya [69] and Bruckstein and Kailath [34, 35] noted that the Schur algorithm arose naturally in studying inverse scattering problems for discrete transmission lines (and by extension for Schrodinger equations [36]).

Schur's work was closely studied and extended by Lev-Ari and Kailath [130, 133, 134] via the use of generating functions and Schur complements. Chun, Kailath, and Lev-Ari [48, 52, 113] showed the further power of the Schur complement concept by noting that problems such as triangular and orthogonal (QR) factorization for composite (Toeplitz- and Hankel-derived) matrices such as $T^{-1}, T_1T_2, T_1 - T_2T_3^{-1}T_4$, H_1H_2, etc, where $\{T_i, H_i\}$ are Toeplitz and Hankel matrices themselves, could be efficiently solved by first considering these composite matrices as Schur complements in suitable Toeplitz or Hankel block matrices, then introducing a further generalization of displacement structure and developing a generalized Schur algorithm in a so-called array or square-root form. This refers to the fact that the generic form of the algorithm is the triangularization by a $J-$unitary matrix (and for a recursive solution, a sequence of elementary circular and hyperbolic rotations/reflections) of a suitably defined prearray. Such algorithms were first proposed, for numerical reasons, in least-squares theory [93, 146] and were introduced for Levinson-type algorithms in [122]. They now are the main focus of many computationally effective and recursive filtering algorithms (see, e.g., [179]). This focus was given further impetus by the rederivation [135], using state-space concepts and some embedding results [83], of an even more general Schur algorithm originally presented in [130, Chapter 8]. These results were further simplified and generalized by Sayed and Kailath in [120, 173, 182], [171, Ch. 2] using purely matrix-based arguments. This work made clear that the various generalized Schur algorithms arose by using various definitions of displacement structure to speed up the classical (Gauss/Jacobi/Cholesky) procedures for triangular factorization of a matrix. It also made clear that a lossless cascade network/transmission line was naturally associated with every generalized Schur algorithm. Such lossless systems have certain "blocking properties" (see Sec. 6.1.1) which immediately lead to new recursive solutions of many (constrained and unconstrained) rational interpolation problems [29, 171, 173, 182]. Rational interpolation is a subject with a very rich history of its own; here we refer only to the recent books [20, 71, 76, 100, 169] and their many references.

Recently, we have returned to the original problems that led to the concept of displacement structure. The Chandrasekhar recursions derived in 1972–1973 [106, 107, 118, 146, 147] are now seen to be a consequence of combining state-space structure and displacement structure. This connection, known to us for many years, recently led us to generalize the original Chandrasekhar recursions to certain time-variant state-space systems [178], which then led us to a definition of time-variant displacement structure [171, 172, 173, 174, 184]. This not only allowed us to extend our earlier approach to now obtain recursive solutions to certain time-variant interpolation problems [174], studied in [19, 68], but also led to new algorithms for several types of matrix completion problems [56, 173], and for adaptive filtering and instrumental-variable problems [170, 172, 184].

This brief historical review has noted some of the highlights of our now two-decade long study of the role of displacement structure in deriving efficient computational algorithms for a variety of problems in engineering and mathematics. Many other authors have worked on closely related topics and we have profited from interactions of various kinds with them. It would be impossible to review all these contributions here or to make a list of acknowledgment without risk of omitting some. The richness of this field makes many different perspectives possible and valuable, and we hope that our survey will encourage other authors to offer similar unique perspectives on the variety of problems described below.

The following is a brief outline of the paper. We first review some early results on matrices with displacement structure and highlight connections with a classical algorithm of Schur and with inverse scattering problems. We then introduce several generalizations of the notion of displacement structure, along with examples that motivate the need for such extensions and justify the importance of direct factorization algorithms. A hierarchy of generalized Schur algorithms is derived and exhibited in several different forms, including the so-called proper and generating-function forms. As mentioned before, the term generalized Schur algorithm will be used in a generic sense for fast algorithms for computing Schur complements (and thereby obtaining triangular factors) of matrices with displacement structure. Connections with lossless systems, embedding relations, and transmission zeros are highlighted and shown to be relevant to the solution of interpolation problems. Generalized Schur algorithms are then studied in the presence of state-space structure, with immediate applications to problems in state-space estimation and adaptive filtering. We conclude with a brief account of extensions of the notion of displacement structure to time-variant matrices and with even briefer remarks on other results, applications and some open problems. A table of contents may be helpful in giving a flavor of the variety of topics encountered here, and as a guide to judicious browsing in this long survey paper.

2. Some Early Results. The concept of displacement structure is perhaps best introduced by considering the much-studied special case of a symmetric (or even Hermitian) Toeplitz matrix, $T = [c_{i-j}]_{i,j=0}^{n-1}$. 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 $O(n^3)$ operations. 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* in a sense first defined in [114, 115] and later much studied and generalized. Our first aim is to give a quick illustration of this claim.

The displacement of an $n \times n$ Hermitian matrix R was originally defined by Kailath et al. (1979) [114, 115] as

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

where the $*$ stands for Hermitian conjugation (complex conjugation for scalars), and Z is the $n \times n$ lower shift matrix with ones on the first subdiagonal and zeros elsewhere; ZRZ^* then 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 (2.1a), and r is referred to as the *displacement rank* of R. The definition can be extended to non-Hermitian, and in fact non-square matrices, and this will be briefly described later. Here we may note that in the Hermitian case, ∇R is Hermitian and therefore has further structure: its eigenvalues are real and so we can define the *displacement inertia* of R as the pair $\{p, q\}$, where p (resp. q) is the number of strictly positive (resp. negative) eigenvalues of ∇R. Of course, the displacement rank is $r = p + q$. Therefore, we can write

$$\nabla R = R - ZRZ^* = GJG^* \quad , \tag{2.1b}$$

where $J = J^* = (I_p \oplus -I_q)$, $p+q = r$, and G is an $n \times r$ matrix. [This representation is clearly not unique; the nonuniqueness will be completely characterized below.].

The pair $\{G, J\}$ will be 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 G. Using the fact that Z is nilpotent, viz., $Z^n = \mathbf{0}$, we can check that the unique solution of (2.1b) for a given $\{G, J\}$ is

$$R = \sum_{i=0}^{n-1} Z^i GJG^* Z^{*i} \, . \tag{2.2a}$$

Let us partition the columns of G into two sets $\{\mathbf{x}_i\}_{i=0}^{p-1}$ and $\{\mathbf{y}_i\}_{i=0}^{q-1}$,

$$G = \left[\begin{array}{cccccc} \mathbf{x}_0 & \mathbf{x}_1 & \ldots \mathbf{x}_{p-1} & \mathbf{y}_0 & \mathbf{y}_1 & \ldots & \mathbf{y}_{q-1} \end{array}\right] \ , \ p + q = r.$$

It is then easy to see that (2.2a) is equivalent to the representation

$$R = \sum_{i=0}^{p-1} \mathbf{L}(\mathbf{x}_i)\mathbf{L}^*(\mathbf{x}_i) - \sum_{i=0}^{q-1} \mathbf{L}(\mathbf{y}_i)\mathbf{L}^*(\mathbf{y}_i) \quad , \tag{2.2b}$$

where the notation $\mathbf{L}(\mathbf{x})$ denotes a lower triangular Toeplitz matrix whose first column is $\mathbf{x}$. [We may remark that the representation (2.2b) allows us to replace the $O(n^2)$ operations usually required to form a matrix-vector product, say $R\mathbf{a}$, by $2(p + q)$ convolutions, each requiring only $O(n \log n)$ operations; for more on such applications see, e.g., [12, 81, 88, 89, 121].].

The choice of a generator matrix in (2.1b) is not unique, since, for example, $G\Theta$ is also a generator for any $J-$unitary matrix Θ ($\Theta J \Theta^* = J$). More generally, there are two other ways of obtaining an alternative G from a minimal G (*i.e.*, a G with a number of columns equal to the displacement rank, r). These are

(i) Replace G by $\left[\begin{array}{ccc} G & G_1 & G_1 \end{array}\right]$, and J by $J \oplus I_\beta \oplus -I_\beta$, where G_1 is any matrix, and β is the number of columns of G_1. The resulting generator is said to be a *neutral* extension of the original minimal generator $\{G, J\}$.

(ii) Replace G by $\left[\begin{array}{cc} G & G_1 \end{array}\right]$, and J by $J \oplus \mathbf{0}_\beta$. The resulting generator is said to be a *trivial* extension of $\{G, J\}$.

These facts can be proved in various ways, but it is interesting to note that such results can be traced back to the work, in very different contexts, of Livsic [139] on colligations and of Potapov [159] on $J-$contractive complex matrix functions.

Returning to the basic definitions, the reader will find it interesting to check that a symmetric Toeplitz matrix $T = \left[c_{|i-j|}\right]_{i,j=0}^{n-1}, c_0 = 1$, has displacement rank 2 (except when all $c_i, i \neq 0$, are zero, a case we shall exclude), and a generator for T is $\{\mathbf{x_0}, \mathbf{y_0}, (1 \oplus -1)\}$, where $\mathbf{x_0} = \text{col}\{1, c_1, \ldots, c_{n-1}\}$ and $\mathbf{y_0} = \text{col}\{0, c_1, \ldots, c_{n-1}\}$ (the notation col$\{\cdot\}$ denotes a column vector with the specified entries),

$$(2.3a) \qquad T - ZTZ^* = \begin{bmatrix} 1 & 0 \\ c_1 & c_1 \\ \vdots & \vdots \\ c_{n-1} & c_{n-1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ c_1 & c_1 \\ \vdots & \vdots \\ c_{n-1} & c_{n-1} \end{bmatrix}^* .$$

Now a very interesting and quite early result on the inverse of a Toeplitz matrix is a celebrated formula of Gohberg and Semencul [92] (see also [86] and [121] for more accessible references), a special case of which states that the inverse of a real *symmetric* Toeplitz matrix T has the form

$$(2.3b) \qquad T^{-1} = \mathbf{L}(\mathbf{a})\mathbf{L}^*(\mathbf{a}) - \mathbf{L}(\mathbf{b})\mathbf{L}^*(\mathbf{b}) \ ,$$

for certain vectors $\{\mathbf{a}, \mathbf{b}\}$ (whose exact form is not relevant at the moment). But from (2.1b) and (2.2b) we now see that T^{-1} satisfies a displacement equation,

$$(2.3c) \qquad T^{-1} - ZT^{-1}Z^* = \mathbf{aa}^* - \mathbf{bb}^* = \begin{bmatrix} \mathbf{a} & \mathbf{b} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{a}^* \\ \mathbf{b}^* \end{bmatrix} .$$

In other words, though T^{-1} is (in general) not Toeplitz, it has the *same* displacement rank and inertia as T, viz., both T and T^{-1} have low displacement rank. As we mentioned earlier, it is this fact that is preserved under inversion and not the Toeplitzness of the matrix.

The striking fact is that this is a special case of a more general result on matrices with displacement structure.

LEMMA 2.1. *The displacement inertia of a Hermitian nonsingular matrix R with respect to $R - ZRZ^*$ is equal to the displacement inertia of its inverse with respect to $R^{-1} - Z^*R^{-1}Z$. That is, Inertia $(R - ZRZ^*) =$ Inertia $(R^{-1} - Z^*R^{-1}Z)$.*

Proof. Consider the identities

$$\begin{bmatrix} R & Z \\ Z^* & R^{-1} \end{bmatrix} = \begin{bmatrix} I & \mathbf{0} \\ Z^*R^{-1} & I \end{bmatrix} \begin{bmatrix} R & \mathbf{0} \\ \mathbf{0} & R^{-1} - Z^*R^{-1}Z \end{bmatrix} \begin{bmatrix} I & \mathbf{0} \\ Z^*R^{-1} & I \end{bmatrix}^*$$
$$= \begin{bmatrix} I & ZR \\ \mathbf{0} & I \end{bmatrix} \begin{bmatrix} R - ZRZ^* & \mathbf{0} \\ \mathbf{0} & R^{-1} \end{bmatrix} \begin{bmatrix} I & ZR \\ \mathbf{0} & I \end{bmatrix}^* .$$

The required result now follows by invoking Sylvester's theorem that congruence transformations preserve inertia. □

An immediate consequence of the above lemma is a simple justification of the form of the Gohberg-Semencul formula (2.3b): we know from (2.3a) that for a *symmetric* Toeplitz matrix, *Inertia* $(T - ZTZ^*) = (1,1)$. It then follows from Lemma 2.1 that the inertia of $(T^{-1} - Z^*T^{-1}Z)$ is also $(1,1)$. But $\tilde{I}T^{-1}\tilde{I} = T^{-1}$ (since $\tilde{I}T\tilde{I} = T$)

and $\tilde{I}Z^*\tilde{I} = Z$, where $\tilde{I}$ is the reverse identity matrix with ones on the antidiagonal and zeros elsewhere. Hence, *Inertia* $(T^{-1} - ZT^{-1}Z^*) = (1,1)$, which shows that we can factor $T^{-1} - ZT^{-1}Z^*$ as in (2.3c), for some column vectors $\{\mathbf{a}, \mathbf{b}\}$. Formulas for $\{\mathbf{a}, \mathbf{b}\}$ can be determined with a little more calculation (see, e.g., [49]); numerical (rather than analytical) procedures for this and more general calculations will be described in Section 5. However we should remark that there can be many forms of the Gohberg-Semencul formulas since the generator matrix $\begin{bmatrix} \mathbf{a} & \mathbf{b} \end{bmatrix}$ can be replaced by $\begin{bmatrix} \mathbf{a} & \mathbf{b} \end{bmatrix}\Theta$, where Θ is any matrix such that $\Theta J \Theta^* = J$, $J = (1 \oplus -1)$.

But a more striking aspect of the proof of Lemma 2.1 is that it is independent of the exact form of the matrix Z! This fact has several interesting implications, of which the most important is the following result, deliberately first stated in somewhat vague terms:

The Schur complements of a structured matrix R inherit its displacement structure. Moreover, a so-called generalized Schur algorithm yields the generators of the Schur complements.

The second statement will be elaborated at length later (beginning in Section 4). A more precise form of the first statement (and a generalization of Lemma 2.1) is the following. First let us replace Z with a general matrix F.

THEOREM 2.2. *The displacement inertia of a Hermitian nonsingular matrix R with respect to $R - FRF^*$ is equal to the displacement inertia of its inverse with respect to $R^{-1} - F^*R^{-1}F$. That is,*

$$\textit{Inertia}\,(R - FRF^*) = \textit{Inertia}\,(R^{-1} - F^*R^{-1}F).$$

Moreover, if we further assume that F is block-lower triangular,

$$F = \begin{bmatrix} F_1 & \mathbf{0} \\ F_2 & F_3 \end{bmatrix},$$

partition R accordingly with F,

$$R = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix},$$

assume R_{11} is invertible, and introduce the Schur complement $S = R_{22} - R_{21}R_{11}^{-1}R_{12}$, then it holds that

$$\textit{rank}\,(R_{11} - F_1 R_{11} F_1^*) \leq \textit{rank}\,(R - FRF^*),$$

$$\textit{rank}\,(S - F_3 S F_3^*) \leq \textit{rank}\,(R - FRF^*).$$

Remark: The reader may get more insight into this result by focusing on the special case $F_1 = Z = F_3$ and $F_2 = 0$, which was studied by Morf [144] and Bitmead and Anderson [26].

Proof. The proof of the first statement of the theorem is similar to that of Lemma 2.1 with Z replaced by F.

A proof of the second part of the theorem is as follows: the first inequality follows immediately since R_{11} is a submatrix of R. For the second inequality we first note that

$$\text{rank}\,(R^{-1} - F^* R^{-1} F) \;=\; \text{rank}\,(R - FRF^*).$$

We now invoke a block matrix formula for R^{-1},

$$R^{-1} = \begin{bmatrix} R_{11}^{-1} + ES^{-1}P & -ES^{-1} \\ S^{-1}P & S^{-1} \end{bmatrix}, \quad E = R_{11}^{-1}R_{12}, \quad P = R_{21}R_{11}^{-1}\,,$$

and observe that S^{-1} is a submatrix of R^{-1}. Hence,

$$\text{rank}\,(S^{-1} - F_3^* S^{-1} F_3) \;\leq\; \text{rank}\,(R^{-1} - F^* R^{-1} F).$$

But by the first result in the theorem we have

$$\text{rank}\,(S - F_3 S F_3^*) \;=\; \text{rank}\,(S^{-1} - F_3^* S^{-1} F_3).$$

We thus conclude that rank $(S - F_3SF_3^*) \;\leq\;$ rank $(R - FRF^*)$. This proof is patterned on one in Morf [145], which assumed $F = Z$; see also [26]. □

While choices of F other than Z were noted even in the early papers on displacement structure [114, 115], and further choices were later studied in [11, 24, 30, 75, 81, 88], this freedom was apparently first exploited in the work of Chun and Kailath [48, 52, 113] on fast factorization algorithms for Toeplitz- and Hankel-like matrices (see Section 5 below); later we used this freedom for various other problems in interpolation theory, matrix completion problems, and adaptive filtering (see, e.g., [29, 56, 120, 173, 176, 182]).

At this point we might note another early definition of matrix structure. Inspired by some work of Sakhnovich on integral operators, Heinig and Rost [99] studied matrices for which $\nabla R = FR + RA^*$ had low rank for suitable matrices $\{F, A\}$. Their monograph has many interesting results and references and we shall explore it in some detail later (Sections 7.2 and 8). One reason for this postponement is that we shall, in stages, introduce a more general definition that includes both the definitions mentioned so far. Another reason is that the two definitions are in many senses equivalent. Thus, in linear system theory, expressions of the form $R - FRF^*$ and $FR + RF^*$ are associated with the Lyapunov equations for discrete- and continuous-time systems, and these can be transformed into each other by well-known formulas (e.g., [108, p. 180]); another way of stating this is to say that in the function domain, these two expressions correspond to systems (operators) studied with respect to the unit circle or to the (left) half-plane. More on this later in Sections 7 and 8.

Our development of more general definitions of displacement structure arose via function theory, through a remarkable paper of Schur [185], which entered our work through the previously mentioned collaboration with P. Dewilde (see [70]). The careful reader will note, as we progress in our discussion, that the links between Schur complements, complex function theory, and structured matrices become increasingly strong.

3. The Classical Schur Algorithm. In his paper, Schur was concerned with checking whether a power series is analytic and bounded in the unit disc. While several interesting matrix results are also given in the paper (including the famous formula for the determinant of block matrices), matrix factorization is not really considered;

rather Schur speaks about reducing a quadratic form to a sum of squares by what he calls the Jacobi procedure. In fact, this is just Gaussian elimination on the coefficient matrix, as we shall explain in Section 4 below.

Schur's paper [185] was motivated by the earlier work of Toeplitz and Carathéodory on the classical trigonometric moment problem [3]. Toeplitz (1907) showed that $\{c_0, c_1, c_2, \ldots\}$ is a moment sequence (*i.e.*, $\{c_i\}_{i\geq 0}$ are Fourier coefficients of a power spectral density function) if, and only if, the Toeplitz matrices $T_k = \left[c_{|i-j|}\right]_{i,j=0}^{k}$ are positive semi-definite for all k. Carathéodory (1911) connected this condition with function theory by defining the power series

$$c(z) = c_0 + 2\sum_{i=1}^{\infty} c_i z^i \ ,$$

and showing that $T_k \geq 0$ for every k if, and only if, $c(z)$ is positive real in the open unit disc ($|z| < 1$), *i.e.*, $c(z)$ is analytic and $\text{Re}\, c(z) \geq 0$ in $|z| < 1$. Schur (1917) considered the bilinear transformation

$$s(z) = \frac{c(z) - c_0}{c(z) + c_0} \ ,$$

and noted that $c(z)$ is positive-real if, and only if, $s(z)$ is analytic and bounded by unity in $|z| < 1$ (such functions will be referred to as *Schur functions*). Moreover, and strikingly for the times, Schur presented a recursive test (rather than large determinantal expressions) for checking if a scalar function $s(z)$ is of Schur type or not.

THEOREM 3.1. *Consider the following recursive algorithm that starts with a given function $s(z)$,*

$$s_{i+1}(z) = \frac{1}{z}\,\frac{s_i(z) - \gamma_i}{1 - \gamma_i^* s_i(z)} \ , \quad s_0(z) = s(z), \quad \textit{and} \quad \gamma_i = s_i(0) \ . \tag{3.1}$$

Then the following statements hold [185]

- *$s(z)$ is analytic and bounded by unity in $|z| < 1 \iff |\gamma_i| \leq 1$ for all i.*
- *$|\gamma_i| < 1$ for $0 \leq i < n$ and $|\gamma_n| = 1$ for some n if, and only if, $s(z)$ is a finite Blaschke product of degree n.*
- *Starting with a Schur function $s(z)$, each function $s_i(z)$ is also of Schur type.*

Of course, positive-definite matrices and functions such as $c(z)$ and $s(z)$ are encountered in many applications. We may remark that the foundations of mathematical circuit theory were laid in 1931 when Brune [37] characterized the impedance functions of passive circuits as Carathéodory functions $c(z)$. Later, especially in micro wave theory, it was found to be more useful to use the equivalent scattering function descriptions, which are just the Schur functions, $s(z)$. The coefficients $\{\gamma_i\}$ that characterize Schur functions also have physical significance, leading to the name *reflection coefficients* – see Section 3.2. Positive definite (Toeplitz) matrices arise as covariance matrices of (stationary) random processes, and in this context the coefficients $\{\gamma_i\}$ can be interpreted as *partial correlation coefficients*. This interpretation is usually pursued in the context of the Levinson algorithm (see, e.g., [141, 142]); for a discussion in the Schur context see, e.g., [18, 136, 168]. For reasons of space, in this paper we shall focus only on the algebraic and function theoretic implications and generalizations of Schur's algorithm.

3.1. Array Form of the Schur Algorithm. The function recursion (3.1) is nonlinear in $s_i(z)$, so it is computationally convenient to consider an alternative form, which will play a key role in later discussions. The alternative representation is often referred to as an *array form* [109], because it consists of a sequence of elementary operations, such as rotations and shifts on an array of columns.

Let us first write, without loss of generality, $s_i(z)$ as the ratio of two power series that are analytic in $|z| < 1$ and have no common zeros,

$$s_i(z) = \frac{y_i(z)}{x_i(z)} .$$

For example, we may choose $x_i(z) = 1$ and $y_i(z) = s_i(z)$. It then follows easily that (3.1) can be rewritten in the form

$$\frac{y_{i+1}(z)}{x_{i+1}(z)} = \frac{y_i(z) - \gamma_i x_i(z)}{z\,[x_i(z) - \gamma_i^* y_i(z)]} ,$$

so that we may write

$$\begin{aligned} y_{i+1}(z) &= \alpha_i\,[y_i(z) - \gamma_i x_i(z)] \\ x_{i+1}(z) &= \alpha_i z\,[x_i(z) - \gamma_i^* y_i(z)] , \end{aligned} \tag{3.2}$$

where α_i is an arbitrary nonzero complex scalar. This additional degree of freedom in choosing α_i can be favorably exploited in order rewrite the update expressions for $y_{i+1}(z)$ and $x_{i+1}(z)$ in different forms (see, e.g., [25]). But for our purposes here a most interesting choice is to set

$$\alpha_i = \frac{1}{\sqrt{1 - |\gamma_i|^2}} ,$$

provided all the $\{\gamma_i\}$ are less than unit-magnitude, an important special case. The more general case will be studied later in Sections 4.3.1 and 7 (see also [1, 2]) where $s(z)$ is not assumed analytic in the unit circle).

A justification for the above particular choice follows by noting that we can now combine the expressions in (3.2) into the form

$$z \begin{bmatrix} x_{i+1}(z) & y_{i+1}(z) \end{bmatrix} = \begin{bmatrix} x_i(z) & y_i(z) \end{bmatrix} \Theta_i \begin{bmatrix} z & 0 \\ 0 & 1 \end{bmatrix} , \tag{3.3a}$$

where Θ_i is an elementary hyperbolic rotation determined by the coefficient γ_i,

$$\Theta_i = \frac{1}{\sqrt{1 - |\gamma_i|^2}} \begin{bmatrix} 1 & -\gamma_i \\ -\gamma_i^* & 1 \end{bmatrix}, \quad \gamma_i = \lim_{z \to 0} \frac{y_i(z)}{x_i(z)} . \tag{3.3b}$$

[The name arises from the fact that $\begin{bmatrix} a & b \end{bmatrix} \Theta_i = \begin{bmatrix} c & d \end{bmatrix}$ implies $|a|^2 - |b|^2 = |c|^2 - |d|^2$, i.e., $\begin{bmatrix} a & b \end{bmatrix}$ and $\begin{bmatrix} c & d \end{bmatrix}$ are both on the hyperbola $|x|^2 - |y|^2 =$constant.]. The rotation matrix Θ_i is well defined only when the reflection coefficients are strictly less than one in magnitude. If we introduce the signature matrix $J = (1 \oplus -1)$, it is then straightforward to verify that Θ_i is a J-unitary matrix, viz., $\Theta_i J \Theta_i^* = J$. This is a consequence of the above special choice for α_i.

Expression (3.3a) can be given an array interpretation by invoking the power series expansions of $x_i(z)$ and $y_i(z)$, say

$$\begin{aligned} x_i(z) &= x_{ii} + x_{i+1,i}z + x_{i+2,i}z^2 + \ldots \\ y_i(z) &= y_{ii} + y_{i+1,i}z + y_{i+2,i}z^2 + \ldots . \end{aligned}$$

Let us introduce the two-column (semi-infinite) matrix G_i composed of the power series coefficients of $x_i(z)$ and $y_i(z)$, viz.,

$$G_i = \begin{bmatrix} x_{ii} & y_{ii} \\ x_{i+1,i} & y_{i+1,i} \\ x_{i+2,i} & y_{i+2,i} \\ \vdots & \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{x}_i & \mathbf{y}_i \end{bmatrix} . \tag{3.4}$$

The matrices G_i will be called generator matrices, for reasons that will be made clear later, in Section 4.2. Recursion (3.3a) starts by multiplying the prearray G_0 by Θ_0. Because of the way γ_0 was defined, this results in a zero element in the first entry of the second column of the postarray, denoted by $\bar{G}_0$. That is, with $\gamma_0 = \frac{y_{00}}{x_{00}}$,

$$G_0\Theta_0 = \begin{bmatrix} x_{00} & y_{00} \\ x_{10} & y_{10} \\ x_{20} & y_{20} \\ x_{30} & y_{30} \\ \vdots & \vdots \end{bmatrix} \Theta_0 = \begin{bmatrix} \bar{x}_{00} & 0 \\ \bar{x}_{10} & \bar{y}_{10} \\ \bar{x}_{20} & \bar{y}_{20} \\ \bar{x}_{30} & \bar{y}_{30} \\ \vdots & \vdots \end{bmatrix} = \bar{G}_0 .$$

Next multiplying by

$$\begin{bmatrix} z & 0 \\ 0 & 1 \end{bmatrix}$$

corresponds to shifting down the first column of the postarray $\bar{G}_0$ by one element while keeping unaltered the second column, which leads to

$$\bar{G}_0 = \begin{bmatrix} \bar{x}_{00} & 0 \\ \bar{x}_{10} & \bar{y}_{10} \\ \bar{x}_{20} & \bar{y}_{20} \\ \bar{x}_{30} & \bar{y}_{30} \\ \vdots & \vdots \end{bmatrix} \overset{shift}{\longrightarrow} \begin{bmatrix} 0 & 0 \\ \bar{x}_{00} & \bar{y}_{10} \\ \bar{x}_{10} & \bar{y}_{20} \\ \bar{x}_{20} & \bar{y}_{30} \\ \vdots & \vdots \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ x_{11} & y_{11} \\ x_{21} & y_{21} \\ x_{31} & y_{31} \\ \vdots & \vdots \end{bmatrix} = \begin{bmatrix} 0 \quad 0 \\ G_1 \end{bmatrix} .$$

Notice that we have, for convenience, renamed the entries of the resulting matrix and defined G_1. It is also easy to see that the entries of the i^{th} row of G_1 depend only on the entries of rows 0 through i of $G_0 = G$.

This completes the first step of (3.3a). The recursive procedure now continues as follows: compute γ_1 as the ratio of y_{11} and x_{11}, multiply the prearray G_1 by Θ_1 in order to introduce a zero in the first entry of the second column of the postarray $\bar{G}_1$, shift down the first column of $\bar{G}_1$ and so on. Schematically, we have the following simple array picture:

$$G_i = \begin{bmatrix} x & x \\ x & x \\ x & x \\ \vdots & \vdots \end{bmatrix} \overset{\Theta_i(\gamma_i)}{\longrightarrow} \begin{bmatrix} x' & 0 \\ x' & x' \\ x' & x' \\ \vdots & \vdots \end{bmatrix} \overset{shift}{\longrightarrow} \begin{bmatrix} 0 & 0 \\ x'' & x' \\ x'' & x' \\ \vdots & \vdots \end{bmatrix} = \begin{bmatrix} 0 \quad 0 \\ G_{i+1} \end{bmatrix} . \tag{3.5a}$$

In words:

- Use the top row of G_i to define a $J-$unitary matrix Θ_i that transforms this row to the form $\left[\begin{array}{cc} x' & 0 \end{array}\right]$;
- Multiply G_i by Θ_i and keep the second column;
- Shift down the first column of $G_i\Theta_i$;
- These two operations result in G_{i+1}.

In matrix language, (3.5a) can be stated as

$$\text{(3.5b)} \quad \left[\begin{array}{cc} 0 & 0 \\ & G_{i+1} \end{array}\right] = \mathcal{Z} G_i \Theta_i \left[\begin{array}{cc} 1 & 0 \\ 0 & 0 \end{array}\right] + G_i \Theta_i \left[\begin{array}{cc} 0 & 0 \\ 0 & 1 \end{array}\right] , \quad G_0 = \left[\begin{array}{cc} \mathbf{x}_0 & \mathbf{y}_0 \end{array}\right],$$

where $\mathcal{Z}$ denotes the (semi-infinite) lower triangular shift matrix with ones on the first subdiagonal and zeros elsewhere.

We have thus presented three different ways of writing the Schur recursion: the nonlinear form (3.1) given by Schur, the linearized function form (3.3a), which can also be rewritten in the so-called generating function form (more on this later in Section 8),

$$\text{(3.6a)} \qquad zG_{i+1}(z) = G_i(z)\Theta_i(z) \ ,$$

where we have defined

$$\text{(3.6b)} \qquad G_i(z) = \left[\begin{array}{cc} x_i(z) & y_i(z) \end{array}\right] \quad \text{and} \quad \Theta_i(z) = \Theta_i \left[\begin{array}{cc} z & 0 \\ 0 & 1 \end{array}\right],$$

and finally the array or matrix form (3.5b), which will be greatly extended in future sections.

We have described the generating function and matrix array forms of the Schur algorithm in some detail for two reasons. First, the array form, (3.5a) or (3.5b), can be used to show (a result implicit in Schur's paper) that the Schur algorithm directly yields the triangular factorization, not only of positive-definite Toeplitz matrices, but of a more general class of structured matrices. More specifically, and for notational convenience, if we also denote the first n entries of the *semi-infinite* columns $\mathbf{x}_0$ and $\mathbf{y}_0$ by $\mathbf{x}_0$ and $\mathbf{y}_0$, respectively, then the first n steps of Schur's recursion will be shown (in the next section) to provide the triangular factorization of $n \times n$ positive-definite structured matrices R of the form

$$\text{(3.7)} \qquad R = \mathbf{L}(\mathbf{x}_0)\mathbf{L}^*(\mathbf{x}_0) - \mathbf{L}(\mathbf{y}_0)\mathbf{L}^*(\mathbf{y}_0) \ ,$$

which we may recall (cf. (2.1b)–(2.2b)) corresponds to R having displacement inertia $\{1, 1\}$, i.e.,

$$R - ZRZ^* = \mathbf{x}_0\mathbf{x}_0^* - \mathbf{y}_0\mathbf{y}_0^* \ .$$

In the special case,

$$\mathbf{x}_0 = \text{col}\{1, c_1, c_2, \ldots, c_{n-1}\}, \ \ \mathbf{y}_0 = \text{col}\{0, c_1, c_2, \ldots, c_{n-1}\} \ ,$$

we saw earlier (see (2.3a)) that $R = [c_{i-j}]_{i,j=0}^{n-1}$ is Toeplitz, so that (3.7) is a natural generalization of Toeplitz matrices, which we have called quasi-Toeplitz [130]. Another reason for the name is an identity also found in Schur's paper that R in (3.7) can be expressed in the form

$$R = \mathbf{L}(\mathbf{x}_0 - \mathbf{y}_0) \ T \ \mathbf{L}^*(\mathbf{x}_0 - \mathbf{y}_0) \ ,$$

where T is the easily determined Toeplitz matrix,

$$T = \frac{1}{2}\left[\mathbf{L}^{-1}(\mathbf{x}_0 - \mathbf{y}_0)\mathbf{L}(\mathbf{x}_0 + \mathbf{y}_0) + \mathbf{L}^{*}(\mathbf{x}_0 + \mathbf{y}_0)\mathbf{L}^{-*}(\mathbf{x}_0 - \mathbf{y}_0)\right] .$$

That is, every R in (3.7) is congruent, via a special congruence relation, to a Toeplitz matrix. The paper [65] contains more on such matrices.

In fact, we can generalize the various forms of the Schur algorithm to show that the algorithm and the associated matrix factorization result can be extended to matrices with any displacement inertia $\{p, q\}$, and in fact to even more general cases where Z is replaced by an arbitrary lower-triangular matrix F. Perhaps the most direct way of doing this is to consider afresh the problem of triangular matrix factorization, which is basically effected by the Gaussian elimination technique. Adding displacement structure allows one to speed up the Gaussian elimination procedure and can be shown to lead to the above algorithm of Schur and to various generalizations.

3.2. Cascade/Transmission Line Interpretations and Inverse Scattering. Before proceeding to do that, however, it is worthwhile to draw attention to a cascade network/transmission line interpretation of the Schur algorithm, discussed in some detail in references [34, 35, 109]. We shall describe simple physical arguments based on causality that show that the Schur algorithm arises as perhaps the most natural way of solving the inverse scattering problem for discrete transmission lines. For those familiar with the signal flow and block diagram representations used by engineers, the transmission line interpretation gives a lot of insight, and suggests new results and new proofs, for a surprisingly diverse set of problems. For example, references [34, 35] show how the transmission line picture gives nice interpretations of the classical Gelfand-Levitan, Marchenko and Krein equations, and in fact yields various generalizations thereof; reference [36] discusses discrete Schrodinger equations. Several other applications will be indicated later.

We shall first explain how the generator recursion (3.5b) can be graphically depicted as a cascade of elementary sections as shown in Figure 3.1, where we have defined $\gamma_i^c = \sqrt{1 - |\gamma_i|^2}$.

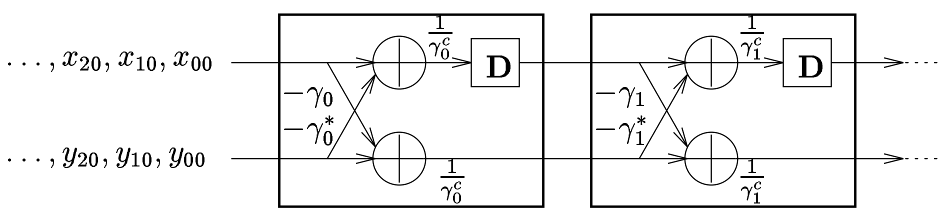

FIG. 3.1. *The feedforward structure (cascade network) associated with Schur's recursion.*

Each section consists of a hyperbolic rotation Θ_i followed by a unit-time delay (or storage) element denoted by $\mathbf{D}$. The cascade represents a feedforward (and pipelineable) implementation of the array algorithm, where the entries of the two columns of G_0 ($\mathbf{x_0}$ and $\mathbf{y_0}$) are available at the input lines of the first section. By reversing the direction of flow in the lower line, we get a physical lossless discrete-time transmission-line, as shown in Figure 3.2, where each section is now composed of a unitary gain matrix Σ_i ($\Sigma_i \Sigma_i^* = I$) followed by a unit-time delay element,

$$\Sigma_i = \begin{bmatrix} \gamma_i^c & \gamma_i \\ -\gamma_i^* & \gamma_i^c \end{bmatrix} .$$

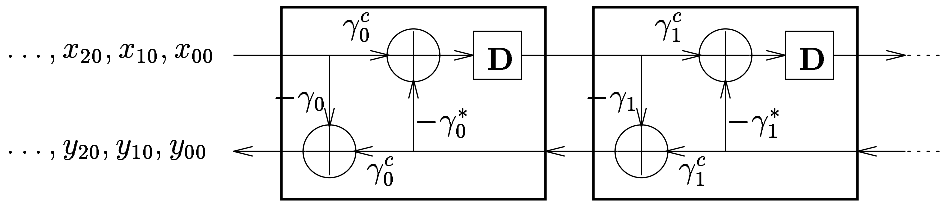

FIG. 3.2. *The feedback structure (transmission line) associated with Schur's recursion.*

A physical motivation and derivation of a layered medium structure as in Figure 3.2 can be given by showing that it corresponds to a discretization of the wave propagation (or telegrapher's) equations in an electromagnetic medium with varying local impedance; the relevant details can be found, for example, in [109]. The name *reflection coefficients* for the Schur coefficients $\{\gamma_i\}$ arises from the picture in Figure 3.2; at each section, a fraction γ_i of the incoming signal is reflected and the rest, γ_i^c, is transmitted.

The following so-called inverse-scattering problem can then be associated with such layered media: given an arbitrary pair of input-response sequences of a layered medium as in Figure 3.2, say $\{\ldots, x_{20}, x_{10}, x_{00}\}$ and $\{\ldots, y_{20}, y_{10}, y_{00}\}$, determine the medium (or reflection) parameters $\{\gamma_0, \gamma_1, \gamma_2, \ldots\}$, under the assumption that the line was initially quiescent. As mentioned above, this is a prototype of a famous problem, which has been attacked in many ways. The most widely known are methods using special choices of input sequences, based on which the inversion problem is shown to be equivalent to the solution of sets of linear equations, special forms of which are famous as the Gelfand-Levitan, Marchenko and Krein equations of classical inverse scattering theory (see [34, 35]). Here we wish to indicate how a very natural solution of the inverse problem follows as a consequence of the easily verifiable "causality" of the transmission-line; moreover that this is just the Schur algorithm.

First remark that the storage elements **D** in the upper line act as unit-time delay elements between two successive layers. The subindexes in $\{x_{j0}, y_{j0}\}$ indicate that these are the input and response samples available at the left-end of the line (or at the first section Σ_0) at time j. Now, the output at time 0, y_{00}, is clearly equal to $\gamma_0 x_{00}$ since no signal is fed back through the bottom line due to zero initial conditions in the layered structure and to the presence of the delay element between the first two sections. This argument can easily be extended to prove that, for a quiescent transmission-line, the output at time j, namely y_{j0}, depends only on the inputs up to and including time j, viz., $\{x_{j0}, x_{j-1,0}, \ldots, x_{10}, x_{00}\}$.

Now consider again Figure 3.2 and assume that x_{00} enters the line at time 0. The causality property shows that there will be no left-propagating wave response from the line for at least one-time unit. Therefore, we can conclude that

$$\gamma_0 = \frac{y_{00}}{x_{00}} \, .$$

At time 1, x_{10} will enter the line and, by definition, x_{11} will be the right-going input of the second section of the discrete transmission line, and y_{11} will be the left-going output. (The subindexes in $\{x_{ji}, y_{ji}\}$ indicate signals at time j that are available at the left-end of the i^{th} section). As before, the line being initially at rest and the delay structure will mean that there will be no left-going input to this (second) section until at least time 2. Therefore, we can conclude that

$$\gamma_1 = \frac{y_{11}}{x_{11}} \ ,$$

and in general $\gamma_i = y_{ii}/x_{ii}$.

This is nice but, of course, we are only given the input-response sequences at the left-most end of the line, viz., $\{x_{j0}, y_{j0},\ j \geq 0\}$ not the waves $\{x_{ji}, y_{ji},\ j \geq i\}$ at any intermediate section i. Here is where we can appeal to the equivalent (direct) cascade of Figure 3.1, which shows how to use the $\{\Theta_i\}$ to propagate the $\{x_{j0}, y_{j0},\ j \geq 0\}$ pair into the line. In other words, having γ_0, we can form Θ_0 and then use it to obtain (cf. (3.5b))

$$\begin{bmatrix} 0 & 0 \\ x_{11} & y_{11} \\ x_{21} & y_{21} \\ x_{31} & y_{31} \\ \vdots & \vdots \end{bmatrix} = \begin{bmatrix} x_{00} & y_{00} \\ x_{10} & y_{10} \\ x_{20} & y_{20} \\ x_{30} & y_{30} \\ \vdots & \vdots \end{bmatrix} \Theta_0 \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} + \mathcal{Z} \begin{bmatrix} x_{00} & y_{00} \\ x_{10} & y_{10} \\ x_{20} & y_{20} \\ x_{30} & y_{30} \\ \vdots & \vdots \end{bmatrix} \Theta_0 \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} .$$

With $\{x_{j1}, y_{j1},\ j \geq 1\}$ available, we can find γ_1, and then obtain $\{x_{j2}, y_{j2},\ j \geq 2\}$ by applying Θ_1 to $\{x_{j1}, y_{j1},\ j \geq 1\}$, and so on.

In summary, this identification scheme takes the scattering data $\{x_{j0}, y_{j0}\}$ and uses them to identify the first section of the medium; and it then replaces the original data by a set of "synthetic" scattering data corresponding to the yet undetermined part of the medium. Hence, the name *layer peeling* is often used for this procedure, which the reader can check is precisely the array form of the Schur algorithm. It is striking how naturally it arises as a direct physical solution of the inverse scattering problem. Of course, it therefore may not be a surprise to learn that geophysicists had already discovered this direct method, which they dubbed as *dynamic deconvolution* or sometimes, as *downward continuation* (see, e.g., [38, 167]). The alternative linear equations method turns out to have the physical interpretation of *layer adjoining*, rather than layer peeling. However, we shall forego a description here, referring interested readers to [34, 35], and to [162, 163] and [33, 54, 55], where the ideas are applied to problems in digital filter design and algebraic coding theory, respectively; it is also explained why the layer-peeling algorithms are better suited to parallel implementation. Transmission line interpretations will be encountered again in studying interpolation problems in Sec. 6.1.1.

4. Several Extensions of the Array Form Schur Algorithm. We now return to our earlier claim that the classical Schur algorithm factors structured matrices as in (3.7). The above transmission line picture can be used to give a physical proof of this fact based on the energy conservation principle – see [109, 112]. However, for further development, it will be better to pursue a more algebraic route. For this, we shall first review the Gaussian elimination procedure and then proceed to show that it collapses to Schur's recursion when the structure implied by (3.7) is properly incorporated into the calculations. Once this is established, we shall move on to gradually generalize Schur's recursion and to consider more general classes of structured matrices.

4.1. Triangular Factorization. The well-known procedures for the triangular factorization of a strongly regular (Hermitian) matrix (*i.e.,* a matrix with nonzero leading minors), $R = [r_{mj}]_{m,j=0}^{n-1}$, go by many names: Jacobi, Cholesky, Schur reduction, etc. In fact, they are all effectively just Gaussian elimination (see, e.g., [189]).

It is well-known that the assumption of strong regularity of R guarantees the existence of a triangular factorization of the form $R = LD^{-1}L^*$, where L is a lower-triangular matrix with the same diagonal entries as the diagonal matrix D (see also [82, 94]). Equivalently, if we introduce the normalization

$$\tilde{L} = LD^{-1} ,$$

then we can also express R in the alternative factored form $R = \tilde{L}D\tilde{L}^*$, where the lower triangular factor $\tilde{L}$ now has unit diagonal entries. This latter factorization is perhaps more common but, in any case, the columns of L and $\tilde{L}$ are simply scaled versions of each other and it therefore does not matter whether we work with L or $\tilde{L}$. Here we prefer to work with L, because, as suggested by our later expression (6.3a), its columns will have a natural interpretation as the states of first-order sections.

Now, the columns of L and the diagonal entries of D can be recursively computed as follows. Let l_0 and d_0 denote the first column and the $(0,0)$ entry of R, respectively. If we subtract from R the outer product $l_0 d_0^{-1} l_0^*$, we obtain a new matrix with an identically zero first row and column. That is,

$$\text{(4.1a)} \qquad R - l_0 d_0^{-1} l_0^* = \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & R_1 \end{bmatrix} = \tilde{R}_1 \ ,$$

where $R_1 = \left[r_{mj}^{(1)} \right]_{m,j=0}^{n-2}$ is called the Schur complement of r_{00} in R. In the past (see, e.g., [133]), we have called (4.1a) a Schur reduction step; it can now be repeated 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. Each further step corresponds to a recursion of the form

$$\text{(4.1b)} \qquad \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & R_{i+1} \end{bmatrix} = R_i - l_i d_i^{-1} l_i^* \ ,$$

where $d_i = r_{00}^{(i)}$ (the $(0,0)$ entry of the i^{th} Schur complement R_i), and l_i denotes the first column of R_i.

Hence, starting with the $n \times n$ matrix R and performing n consecutive Schur complement steps, we obtain the triangular factorization of R, viz.,

$$\text{(4.1c)} \quad R = l_0 d_0^{-1} l_0^* + \begin{bmatrix} 0 \\ l_1 \end{bmatrix} d_1^{-1} \begin{bmatrix} 0 \\ l_1 \end{bmatrix}^* + \begin{bmatrix} 0 \\ 0 \\ l_2 \end{bmatrix} d_2^{-1} \begin{bmatrix} 0 \\ 0 \\ l_2 \end{bmatrix}^* \ldots = LD^{-1}L^* \ ,$$

where $D = \text{diagonal}\{d_0, \ldots, d_{n-1}\}$, and the (nonzero parts of the) columns of the lower triangular matrix L are $\{l_0, \ldots, l_{n-1}\}$. This procedure requires $O(n^3)$ elementary operations (additions and multiplications).

The connection with triangular factorization can also be seen by rewriting (4.1a) as

$$\begin{aligned} R &= l_0 d_0^{-1} l_0^* + \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & R_1 \end{bmatrix} , \\ &= \begin{bmatrix} l_0 & \begin{matrix} 0 \\ I_{n-1} \end{matrix} \end{bmatrix} \begin{bmatrix} d_0^{-1} & \\ & R_1 \end{bmatrix} \begin{bmatrix} l_0 & \begin{matrix} 0 \\ I_{n-1} \end{matrix} \end{bmatrix}^* . \end{aligned}$$

If we partition the entries of l_0 as $l_0 = \text{col}\{d_0, t_0\}$, where t_0 is also a column vector, then the last equality can be written as

$$R = \begin{bmatrix} 1 & 0 \\ t_0 d_0^{-1} & I_{n-1} \end{bmatrix} \begin{bmatrix} d_0 & \\ & R_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_0 d_0^{-1} & I_{n-1} \end{bmatrix}^* ,$$

or, equivalently, as

$$\begin{bmatrix} 1 & 0 \\ -t_0 d_0^{-1} & I_{n-1} \end{bmatrix} R = \begin{bmatrix} d_0 & \mathbf{0} \\ t_0 & R_1 \end{bmatrix}^* .$$

This explains why (4.1a), which we called Schur reduction, is the same as Gaussian elimination. We should note that the above procedure can readily be extended to strongly-regular non-Hermitian matrices yielding the so-called LDU decompositions; according to Stewart [189], it was Alan Turing who in 1948 first made explicit the connection between Gaussian elimination and triangular factorization.

We may finally remark that the requirement of strong regularity can be relaxed. Indeed, if it happens that the $(0,0)$ entry of R_i is zero and if the first column of R_i is also identically zero, for some i, then we can still proceed with the Gaussian elimination procedure by setting, for example, $d_i = 1$ and $l_i = \text{col}\{0, \mathbf{0}\}$. This trivial case will be excluded from our future discussions. Before proceeding, we note that the quantities $\{d_i, l_i\}$ will often be used below, so their definitions are worth keeping in mind.

4.2. Incorporating Quasi-Toeplitz Structure. We now impose a quasi Toeplitz structure on R, as defined in (3.7). That is, we consider *positive-definite* matrices R that satisfy the displacement equation

(4.2a) $$R - ZRZ^* = \begin{bmatrix} \mathbf{x}_0 & \mathbf{y}_0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 & \mathbf{y}_0 \end{bmatrix}^* = G_0 J G_0^*, \text{ say} .$$

As expected from Theorem 2.2, we shall see that the Schur complement R_1 inherits the displacement structure of R, *i.e.,* that there exists G_1 of size $(n-1) \times 2$ such that

(4.2b) $$R_1 - Z R_1 Z^* = G_1 J G_1^*,$$

where, by an excusable abuse of notation, the shift matrix Z is now of size $n-1$. Moreover, G_1 can be directly computed from G_0 (without explicit knowledge of R), which will require only $O(n)$ elementary computations, unlike the $O(n^2)$ needed if we work with R.

Finally, we shall see that the procedure for going from G_0 to G_1 is exactly that given by the first step of the Schur recursion (3.5a), except that now we need only work with the first n rows of G_0. It is to make this connection that we assume (in this section) that R is positive-definite and so, of course, strongly regular.

The above claims can be proved in several different ways, e.g., by checking that the first n rows of the recursion (3.5b) define a matrix G_1 that satisfies (4.2b). We shall proceed as follows. We first use the nonuniqueness of G_0 in (4.2a) to replace it by a matrix in what we shall call *proper* form; this will then simplify the calculation of G_1.

More specifically note that we can replace G_0 in (4.2a) by $G_0\Theta_0$, where Θ_0 is any J-unitary matrix, *i.e.* , $\Theta_0 J \Theta_0^* = J$. In particular, we can choose

$$\Theta_0 = \frac{1}{\sqrt{1-|\gamma_0|^2}} \begin{bmatrix} 1 & -\gamma_0 \\ -\gamma_0^* & 1 \end{bmatrix},$$

where γ_0 is the ratio of the entries of the first row of $G_0 = \begin{bmatrix} x_{00} & y_{00} \end{bmatrix}$, *i.e.*,

$$\gamma_0 = y_{00}/x_{00}.$$

The fact that $|\gamma_0| < 1$ follows from the observation that the $(0,0)$ entry of R can be seen from (4.2a) to be equal to $|x_{00}|^2 - |y_{00}|^2$, which has to be positive because R is positive definite. It is now easy to verify that

$$\begin{bmatrix} x_{00} & y_{00} \end{bmatrix} \Theta_0 = \begin{bmatrix} \delta_0 & 0 \end{bmatrix}$$

where $|\delta_0|^2 = |x_{00}|^2 - |y_{00}|^2$, so that $G_0\Theta_0 = \bar{G}_0$ will have the form

$$G_0\Theta_0 = \bar{G}_0 = \begin{bmatrix} \delta_0 & 0 \\ \bar{x}_{11} & \bar{y}_{11} \\ \bar{x}_{21} & \bar{y}_{21} \\ \vdots & \vdots \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{x}}_0 & \bar{\mathbf{y}}_0 \end{bmatrix}, \quad \text{say.}$$

A generator of this form will be said to be *proper*. One consequence of properness is that from the equation

$$R - ZRZ^* = G_0JG_0^* = \bar{G}_0J\bar{G}_0^*$$

we can conclude that l_0, the first column of R, is given by

$$l_0 = \bar{G}_0 \begin{bmatrix} \delta_0^* \\ 0 \end{bmatrix} = \delta_0^*\bar{\mathbf{x}}_0$$

and also that d_0, the $(0,0)$ entry of R, is given by $d_0 = |\delta_0|^2$. Now we are ready to explore the displacement structure of

$$\widetilde{R}_1 = R - l_0d_0^{-1}l_0^* = \begin{bmatrix} 0 & 0 \\ 0 & R_1 \end{bmatrix}.$$

In fact,

$$\begin{aligned} \tilde{R}_1 - Z\tilde{R}_1Z^* &= R - l_0d_0^{-1}l_0^* - Z(R_0 - l_0d_0^{-1}l_0^*)Z^* \\ &= \bar{G}_0J\bar{G}_0^* - l_0d_0^{-1}l_0^* + Zl_0d_0^{-1}l_0^*Z^* \\ &= \bar{\mathbf{x}}_0\bar{\mathbf{x}}_0^* - \bar{\mathbf{y}}_0\bar{\mathbf{y}}_0^* - \bar{\mathbf{x}}_0\bar{\mathbf{x}}_0^* + Z\bar{\mathbf{x}}_0\bar{\mathbf{x}}_0^*Z^* \\ &= \begin{bmatrix} Z\bar{\mathbf{x}}_0 & \bar{\mathbf{y}}_0 \end{bmatrix} J \begin{bmatrix} Z\bar{\mathbf{x}}_0 & \bar{\mathbf{y}}_0 \end{bmatrix}^* \\ &= \begin{bmatrix} 0 \\ G_1 \end{bmatrix} J \begin{bmatrix} 0 \\ G_1 \end{bmatrix}^*, \quad \text{say,} \end{aligned}$$

where the last equality uses the fact that the first entries of the columns $\{Z\bar{\mathbf{x}}_0, \bar{\mathbf{y}}_0\}$ are zero. Now from the definition of $\tilde{R}_1$ we we conclude that

$$R_1 - ZR_1Z^* = G_1JG_1^*,$$

so that $\{G_1, J\}$ is a generator for the Schur complement R_1. Note also that G_1 is obtained directly from G_0 via the operations

$$G_0\Theta_0 = \bar{G}_0 = \begin{bmatrix} \bar{\mathbf{x}}_0 & \bar{\mathbf{y}}_0 \end{bmatrix}, \qquad \begin{bmatrix} 0 \\ G_1 \end{bmatrix} = \begin{bmatrix} Z\bar{\mathbf{x}}_0 & \bar{\mathbf{y}}_0 \end{bmatrix},$$

or, equivalently,

$$\begin{bmatrix} 0 \\ G_1 \end{bmatrix} = Z\bar{G}_0 \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \bar{G}_0 \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$
$$= ZG_0\Theta_0 \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + G_0\Theta_0 \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

But this is exactly what we get from the first n rows of the array form of Schur's recursion (3.5a) or (3.5b).

Now we can clearly repeat this procedure to get the next Schur complement and so on. In summary, we have the following result.

THEOREM 4.1 (The Classical Schur Algorithm in Array Form). *Consider an $n \times n$ positive-definite quasi-Toeplitz matrix R,*

$$R - ZRZ^* = G \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} G^* , \quad Z \text{ is } n \times n.$$

The successive Schur complements of R with respect to its leading $i \times i$ submatrices are also quasi-Toeplitz,

$$R_i - ZR_iZ^* = G_iJG_i^* , \quad Z \text{ is now } (n-i) \times (n-i),$$

with generator matrices G_i obtained by the following recursive construction: start with $G_0 = G$ and repeat for $i \geq 0$:

1. *Let $g_i = \begin{bmatrix} x_{ii} & y_{ii} \end{bmatrix}$ denote the first row of G_i and define the hyperbolic rotation Θ_i,*

$$\Theta_i = \frac{1}{\sqrt{1-|\gamma_i|^2}} \begin{bmatrix} 1 & -\gamma_i \\ -\gamma_i^* & 1 \end{bmatrix}, \quad \gamma_i = \frac{y_{ii}}{x_{ii}} ,$$

that rotates g_i to the form

$$g_i\Theta_i = \begin{bmatrix} \delta_i & 0 \end{bmatrix} .$$

2. *Multiply G_i by Θ_i and keep the last column of $G_i\Theta_i$;*
3. *Shift down the first column of $G_i\Theta_i$ by one position;*
4. *This provides G_{i+1}.*

In matrix form we have

$$\begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = ZG_i\Theta_i \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + G_i\Theta_i \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} .$$

Moreover, the (nonzero parts of the) columns of the triangular factor L of $R = LD^{-1}L^*$, and of $\tilde{L}$ in $R = \tilde{L}D\tilde{L}^*$, are given by

$$l_i = \delta_i^* G_i\Theta_i \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad \tilde{l}_i = \frac{1}{\delta_i} G_i\Theta_i \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

while

$$D = \text{diagonal}\,\{d_i\} \ , \ \ d_i = |\delta_i|^2 .$$

This theorem justifies our earlier claim that the classical Schur algorithm gives a recursive procedure for the triangular factorization of (positive-definite) quasi-Toeplitz matrices, and in particular, of Toeplitz matrices. In linear algebra, the special result for Toeplitz matrices has been often rediscovered in different contexts, *e.g.*, by Bareiss [22], Morf [144], Rissanen [166], LeRoux and Gueguen [129]; the connection with Schur's work was first made in [70] and more explicitly in [133].

4.3. Extensions of Displacement Structure. The striking fact is that the above results can be extended to matrices with quite general displacement structure. As first noted in [52], an extension that can be immediately handled by the arguments given so far is that of $n \times n$ Hermitian positive-definite matrices R that obey the displacement equation

$$R - FRF^* = G \underbrace{\begin{bmatrix} I_p & \mathbf{0} \\ \mathbf{0} & -I_q \end{bmatrix}}_{J} G^* \ , \quad F \text{ strictly lower triangular.}$$

The choice $F = Z, p = 1 = q$ and $G = \begin{bmatrix} \mathbf{x}_0 & \mathbf{y}_0 \end{bmatrix}$ corresponds to (4.2a). It turns out that the procedure of Theorem 4.1 generalizes in a very natural way to this case: the dimensions of the successive generators G_i change from $(n-1) \times 2$ to $(n-1) \times r$, $r = p + q$, and the generator recursion changes to

$$\text{(4.3a)} \qquad \begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = F_i G_i \Theta_i \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{r-1} \end{bmatrix} + G_i \Theta_i \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & I_{r-1} \end{bmatrix} ,$$

where F_i is the submatrix obtained from F_{i-1} by deleting its first row and column (with $F_0 = F$), and Θ_i is any $J-$unitary rotation matrix that annihilates all the entries of the top row of G_i, denoted by g_i, except for a single entry in the first position,

$$\text{(4.3b)} \qquad g_i \Theta_i = \begin{bmatrix} \delta_i & 0 & \ldots & 0 \end{bmatrix} .$$

Section 4.4 further expands on this particular point and shows that such a Θ_i can always be found. That is, G_{i+1} in (4.3a) is obtained as follows:

1. Multiply G_i by Θ_i;
2. Keep the last $r - 1$ columns of $G_i \Theta_i$;
3. Multiply the first column of $G_i \Theta_i$ by F_i.

Moreover, the columns of the triangular factor of R are given by

$$\text{(4.3c)} \qquad l_i = \delta_i^* G_i \Theta_i \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} , \ \ d_i = |\delta_i|^2 .$$

The proof of the above facts follows precisely the same steps as in Theorem 4.1 and will not be given here. The only issue that remains to be explained is how to achieve (4.3b). This can be done in several ways, with one method being described in detail in Section 4.4.

4.3.1. Replacing Positive-Definiteness with Strong Regularity. The assumption of positive-definiteness can be replaced in a fairly straightforward way by the assumption of strong regularity. As one may expect, in this case (4.3b) will be slightly modified (as in (4.5b) below). Somewhat more work is required to allow F to be lower-triangular rather than strictly lower triangular. So let us now consider strongly regular $n \times n$ Hermitian matrices that obey the displacement equation

$$\text{(4.4a)} \qquad R - FRF^* = G \underbrace{\begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix}}_{J} G^* \ , \ p + q = r, \ F \ \text{ lower triangular} ,$$

and where it is further assumed that the diagonal entries of F satisfy

$$\text{(4.4b)} \qquad (1 - f_i f_j^*) \neq 0$$

for all i, j, so as to guarantee a unique solution R of (4.4a). The triangular factorization of structured matrices as in (4.4a) can be obtained as follows.

THEOREM 4.2 (A Generalized Schur Algorithm in Array Form). *Consider an $n \times n$ strongly-regular matrix R as in (4.4a) and (4.4b). The successive Schur complements of R with respect to its leading $i \times i$ submatrices are also structured,*

$$R_i - F_i R_i F_i^* = G_i J G_i^* \ , \quad F_i \text{ is } (n-i) \times (n-i), \tag{4.5a}$$

where F_i is the submatrix obtained from F_{i-1} by deleting its first row and column, and the generator matrices G_i obey the following recursive construction: start with $G_0 = G$, $F_0 = F$ and repeat for $i \geq 0$:

1. *At step i we have G_i and F_i. Let g_i denote the top row of G_i;*
2. *Choose any $J-$unitary rotation matrix Θ_i that annihilates all the entries of g_i except for a single entry: this entry has to be in the first p positions if $g_i J g_i^* > 0$, and in the last q positions if $g_i J g_i^* < 0$ (strong regularity rules out the case $g_i J g_i^* = 0$). So let (this is further detailed in Section 4.4)*

$$g_i \Theta_i = \left[\begin{array}{ccccccc} 0 & \ldots & 0 & \delta_i & 0 & \ldots & 0 \end{array} \right] , \tag{4.5b}$$

where the nonzero entry is, say, in the j^{th} position.

3. *Compute G_{i+1} as follows:*
   3.1. *Multiply G_i by Θ_i;*
   3.2. *Keep all columns of $G_i\Theta_i$ unchanged except for the j^{th} column;*
   3.3. *Multiply the j^{th} column of $G_i\Theta_i$ by Φ_i, where Φ_i is the "Blaschke" matrix,*

$$\Phi_i = (F_i - f_i I_{n-i})(I_{n-i} - f_i^* F_i)^{-1} \ .$$

   3.4. *This provides G_{i+1}.*

Note that for strictly lower F we have $f_i = 0$ and, consequently, Φ_i collapses to F_i. In matrix notation, the procedure is

$$\left[\begin{array}{c} \mathbf{0} \\ G_{i+1} \end{array} \right] = \Phi_i G_i \Theta_i \left[\begin{array}{ccc} \mathbf{0}_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{r-j-1} \end{array} \right] + G_i \Theta_i \left[\begin{array}{ccc} I_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{r-j-1} \end{array} \right] . \tag{4.5c}$$

4. *The triangular factorization, $R = LD^{-1}L^*$, is determined by*

$$l_i = \delta_i^* (I_{n-i} - f_i^* F_i)^{-1} G_i \Theta_i J \left[\begin{array}{c} \mathbf{0} \\ 1 \\ \mathbf{0} \end{array} \right] \quad \text{and} \quad d_i = \frac{J_{jj} |\delta_i|^2}{1 - |f_i|^2} \ . \tag{4.5d}$$

Proof. The proof can be carried out by following similar steps to what we have done before, as we readily demonstrate for the first Schur complement

$$\tilde{R}_1 = \left[\begin{array}{cc} 0 & \mathbf{0} \\ \mathbf{0} & R_1 \end{array} \right] = R - l_0 d_0^{-1} l_0^* \ .$$

The claim is that R_1 is also structured with a generator equal to the matrix G_1 obtained after the first step of (4.5c), viz., $R_1 - F_1 R_1 F_1^* = G_1 J G_1^*$. We can do this by showing that

$$\tilde{R}_1 - F \tilde{R}_1 F^* = \left(R - l_0 d_0^{-1} l_0^* \right) - F \left(R - l_0 d_0^{-1} l_0^* \right) F^* = \left[\begin{array}{c} \mathbf{0} \\ G_1 \end{array} \right] J \left[\begin{array}{c} \mathbf{0} \\ G_1 \end{array} \right]^* ,$$

or equivalently, replacing $R - FRF^*$ by GJG^*, that

$$\underbrace{GJG^*}_{\bar{G}J\bar{G}^*} - l_0 d_0^{-1} l_0^* + F l_0 d_0^{-1} l_0^* F^* = \begin{bmatrix} \mathbf{0} \\ G_1 \end{bmatrix} J \begin{bmatrix} \mathbf{0} \\ G_1 \end{bmatrix}^* . \tag{4.6}$$

We may as well replace GJG^* by the term $\bar{G}J\bar{G}^*$, where $\bar{G} = G\Theta$ is the so-called proper generator matrix that results after applying Θ to G. Its first row has only one nonzero entry and we denote its columns by

$$\bar{G} = \begin{bmatrix} 0 & \cdots & 0 & \bar{\mathbf{w}}_j & 0 & \cdots & 0 \\ \bar{\mathbf{w}}_1 & & \bar{\mathbf{w}}_{j-1} & & \bar{\mathbf{w}}_{j+1} & & \bar{\mathbf{w}}_{p+q} \end{bmatrix} ,$$

where the top entry of $\bar{\mathbf{w}}_j$ is nozero. This allows us to write

$$GJG^* = \bar{G}J\bar{G}^* = \sum_{\substack{i=1 \\ i \neq j}}^{p+q} \begin{bmatrix} 0 \\ \bar{\mathbf{w}}_i \end{bmatrix} J_{ii} \begin{bmatrix} 0 \\ \bar{\mathbf{w}}_i \end{bmatrix}^* + \bar{\mathbf{w}}_j J_{jj} \bar{\mathbf{w}}_j^* ,$$

where J_{ii} indicates the i^{th} diagonal entry of J: it is equal to 1 for the first p positions and equal to -1 for the last q positions. Moreover,

$$l_0 = (I - f_0^* F)^{-1} \bar{\mathbf{w}}_j J_{jj} \delta_0^* , \quad d_0 = \frac{J_{jj} |\delta_0|^2}{1 - |f_0|^2} .$$

Substituting the above expressions for GJG^*, l_0 and d_0 into the left-hand side of (4.6) we obtain that it evaluates to the following

$$\sum_{\substack{i=1 \\ i \neq j}}^{p+q} \begin{bmatrix} 0 \\ \bar{\mathbf{w}}_i \end{bmatrix} J_{ii} \begin{bmatrix} 0 \\ \bar{\mathbf{w}}_i \end{bmatrix}^* + \Phi_0 \bar{\mathbf{w}}_j J_{jj} \bar{\mathbf{w}}_j^* \Phi_0^* ,$$

where $\Phi_0 = (F - f_0 I)(I - f_0^* F)^{-1}$. The above expression can be factored as

$$\begin{bmatrix} \mathbf{0} \\ G_1 \end{bmatrix} J \begin{bmatrix} \mathbf{0} \\ G_1 \end{bmatrix}^* ,$$

where

$$\begin{bmatrix} \mathbf{0} \\ G_1 \end{bmatrix} = \begin{bmatrix} 0 & \cdots & 0 & \Phi_0 \bar{\mathbf{w}}_j & 0 & \cdots & 0 \\ \bar{\mathbf{w}}_1 & & \bar{\mathbf{w}}_{j-1} & & \bar{\mathbf{w}}_{j+1} & & \bar{\mathbf{w}}_{p+q} \end{bmatrix} .$$

This is clearly equivalent to the matrix description (4.5c). □

We note again that the algorithm in Theorem 4.2 reduces to the classical Schur algorithm when $F = Z, p = 1 = q$. However, whether it is a fast $O(n^2)$ algorithm in the general case depends upon the nature of the matrix F. If F is such that matrix-vector products $F\mathbf{x}$ require only $O(n)$ or $O(n \log n)$ operations, then there will be a computational reduction. Fortunately, this appears to be true in many important applications. For example, in interpolation problems [29, 176, 182], F is usually in

Jordan form, for which we may recall the following well-known formula: when F is a single Jordan block, with f_0 on the diagonal, then

$$\Phi_0 = (F - f_0 I)(I - f_0^* F)^{-1} = \begin{bmatrix} B(f_0) & & & \\ B^{(1)}(f_0) & B(f_0) & & \\ \vdots & \ddots & \ddots & \\ B^{(n-1)}(f_0) & \dots & B^{(1)}(f_0) & B(f_0) \end{bmatrix},$$

where

$$B(z) = \frac{z - f_0}{1 - f_0^* z}$$

is the elementary Blaschke function associated with f_0, and $B^{(i)}(z) = d^i B(z)/dz^i$.

4.3.2. Terminology: Generalized Schur Algorithms. The above fast algorithms for triangular factorization of strongly regular matrices were obtained by combining the standard Gaussian elimination (also called Schur reduction) method for triangular factorization with the concept of displacement structure. For simplicity we presented a hierarchy of algorithms (for $F = Z$, F strictly lower triangular, F lower triangular), each of which reduces to (the array form of) Schur's original algorithm when applied to (quasi-)Toeplitz matrices. We shall use the generic term "Generalized Schur Algorithms" for all such algorithms (those above and several others to be presented later), because they are all based on the key fact that displacement structure allows us to speed up the computation of Schur complements (and successive Schur complementation is equivalent to triangular factorization).

This may be a good place to mention that the above results can be extended to strongly regular *non-Hermitian* matrices obeying displacement equations of the form

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

where F and A are $n \times n$ lower triangular matrices, and G and B are $n \times r$ (generator) matrices. The $r \times r$ signature matrix J can be incorporated into either G or B. But we often keep it explicit in order to make the analogy with the Hermitian case as close as possible. Every such matrix admits a triangular factorization of the form $R = LD^{-1}U$, where L is lower triangular and U is upper triangular with identical diagonal entries, equal to those of D. We shall not give the results at this time, because in Section 7 we shall consider the more general displacement equation studied by Kailath and Sayed [120, 175]

$$\Omega R \Delta^* - FRA^* = GJB^*,$$

where $\{\Omega, \Delta, F, A\}$ are lower triangular matrices.

Here instead, we shall take care of some unfinished business, viz., showing how to obtain proper generators. However, we should remark here that the issue of properness is important only because we wished to generalize the array form (3.5a) of the classical Schur algorithm. The other forms of the Schur algorithm can be generalized as well, and will give us algorithms that include the array algorithms as special cases – see Section 7.

4.4. Proper Generator Matrices. The first step in the algorithm of Theorem 4.2 is to reduce the given generator G_i to a so-called *proper* form, say $\bar{G}_i$, in which the top row of $\bar{G}_i$ contains only one nonzero element.

The existence of proper generators can be argued in several ways, one of which is based on the following simple, yet powerful, matrix result that plays an important role in the derivation of square-root algorithms (see, e.g., [173]). We include a simple proof for completeness.

LEMMA 4.3. *Consider two $n \times m$ $(n \leq m)$ matrices A and B. If $AJA^* = BJB^*$ is of full rank, for some $m \times m$ signature matrix $J = (I_p \oplus -I_q), p+q = m$, then there exists an $m \times m$ $J-$unitary matrix Θ $(\Theta J \Theta^* = J)$ such that $A = B\Theta$.*

Proof. One proof follows by invoking the hyperbolic singular value decompositions of A and B (see, *e.g.*, [151]), viz.,

$$A = U_A \begin{bmatrix} \Sigma_{A,+} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Sigma_{A,-} & \mathbf{0} \end{bmatrix} V_A^* \ , \ \ B = U_B \begin{bmatrix} \Sigma_{B,+} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Sigma_{B,-} & \mathbf{0} \end{bmatrix} V_B^* \ ,$$

where U_A and U_B are $n \times n$ unitary matrices, V_A and V_B are $m \times m$ $J-$unitary matrices, and $\Sigma_{A,+}, \Sigma_{A,-}$, $\Sigma_{B,+}$, and $\Sigma_{A,-}$, are $p' \times p'$, $q' \times q'$, $p' \times p'$, and $q' \times q'$ diagonal matrices, respectively, with $p' + q' = n$. It further follows from the full rank condition and the equality $AJA^* = BJB^*$, that $\Sigma_{A,+} = \Sigma_{B,+}$, $\Sigma_{A,-} = \Sigma_{B,-}$, and that we can choose $U_A = U_B$. Let $\Theta = JV_BJV_A^*$ then $\Theta J \Theta^* = J$ and $B\Theta = A$. □

Now assume we are given a generator G_i for the i^{th} Schur complement R_i,

$$R_i - F_i R_i F_i^* = G_i J G_i^* \ , \ \ J = (I_p \oplus -I_q) \ .$$

Comparing the $(0,0)$ entry on both sides leads to the equality,

$$d_i(1 - |f_i|^2) = g_i J g_i^* \ ,$$

where g_i is the top row of G_i and d_i is the $(0,0)$ entry of R_i. The strong regularity of R guarantees $d_i \neq 0$, which means that $g_i J g_i^*$ is either a positive or a negative real number. Let us first assume that $g_i J g_i^* > 0$, which is certainly the case with the algorithm of Section 4.3. It then follows that we can write

$$\begin{bmatrix} \delta_i & \mathbf{0} \end{bmatrix} J \begin{bmatrix} \delta_i^* \\ \mathbf{0} \end{bmatrix} = g_i J g_i^* \ ,$$

where δ_i is a square-root of $d_i(1 - |f_i|^2)$. This fits into the statement of Lemma 4.3 and we conclude that there exists a $J-$unitary rotation matrix Θ_i that reduces g_i to the form

$$g_i \Theta_i = \begin{bmatrix} \delta_i & \mathbf{0} \end{bmatrix} \ .$$

Consequently, the generator matrix $G_i \Theta_i$ will be proper. It is also clear that the nonzero entry δ_i could have been placed in any of the first p positions of the prearray.

We now consider the case $g_i J g_i^* < 0$, which may occur when R is strongly regular but not necessarily positive-definite as in Section 4.3.1. It then follows that we can write

$$\begin{bmatrix} \mathbf{0} & \delta_i \end{bmatrix} J \begin{bmatrix} \mathbf{0} \\ \delta_i^* \end{bmatrix} = g_i J g_i^* \ ,$$

where δ_i is a square-root of $-d_i(1 - |f_i|^2)$. This again fits into the statement of Lemma 4.3 and we conclude that there exists a $J-$unitary rotation matrix Θ_i that reduces g_i to the form

$$g_i\Theta_i = \begin{bmatrix} \mathbf{0} & \delta_i \end{bmatrix} ,$$

which again yields a proper generator $G_i\Theta_i$. It is also clear that the nonzero entry could have been placed in any of the last q positions of the prearray.

The rotation matrix Θ_i can be implemented in a variety of ways: by using a sequence of elementary Givens and hyperbolic rotations [94], Householder transformations [161, 27], as well as square-root and division-free versions of the elementary rotations (see, e.g., [84, 103]).

4.4.1. Elementary Circular and Hyperbolic Rotations. An elementary 2×2 unitary rotation Θ (also known as Givens or circular rotation) takes a row vector $\mathbf{x} = \begin{bmatrix} a & b \end{bmatrix}$ and rotates it to lie along the basis vector $\begin{bmatrix} 1 & 0 \end{bmatrix}$. More precisely, it performs the transformation

$$\begin{bmatrix} a & b \end{bmatrix} \Theta = \begin{bmatrix} \pm\sqrt{|a|^2 + |b|^2} & 0 \end{bmatrix} . \tag{4.7a}$$

The quantity $\pm\sqrt{|a|^2 + |b|^2}$ that appears on the right-hand side is consistent with the fact that the prearray, $\begin{bmatrix} a & b \end{bmatrix}$, and the postarray, $\begin{bmatrix} \pm\sqrt{|a|^2 + |b|^2} & 0 \end{bmatrix}$, must have equal Euclidean norms. An expression for Θ is given by

$$\Theta = \frac{1}{\sqrt{1 + |\gamma|^2}} \begin{bmatrix} 1 & \gamma \\ \gamma^* & -1 \end{bmatrix} \quad \text{where} \quad \gamma = \frac{b}{a} , \quad a \neq 0. \tag{4.7b}$$

In the trivial case $a = 0$ we simply choose Θ as the permutation matrix,

$$\Theta = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} .$$

We should also note that, in the special case of real data, a general unitary rotation as in (4.7b) can be expressed in the alternative form:

$$\Theta = \begin{bmatrix} c & s \\ s & -c \end{bmatrix} ,$$

where the so-called cosine and sine parameters, c and s, respectively, are defined by

$$c = \frac{1}{\sqrt{1 + |\gamma|^2}} , \quad s = \frac{\gamma}{\sqrt{1 + |\gamma|^2}} .$$

This justifies the name *circular rotation* for Θ, since the effect of Θ is to rotate the original vector $\mathbf{x}$ along the *circle* of equation $x^2 + y^2 = |a|^2 + |b|^2$, by an angle θ determined by the inverse of the above cosine and/or sine parameters, $\theta = \tan^{-1} \gamma$, in order to align it with the basis vector $\begin{bmatrix} 1 & 0 \end{bmatrix}$. The trivial case $a = 0$ corresponds to a 90 degrees rotation in an appropriate clockwise (if $b \geq 0$) or anti-clockwise (if $b < 0$) direction.

On the other hand, an elementary 2×2 hyperbolic rotation Θ takes a row vector $\mathbf{x} = \begin{bmatrix} a & b \end{bmatrix}$ and rotates it to lie either along the basis vector $\begin{bmatrix} 1 & 0 \end{bmatrix}$ (if $|a| > |b|$) or along the basis vector $\begin{bmatrix} 0 & 1 \end{bmatrix}$ (if $|a| < |b|$). More precisely, it performs either of the transformations

$$\text{(4.8a)} \qquad \begin{bmatrix} a & b \end{bmatrix} \Theta = \begin{bmatrix} \pm\sqrt{|a|^2 - |b|^2} & 0 \end{bmatrix} \text{ if } |a| > |b| \ ,$$

$$\text{(4.8b)} \qquad \begin{bmatrix} a & b \end{bmatrix} \Theta = \begin{bmatrix} 0 & \pm\sqrt{|b|^2 - |a|^2} \end{bmatrix} \text{ if } |a| < |b|.$$

The quantity $\sqrt{\pm(|a|^2 - |b|^2)}$ that appears on the right-hand side of the above expressions is consistent with the fact that the prearray, $\begin{bmatrix} a & b \end{bmatrix}$, and the postarrays must have equal $J-$norms. By the $J-$norm of a row vector $\mathbf{x}$ we mean the indefinite quantity $\mathbf{x}J\mathbf{x}^*$, which can be positive, negative, or even zero. Here,

$$J = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = (1 \oplus -1) \ .$$

An expression for a $J-$unitary hyperbolic rotation Θ that achieves (4.8a) or (4.8b) is given by

$$\text{(4.8c)} \quad \Theta = \frac{1}{\sqrt{1-|\gamma|^2}} \begin{bmatrix} 1 & -\gamma \\ -\gamma^* & 1 \end{bmatrix} \quad \text{where} \quad \gamma = \frac{b}{a} \ , \ a \neq 0 \ , \ |a| > |b| \ ,$$

$$\text{(4.8d)} \quad \Theta = \frac{1}{\sqrt{1-|\gamma|^2}} \begin{bmatrix} 1 & -\gamma \\ -\gamma^* & 1 \end{bmatrix} \quad \text{where} \quad \gamma^* = \frac{a}{b} \ , \ b \neq 0 \ , \ |a| < |b| \ ,$$

We should also note that, in the case of real data, a general hyperbolic rotation as in (4.8c) or (4.8d) can be expressed in the alternative form:

$$\Theta = \begin{bmatrix} ch & -sh \\ -sh & ch \end{bmatrix} \ ,$$

where the so-called hyperbolic cosine and sine parameters, ch and sh, respectively, are defined by

$$ch = \frac{1}{\sqrt{1-|\gamma|^2}} \ , \quad sh = \frac{\gamma}{\sqrt{1-|\gamma|^2}} \ .$$

This justifies the name *hyperbolic rotation* for Θ, since the effect of Θ is to rotate the original vector $\mathbf{x}$ along the *hyperbola* of equation $x^2 - y^2 = |a|^2 - |b|^2$, by an angle θ determined by the inverse of the above hyperbolic cosine and/or sine parameters, $\theta = \tanh^{-1} \gamma$, in order to align it with the appropriate basis vector. Note also that the special case $|a| = |b|$ corresponds to a row vector $\mathbf{x} = \begin{bmatrix} a & b \end{bmatrix}$ with zero hyperbolic norm since $|a|^2 - |b|^2 = 0$. It is then easy to see that there does not exist a hyperbolic rotation that will rotate $\mathbf{x}$ to lie along either bases vectors.

4.4.2. An Example. Consider, for example, a 4−column (real) generator matrix G along with a 4×4 signature matrix J,

$$G = \begin{bmatrix} a & b & c & d \\ x & x & x & x \\ x & x & x & x \\ \vdots & \vdots & \vdots & \vdots \\ x & x & x & x \end{bmatrix} \ , \quad J = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & -1 & \\ & & & -1 \end{bmatrix} \ ,$$

and assume that we are interested in applying a $J-$ unitary transformation Θ to G in order to align its first row along the basis vector $\mathbf{e}_0$, viz., we want

$$G\Theta = \begin{bmatrix} \delta & 0 & 0 & 0 \\ x' & x' & x' & x' \\ x' & x' & x' & x' \\ \vdots & \vdots & \vdots & \vdots \\ x' & x' & x' & x' \end{bmatrix}.$$

Then one way to achieve this, among *many* possible options, would be the following: we first annihilate the b entry by using a circular rotation that leaves unchanged the last two columns of the prearray,

$$\begin{bmatrix} \boxed{a} & \boxed{b} & c & d \\ x & x & x & x \\ x & x & x & x \\ \vdots & \vdots & \vdots & \vdots \\ x & x & x & x \end{bmatrix} \begin{bmatrix} c_0 & s_0 & & \\ s_0 & -c_0 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} = \begin{bmatrix} a_1 & 0 & c & d \\ \# & \# & x & x \\ \# & \# & x & x \\ \vdots & \vdots & \vdots & \vdots \\ \# & \# & x & x \end{bmatrix}.$$

We then annihilate the d entry by using a second circular rotation that leaves unchanged the first two columns of the prearray,

$$\begin{bmatrix} a_1 & 0 & \boxed{c} & \boxed{d} \\ \# & \# & x & x \\ \# & \# & x & x \\ \vdots & \vdots & \vdots & \vdots \\ \# & \# & x & x \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & c_1 & s_1 \\ & & s_1 & -c_1 \end{bmatrix} = \begin{bmatrix} a_1 & 0 & c_1 & 0 \\ \# & \# & * & * \\ \# & \# & * & * \\ \vdots & \vdots & \vdots & \vdots \\ \# & \# & * & * \end{bmatrix}.$$

We finally annihilate the c_1 entry by using a hyperbolic rotation, which leaves unchanged the second and fourth columns of the prearray (assuming $|c_1| < |a_1|$),

$$\begin{bmatrix} \boxed{a_1} & 0 & \boxed{c_1} & 0 \\ \# & \# & * & * \\ \# & \# & * & * \\ \vdots & \vdots & \vdots & \vdots \\ \# & \# & * & * \end{bmatrix} \begin{bmatrix} ch_1 & & -sh_1 & \\ & 1 & & \\ -sh_1 & & ch_1 & \\ & & & 1 \end{bmatrix} = \begin{bmatrix} \delta & 0 & 0 & 0 \\ y & \# & y & * \\ y & \# & y & * \\ \vdots & \vdots & \vdots & \vdots \\ y & \# & y & * \end{bmatrix},$$

The parameters that define the previous rotations are clearly given by

$$c_0 = \frac{a}{\sqrt{|a|^2 + |b|^2}}, \quad s_0 = \frac{b}{\sqrt{|a|^2 + |b|^2}},$$

$$c_1 = \frac{c}{\sqrt{|c|^2 + |d|^2}}, \quad s_1 = \frac{d}{\sqrt{|c|^2 + |d|^2}},$$

$$ch_1 = \frac{a_1}{\sqrt{|a_1|^2 - |c_1|^2}}, \quad sh_1 = \frac{c_1}{\sqrt{|a_1|^2 - |c_1|^2}}.$$

4.4.3. Rotations for the Non-Hermitian Case. In future sections, especially when we deal with non-Hermitian structured matrices, a pair of rotation matrices will sometimes be needed in order to convert generators to proper form. We first focus on elementary 2×2 rotation matrices that are required to satisfy the relation

$$\Theta\Gamma^* = I \,. \tag{4.9a}$$

More specifically, consider two row vectors $\mathbf{x} = \begin{bmatrix} a & b \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} c & d \end{bmatrix}$, and suppose that we are interested in determining two matrices Θ and Γ that satisfy (4.9a) and such that they perform the transformations

$$\begin{bmatrix} a & b \end{bmatrix}\Theta = \begin{bmatrix} \alpha & 0 \end{bmatrix} \quad , \quad \begin{bmatrix} c & d \end{bmatrix}\Gamma = \begin{bmatrix} \beta & 0 \end{bmatrix}. \tag{4.9b}$$

It follows from (4.9a) that the scalars α and β should satisfy the equality $\alpha\beta^* = ac^* + bd^*$. We further assume that $ac^* + bd^* \neq 0$, which will be guaranteed by the strong regularity assumption throughout the paper . Expressions for Θ and Γ that achieve (4.9b) can be chosen as follows:

(i) If $a \neq 0$ and $c \neq 0$ then define $\gamma = b/a$, $\lambda = d/c$ and write

$$\Theta = \frac{1}{1+\gamma\lambda^*}\begin{bmatrix} 1 & \gamma \\ \lambda^* & -1 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} 1 & \lambda \\ \gamma^* & -1 \end{bmatrix}.$$

(ii) If $a = 0$ and $c \neq 0$ then d is necessarily nonzero because of the condition $ac^* + bd^* \neq 0$. We can, therefore, choose Θ and Γ as follows

$$\Theta = \begin{bmatrix} \alpha^* & 1 \\ 1 & 0 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} 0 & 1 \\ 1 & -\alpha \end{bmatrix}, \quad \text{where} \quad \alpha = \frac{c}{d}.$$

(iii) If $a \neq 0$ and $c = 0$ then b is necessarily nonzero, and we choose

$$\Theta = \begin{bmatrix} 0 & 1 \\ 1 & -\alpha \end{bmatrix}, \quad \Gamma = \begin{bmatrix} \alpha^* & 1 \\ 1 & 0 \end{bmatrix}, \quad \text{where} \quad \alpha = \frac{a}{b}.$$

If we instead want to determine rotation matrices Θ and Γ that satisfy $\Theta J\Gamma^* = J$, where J is a signature matrix, and $J^2 = I$, then this is equivalent to requiring

$$\Theta \underbrace{(J\Gamma^* J)}_{\bar{\Gamma}^*} = I \,,$$

which shows that we can reduce the problem to that of determining two rotations Θ and $\bar{\Gamma}$ as in (4.9a).

4.4.4. Householder Reflections. We have described the use of elementary rotations in some detail. As is well-known, elementary (Householder) reflections can also be used (or a mixed sequence of rotations and reflections). Reference [61] discussed the use of Householder reflections in the Hermitian case. But since it is less familiar, here we shall briefly describe non-Hermitian Householder matrices [48]. Let $\mathbf{u}$ and $\mathbf{v}$ be two row vectors. A matrix of the form

$$\Theta = I - 2\mathbf{u}^*\mathbf{v}, \quad \text{with} \quad \mathbf{u}\mathbf{v}^* = 1,$$

will be called an elementary Householder matrix; it reduces to the usual (unitary) Householder matrix when $\mathbf{u} = \mathbf{v}$. Note also that $\Theta = \Theta^{-1}$. Now consider two row vectors $\mathbf{x}$ and $\mathbf{y}$ and assume we want to determine a non-Hermitian Householder matrix Θ so as to reduce them to two other vectors $\mathbf{x}'$ and $\mathbf{y}'$, say

$$\mathbf{x}' = \mathbf{x}\Theta, \quad \mathbf{y}' = \mathbf{y}\Theta^*.$$

For this to be possible, it is clear that we must have $\mathbf{x}'\mathbf{y}'^* = \mathbf{x}\mathbf{y}^*$ and $\mathbf{x}'\mathbf{y}^* = \mathbf{x}\mathbf{y}'^*$. We can determine Θ (i.e., $\mathbf{u}$ and $\mathbf{v}$) as follows. Note that

$$\mathbf{x}' = \mathbf{x} - 2\beta_1\mathbf{v} \quad \text{and} \quad \mathbf{y}' = \mathbf{y} - 2\beta_2\mathbf{u},$$

with $\beta_1 = \mathbf{x}\mathbf{u}^*$ and $\beta_2 = \mathbf{y}\mathbf{v}^*$, so that

$$\mathbf{v} = (\mathbf{x} - \mathbf{x}')/2\beta_1, \quad \mathbf{u} = (\mathbf{y} - \mathbf{y}')/2\beta_2.$$

Therefore,

$$\beta_1 = \mathbf{x}\mathbf{u}^* = \mathbf{x}(\mathbf{y} - \mathbf{y}')^*/2\beta_2, \ \ \beta_2 = \mathbf{y}\mathbf{v}^* = \mathbf{y}(\mathbf{x} - \mathbf{x}')^*/2\beta_1.$$

Now to satisfy $\mathbf{u}\mathbf{v}^* = 1$, we must have $4\beta_1^*\beta_2 = (\mathbf{y} - \mathbf{y}')(\mathbf{x} - \mathbf{x}')^*$, which collapses to the requirement

$$2\beta_1^*\beta_2 = \mathbf{y}(\mathbf{x} - \mathbf{x}')^*.$$

Therefore, any choice of $\{\beta_1, \beta_2\}$ that satisfies this condition will work.

5. Some Applications in Matrix Computation. The generator recursions that we presented so far will be further generalized in Section 7. But we are already in a position to illustrate the applicability of the previous results to several matrix computations. We shall start by showing the value of extending the definition of displacement structure by replacing Z in (2.1b) by more general (strictly lower) triangular matrices F. We shall show that such extensions allow us to efficiently study composite Toeplitz matrices such as $T_1T_2, T_1^{-1}, T_1 - T_2T_3^{-1}T_4$, $T_1T_2 - T_3T_4$, where the $\{T_i\}$ are all Toeplitz matrices. We could always use the displacement representation for T and T^{-1} (*cf.* (2.3a) and (2.3c)) to obtain the displacement representations for the above composite matrices, and then apply the generalized Schur algorithm. A better technique is to find first an appropriate "Toeplitz-block" matrix in which the above composite matrices appear as certain Schur complement matrices, and then to study these Toeplitz-block matrices by introducing an appropriate definition of displacement (see, e.g., [50, 52, 113]). The key fact then used is that displacement structure is preserved under Schur complementation. The ideas are not restricted to Toeplitz-block matrices alone – see [50].

Example 5.1: Study of $\mathbf{T}^{-1}$.

First note that T^{-1} is the Schur complement of the $(1,1)$ block in the Toeplitz-block matrix

$$M = \begin{bmatrix} -T & I \\ I & \mathbf{0} \end{bmatrix}. \tag{5.1a}$$

Now by examining $M - Z_{2n}MZ_{2n}$, we can see that the displacement rank of M is less than or equal to 4, where we have employed the notation Z_{2n} to denote the $2n \times 2n$ lower shift matrix. Therefore, by Theorem 2.2, the rank of the Schur complement,

T^{-1}, must also be less than or equal to 4. However, this is a weak conclusion, because we know that the displacement rank of T^{-1} is 2 (*cf.* (2.3c)).

If we instead employ the definition

$$\nabla M = M - \begin{bmatrix} Z_n & \mathbf{0} \\ \mathbf{0} & Z_n \end{bmatrix} M \begin{bmatrix} Z_n & \mathbf{0} \\ \mathbf{0} & Z_n \end{bmatrix}^* , \tag{5.1b}$$

where we use $F = (Z_n \oplus Z_n)$ in the definition $R - FRF^*$ rather than $F = Z_{2n}$, then it is easy to see that the displacement rank of M is now 2. In fact, the reader might wish to check that for a symmetric Toeplitz matrix $T = [c_{|i-j|}]_{i,j=0}^{n-1}$, $c_0 = 1$, we obtain

$$M - FMF^* = GJG^*, \quad F = Z_n \oplus Z_n,$$

where $J = (1 \oplus -1)$ and

$$G^{\mathbf{T}} = \begin{bmatrix} 0 & c_1 & \dots & c_{n-1} & 1 & 0 & \dots & 0 \\ 1 & c_1 & \dots & c_{n-1} & 1 & 0 & \dots & 0 \end{bmatrix}^{\mathbf{T}} . \tag{5.2}$$

Example 5.2: Simultaneous Factorization of T and $\mathbf{T}^{-1}$.

As we noted before, the classical Schur algorithm gives a fast $O(n^2)$ algorithm for the triangular factorization of a Toeplitz matrix T. Now the triangular factors of T^{-1} cannot be formed just by inverting those of T – direct inversion of any, even triangular, matrix in general requires $O(n^3)$ operations. The celebrated Levinson algorithm [137] in fact gives an $O(n^2)$ algorithm for factoring T^{-1}.

However, we shall see that by applying the generalized Schur algorithm to the matrix M in Example 5.1, we can simultaneously factor both T and T^{-1}. Thus consider the situation after we apply n steps of the generalized Schur algorithm to the generator G of M, viz., (5.2). Of course, we shall then get a generator, say $\{\mathbf{a}, \mathbf{b}\}$, of the Schur complement T^{-1} from which we can recover the matrix T^{-1} as

$$T^{-1} = \mathbf{L}(\mathbf{a})\mathbf{L}^*(\mathbf{a}) - \mathbf{L}(\mathbf{b})\mathbf{L}^*(\mathbf{b}).$$

But what we want is not T^{-1} but its triangular factors. One way of getting these is applying the classical Schur recursion to the generator $\{\mathbf{a}, \mathbf{b}\}$. But in fact the factors of T^{-1} are already available from the results of the first n steps of the Schur recursion (4.5c) applied to the generator of M.

To clarify this, assume we apply the first n recursive steps of the Schur algorithm (4.5c) to a generator of the $2n \times 2n$ matrix M, with $F = (Z_n \oplus Z_n)$. This provides us with the first n columns and the first n diagonal entries of the triangular factors of M, which we denote by L_{2n} and D_{2n}. That is, we obtain the first n columns of L_{2n} and the first n entries of D_{2n} in the factorization $M = L_{2n}D_{2n}^{-1}L_{2n}^*$. Let us denote the leading $n \times n$ block of D_{2n} by D and let us partition the first n columns of L_{2n} into the form

$$\begin{bmatrix} L \\ U \end{bmatrix} ,$$

where L is $n \times n$ lower triangular, and U is an $n \times n$ matrix that we shall soon see has to be upper triangular. It follows from the Schur reduction representation (4.1c) that we must have

$$\begin{bmatrix} -T & I \\ I & \mathbf{0} \end{bmatrix} = M = \begin{bmatrix} L \\ U \end{bmatrix} D^{-1} \begin{bmatrix} L^* & U^* \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & T^{-1} \end{bmatrix} .$$

By equating terms on both sides of the above equality we conclude that $U = L^{-*}D$, $-T^{-1} = UD^{-1}U^*$, and $-T = LD^{-1}L^*$. Hence, the first n recursive steps of the algorithm provide not only the triangular factorization of T but also the triangular factorization of T^{-1}. This is an alternative to the use of the Levinson algorithm for this problem. It has advantages in terms of parallel implementation because it has no inner products in it, unlike the Levinson algorithm; however the Levinson algorithm has somewhat fewer computations (though still $O(n^2)$). It may be noted that the Levinson algorithm can also be derived from the generalized Schur recursion applied to M in (5.1a) – see [48, p. 46].

Example 5.3: QR Factorization of Structured Matrices

Yet another popular algorithm in signal processing is the so-called lattice filtering algorithm, which turns out to be equivalent to the so-called orthogonal triangularization (QR factorization, Q orthogonal) of a particular rectangular Toeplitz matrix (see, e.g., [59, 141]). The first algorithm for fast QR factorization of Toeplitz matrices was given by [191]; later other algorithms were given by Bojanczyk, Brent and de Hoog [17], and Cybenko [60]. Here we shall show, following [52], how displacement ideas lead to an easily-described algorithm.

Let X be an $n \times n$ matrix. Form the displacement of

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

with $F = Z_n \oplus Z_n \oplus Z_n$, and find a generator for M. A general procedure for doing this has been given in [48]; in many cases, e.g., when X is Toeplitz, one can obtain a generator of length 5 almost by inspection.

After n steps of the generalized Schur algorithm applied to a generator of M, we shall have a generator of

$$M_1 = \begin{bmatrix} X^*X & X^* \\ X & \mathbf{0} \end{bmatrix} .$$

After another n steps, we shall as in Example 5.2 have the partial triangularization (where L is $n \times n$ lower triangular, and U is an $n \times n$ matrix)

$$M_1 = \begin{bmatrix} L \\ U \end{bmatrix} D^{-1} \begin{bmatrix} L^* & U^* \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -I \end{bmatrix} .$$

By equating terms on both sides of the above equality we conclude that

$$X^*X = (LD^{-*/2})(LD^{-*/2})^* \ , \ (UD^{-*/2})(LD^{-*/2})^* = X ,$$

and $(UD^{-*/2})(UD^{-*/2})^* = I$. Therefore, we can identify

$$Q = UD^{-*/2} \quad , \quad R = (LD^{-*/2})^* .$$

Here, D is a positive-definite diagonal matrix and $D^{1/2}$ denotes a diagonal matrix whose entries are the square-roots of the diagonal entries of D. In summary, the QR factors of the structured matrix X can be obtained by applying the Schur recursion to a properly defined extended structured matrix M. The interested reader can check that one could compute R^{-1} directly by performing the $2n$ steps of partial triangularization of (compare with [60])

$$M' = \begin{bmatrix} -I & X & \mathbf{0} \\ X^* & \mathbf{0} & I \\ \mathbf{0} & I & \mathbf{0} \end{bmatrix}.$$

Example 5.4: Avoiding Back-Substitution

The previous examples all involved the Schur algorithm for Hermitian matrices. Here is an example of a problem where this does not hold. The generalized Schur algorithm for non Hermitian matrices is given later (Section 7.4), but it is not necessary to know the exact algorithm in order to follow the present discussion.

Consider a linear system of equations of the form

$$Tx = b,$$

where T is an $n \times n$ strongly regular Hermitian Toeplitz matrix, and b is a known column vector. One possibility for determining the entries of x is the following: compute the triangular factorization of T, say

$$T = LD^{-1}L^*,$$

and then solve, via back-substitution, the triangular system of equations in y and x,

$$LD^{-1}y = b \quad \text{and} \quad L^*x = y.$$

A major drawback of a back-substitution step is that it involves serial operations and does not lend itself to a parallelizable algorithm.

A way out of this is to employ a bordering (or embedding) technique (see, e.g., [113]). For this purpose, we define the extended matrix

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

and note that the Schur complement of $-T$ in R is precisely $T^{-1}b$, which is equal to the desired solution x. Now the matrix R itself is also structured since T is Toeplitz. More precisely, we know that $T - ZTZ^*$ has rank 2 and it follows that

$$\begin{bmatrix} -T & b \\ I & \mathbf{0} \end{bmatrix} - \begin{bmatrix} Z & \\ & Z \end{bmatrix} \begin{bmatrix} -T & b \\ I & \mathbf{0} \end{bmatrix} \begin{bmatrix} Z & \\ & 0 \end{bmatrix}^* \quad \text{also has low rank.}$$

Therefore, after n steps of partial triangularization of R, we shall have a generator of its Schur complement, from which we can read out the solution x. Similar ideas can be used for avoiding backsubstitution in finding least-squares solutions of overdetermined linear equations – see [48, 113]. Applications in adaptive filtering and instrumental-variable methods can be found in [172, 184].

Example 5.5: More on Inverses: Admissible Generators

For Toeplitz matrices, T, the two problems of triangular factorization of T and T^{-1} are of the same order of complexity – both have displacement rank 2 and corresponding generators are not hard to find. The situation is more complicated for arbitrary matrices.

Thus let R be an $n \times n$ Hermitian positive-definite matrix with a known generator,

$$R - Z_n R Z_n^* = GJG^*, \quad J = (I_p \oplus -I_q).$$

The matrix

$$N = \begin{bmatrix} R & I \\ I & 0 \end{bmatrix}$$

can be used to determine R^{-1} via the generalized Schur algorithm applied to a generator of N with respect to $F = Z_n \oplus Z_n$. When $R = T$, a generator is easy to find. In general however the procedure is not as immediate. One way out is to use a generator of length $p+q+2$, not $p+q$. One such generator is easily checked to be

$$\begin{bmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_p & \mathbf{e}_1 & \mathbf{y}_1 & \dots & \mathbf{y}_q & \mathbf{e}_1 \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{e}_1/2 & \mathbf{0} & \dots & \mathbf{0} & -\mathbf{e}_1/2 \end{bmatrix}, \quad J = (I_{p+1} \oplus -I_{q+1}),$$

where $G = \begin{bmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_p & \mathbf{y}_1 & \dots & \mathbf{y}_q \end{bmatrix}$ and $\mathbf{e}_1$ is the first unit vector.

However if the given generator G satisfies a condition called *admissibility*, introduced in [133], then we can obtain a generator of smaller length. A generator $\{G, J\}$ for a Hermitian matrix R is said to be admissible if $\mathbf{e}_1$ belongs to the range of G, i.e., there is a linear combination of the columns of G that will give the unit vector, say

$$\sum_{i=1}^{p} \mu_i \mathbf{x}_i + \sum_{i=1}^{q} \nu_i \mathbf{y}_i = \mathbf{e}_1.$$

Then it can be checked that the matrix

$$N_1 = \begin{bmatrix} R & I \\ I & \eta I \end{bmatrix}, \quad \eta = \sum_{i=1}^{p} |\mu_i|^2 - \sum_{i=1}^{q} |\nu_i|^2,$$

obeys

$$N_1 - F N_1 F^* = BJB^*,$$

where $F = (Z_n \oplus Z_n)$, $J = (I_p \oplus -I_q)$ and

$$B = \begin{bmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_p & \mathbf{y}_1 & \dots & \mathbf{y}_q \\ \mu_1 \mathbf{e}_1 & \dots & \mu_p \mathbf{e}_1 & -\nu_1 \mathbf{e}_1 & \dots & -\nu_q \mathbf{e}_1 \end{bmatrix}.$$

Since the Schur complement of R in N_1 is $\eta I - R^{-1}$, we see that the generator of R^{-1} obtained in this way will have length $(p+q)$ if $\eta = 0$ or $(p+q+1)$ if $\eta \neq 0$. For a more detailed study of admissibility, structured inverses and interesting relations to generalized orthogonal polynomials and generalized Christoffel-Darboux formulas, we refer to [130, 133].

Here we note only that T^*T, where $T = [c_{i-j}]$ is an $m \times n$ Toeplitz matrix of full column rank, has an admissible minimal generator of length 4, with columns say

$$\mathbf{x}_1 = T^* t_1 / \|t_1\|_2, \;\; \mathbf{x}_2 = t_2, \;\; \mathbf{y}_1 = Z_n Z_n^* \mathbf{x}_1, \;\; \mathbf{y}_2 = Z_n s_1,$$

and

$$t_1 = \mathrm{col}\{c_0, c_1, \ldots, c_{m-1}\}, \;\; t_2 = \mathrm{col}\{0, c_{-1}, \ldots, c_{1-n}\},$$

$$s_1 = \mathrm{col}\{c_{m-1}, \ldots, c_{m-n}\},$$

where $\|\cdot\|_2$ denotes the Euclidean norm. This explicit description is useful in many problems, e.g., in Example 5.3.

Several other embedding examples can be found in [48, 113]. Further examples that arise in adaptive filtering, instrumental-variable methods, system identification, and in the design of decision feedback equalizers can be found in [4, 5, 46, 47, 172, 184]. Also, in Section 6.1 we shall consider applications with more general matrices F (*i.e.,* not necessarily strictly lower triangular), which arise in the study of interpolation problems [182].

6. Generalized Cascade/Transmission Line Interpretations and Interpolation Problems. We examined earlier in Section 3.2 a cascade network interpretation of the classical Schur algorithm, and pointed out connections with inverse scattering problems. This interpretation is equally applicable to the generalized Schur algorithm of Theorem 4.2, and to other extensions that are given in future sections. As in Section 3.2, the cascade/transmission line interpretation can be used to obtain nice solutions to various problems; here we shall use them to study some general interpolation problems.

We shall begin with the special case of a positive-definite matrix R as in (4.4a), which will be used later in Section 6.1 in the study of analytic interpolation problems. The corresponding array recursion (4.5c) can be written as

$$\begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = \Phi_i G_i \Theta_i \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + G_i \Theta_i \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix}, \tag{6.1a}$$

where Θ_i reduces the top row of G_i to the form

$$g_i \Theta_i = \begin{bmatrix} \delta_i & 0 & \ldots & 0 \end{bmatrix}. \tag{6.1b}$$

Each step of (6.1a) can be depicted graphically as a cascade network of elementary sections, one of which is shown in Figure 6.1; Θ_i is any $J-$unitary matrix that rotates the first row of the i^{th} generator to $\begin{bmatrix} \delta_i & \mathbf{0} \end{bmatrix}$. The rows of G_i enter the section one row at a time. The left-most entry of each row is applied through the top line, while the remaining entries are applied through the bottom lines. The Blasckhe matrix Φ_i then acts on the entries of the top line. When $F_i = Z$, the lower shift matrix, $\Phi_i = Z$, a delay unit. In general, note that the first row of each Φ_i is zero, and in this sense Φ_i acts as a generalized delay element. To clarify this, observe that when the entries of the first row of G_i are processed by Θ_i and Φ_i, the values of the outputs of the section will all be zero. The rows of G_{i+1} will start appearing at these outputs only when the second and higher rows of G_i are processed by the section.

As in the case of the classical Schur algorithm, we can get a transmission line equivalent of the cascade, i.e., a form in which the data flow in the lower q lines

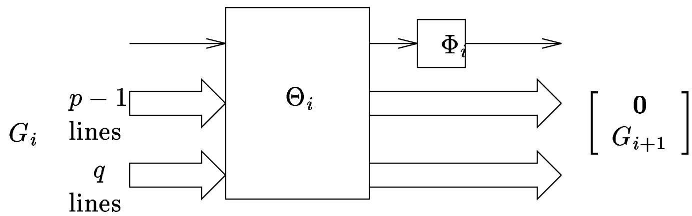

FIG. 6.1. *The feedforward structure of the generalized Schur recursion: The positive-definite case*

.

can be reversed in direction; the $J-$unitary matrix Θ_i is now replaced by a unitary matrix Σ_i. We forego specifying the easily derived form of Σ_i here; it will be given in Section 6.1.1, when we discuss the application of the cascade to solving interpolation problems. We also remark that for matrices R that are not necessarily positive-definite, the feedforward structure of Figure 6.1 can be easily modified to reflect the array form of the corresponding generator recursion. This simply corresponds to modifying the position of the single line in the figure whose entries undergo a transformation by Φ_i.

Moreover, a generalized cascade representation in the function domain can also be derived as follows. In Theorem 4.2, the recursion starts with G_i and gives as outputs G_{i+1} via (4.5c) and also the (nonzero part of the) i^{th} column l_i of the triangular factorization of R via (4.5d). The latter can be rewritten as

$$\text{(6.2)} \qquad \frac{1}{\delta_i^* J_{jj}}(I_{n-i} - f_i^* F_i)l_i = G_i\Theta_i \begin{bmatrix} \mathbf{0} \\ 1 \\ \mathbf{0} \end{bmatrix}.$$

We note that the update of the generator recursion at the i^{th} step ($G_i \rightarrow G_{i+1}$), as well as the determination of the i^{th} triangular factor of R, are determined by transformations on the j^{th} column of the proper generator $G_i\Theta_i$. The remaining columns of $G_i\Theta_i$ remain unchanged and thus do not explicitly contribute to the i^{th} step. Expressions (4.5c) and (6.2) can now be combined into the following interesting state-space form

$$\text{(6.3a)} \begin{bmatrix} l_i & \begin{matrix} \mathbf{0} \\ G_{i+1} \end{matrix} \end{bmatrix} = \begin{bmatrix} F_i l_i & G_i \end{bmatrix} \begin{bmatrix} f_i^* & \frac{\delta_i}{d_i}\begin{bmatrix} \mathbf{0} & 1 & \mathbf{0} \end{bmatrix} \\ \delta_i^* J_{jj}\Theta_i \begin{bmatrix} \mathbf{0} \\ 1 \\ \mathbf{0} \end{bmatrix} & \Theta_i \begin{bmatrix} I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -f_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I \end{bmatrix} \end{bmatrix}.$$

More specifically, we can regard (6.3a) as defining a first-order system in state-space form (see, e.g., [108]) with inputs from the left: the entries of $F_i l_i$ can be regarded as current states and the entries of l_i can be regarded as the next states. Also, the rows of G_i can be regarded as current inputs and those of G_{i+1} as outputs.

In linear system theory, a very useful alternative to the above state-space description is the so-called *transfer function* from input to output (see, e.g., [108]). We shall denote this as $\Theta_i(z)$ and straightforward calculation using z^{-1}-transforms yields the formula

$$\Theta_i(z) = \Theta_i \begin{bmatrix} I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -f_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I \end{bmatrix} + \delta_i^* J_{jj} \Theta_i \begin{bmatrix} \mathbf{0} \\ 1 \\ \mathbf{0} \end{bmatrix} [z^{-1} - f_i^*]^{-1} \frac{\delta_i}{d_i} \begin{bmatrix} \mathbf{0} & 1 & \mathbf{0} \end{bmatrix}$$

$$(6.3b) \qquad = \Theta_i \begin{bmatrix} I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & B_i(z) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I \end{bmatrix}, \quad B_i(z) = \frac{z - f_i}{1 - f_i^* z}.$$

Alternatively, from a purely mathematical point of view we can merely associate the $r \times r$ function $\Theta_i(z)$ with (6.3a), noting for instance that it is unchanged by a similarity transformation of the $r \times r$ so-called system matrix in (6.3a). In the early mathematical work of Livsic, Brodskii, Potapov, and others (see, e.g., [125, 139]), $\Theta_i(z)$ was called the *characteristic function.*

It further follows from the expression for $\Theta_i(z)$ that it satisfies the normalization condition $\Theta_i(z) J \Theta_i^*(z) = J$ on $|z| = 1$ since $B_i(z) B_i^*(z) = 1$ on $|z| = 1$. If we additionally assume that $|f_i| < 1$ then it follows that $\Theta_i(z)$ is analytic in $|z| < 1$ and satisfies $\Theta_i(z) J \Theta_i^*(z) < J$ in $|z| < 1$. That is, the resulting $\Theta_i(z)$ will be what is called a $J-$*lossless matrix function* (see, e.g., [71]). In summary, we have the following.

LEMMA 6.1. *Consider the same setting as Theorem 4.2. Each step of the recursive algorithm of the theorem then gives rise to an $r-$input $r-$output first-order (i.e., one-dimensional) transfer matrix,*

$$\Theta_i(z) = \Theta_i \begin{bmatrix} I_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & B_i(z) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{r-j-1} \end{bmatrix},$$

which satisfies

$$\Theta_i(z) J \Theta_i^*(z) = J \quad on \quad |z| = 1 \, .$$

If it further happens that $|f_i| < 1$ then it follows that $\Theta_i(z)$ is analytic in $|z| < 1$ and satisfies $\Theta_i(z) J \Theta_i^*(z) < J$ in $|z| < 1$. That is, $\Theta_i(z)$ is a $J-$lossless matrix function.

Thus our factorization algorithms are closely linked with a cascade of elementary sections, generalizing the cascade associated with the classical Schur algorithm. The cascade/transmission line picture was used in Section 3.2 to show that the classical Schur algorithm (in array form) provided a natural direct solution of the inverse scattering problem for discrete transmission lines. This inverse scattering problem actually solves the so-called Carathéodory interpolation problem, which can be stated (in one of its forms) as follows: Given a sequence $\{1, 0, \ldots, 0\}$ find all $n-$th order $J-$lossless systems whose response to the input sequence $\{1, 0, 0, \ldots\}$ is $\{s_0, s_1, \ldots, s_{n-1}, \ldots\}$, where the $\{s_i\}$ are the coefficients of the power series expansion of a Schur-type function. The solution is that the classical Schur algorithm applied to the generator matrix

$$G^{\mathbf{T}} = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ s_0 & s_1 & \ldots & s_n \end{bmatrix}^{\mathbf{T}}$$

will determine a cascade of n $J-$lossless sections that will have the desired behaviour; all such solutions can then be obtained (in a standard way) by "loading" (or terminating) the transmission-line form of this cascade with any Schur function.

It turns out that the generalized cascade described just above allows us to give a similar solution to much more general interpolation problems.

6.1. Applications to Interpolation Problems. Interpolation problems of various types have a long history in mathematics and in circuit theory, control theory, and system theory. Not surprisingly, this rich subject can be approached in many ways and in different settings. We cannot give a full account of the many prior contributions to this literature. The works of Krein and Nudelman [126], Nudelman [149, 150], Foias and Frazho [76], Ball, Gohberg, and Rodman [20], Rosenblum and Rovnyak [169], Landau [127, 128], Helton [100], Kovalishna and Potapov [125], and Dym [71] may be consulted for this.

6.1.1. Analytic Interpolation Problems. In this section, we briefly describe a recursive solution to rational analytic interpolation problems that has been recently proposed by Sayed and Kailath [171, 176, 182]; reference [182] further elaborates on connections with earlier work on the subject.

The basis for our approach is the generalized Schur algorithm of Theorem 4.2, along with the result of Lemma 6.1. We shall shortly verify that the recursive algorithm of the theorem, when applied to a conveniently chosen structured matrix, leads to a cascade of $J-$lossless first-order sections, each of which has an evident interpolation property. This is due to the fact that linear systems have "transmission zeros": certain inputs at certain frequencies yield zero outputs. More specifically, each section of the cascade will be shown to be characterized by a $(p+q)\times(p+q)$ rational transfer matrix, $\Theta_i(z)$ say, that has a left zero-direction vector g_i at a frequency f_i, viz.,

$$g_i\Theta_i(f_i) \;\equiv\; \left[\begin{array}{cc} a_i & b_i \end{array}\right]\left[\begin{array}{cc} \Theta_{i,11} & \Theta_{i,12} \\ \Theta_{i,21} & \Theta_{i,22} \end{array}\right](f_i) \;=\; \mathbf{0}\;,$$

which makes evident (with the proper partitioning of the row vector g_i and the matrix function $\Theta_i(z)$) the following interpolation property: $a_i\Theta_{i,12}\Theta_{i,22}^{-1}(f_i) = -b_i$. This suggested to us that one way of solving an interpolation problem is to show how to construct an appropriate cascade so that the local interpolation properties of the elementary sections combine in such a way that the cascade yields a solution to the global interpolation problem. All possible interpolants can then be parametrized by attaching various loads to the right-hand side of the cascade system.

The main feature of our derivation is that it approaches the subject from a matrix-factorization point of view, and that it relies almost completely on matrix-based arguments: we use the interpolation data to construct a so-called generator matrix; the generator is then used to start a recursive algorithm for the computation of the triangular factorization of an associated structured matrix; each step of the algorithm yields a first-order $J-$lossless section with an intrinsic "blocking" or "transmission zero" property; these local blocking relations are then shown to combine to yield the desired interpolation conditions.

A condensed discussion of our approach follows. For this purpose, we focus on the case of a positive-definite Hermitian matrix R that satisfies

$$R - FRF^* = G\left[\begin{array}{cc} I_p & \mathbf{0} \\ \mathbf{0} & -I_q \end{array}\right]G^*\;,\quad r=(p+q),\;\; R>\mathbf{0},$$

with a stable matrix F (i.e., $|f_i|<1$). As stated in Theorem 4.2, and Lemma 6.1, the triangular factorization of R can be recursively computed via the following procedure: start with $G_0=G, F_0=F$, and repeat:

1. At step i we have F_i and G_i. Let g_i denote the top row of G_i.

2. Choose a $J-$unitary rotation matrix Θ_i that reduces g_i to the form $g_i\Theta_i = \begin{bmatrix} \delta_i & \mathbf{0} \end{bmatrix}$ with a single nonzero entry in the first position. This is always possible since R is positive-definite.

3. The next generator is then computed via the expression

$$\begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = \Phi_i G_i \Theta_i \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + G_i \Theta_i \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & I_{r-1} \end{bmatrix} , \tag{6.4a}$$

where $\Phi_i = (F_i - f_i I_{n-i})(I_{n-i} - f_i^* F_i)^{-1}$.

4. Each step of the algorithm also gives rise to a first-order $(p+q) \times (p+q)$ $J-$lossless section

$$\Theta_i(z) = \Theta_i \begin{bmatrix} \frac{z-f_i}{1-zf_i^*} & \mathbf{0} \\ \mathbf{0} & I_{r-1} \end{bmatrix} . \tag{6.4b}$$

The relevant observation to make here is that each section $\Theta_i(z)$ has an obvious blocking property, which results from the easily verified equality $g_i\Theta_i(f_i) = \mathbf{0}$,

$$g_i\Theta_i(f_i) = g_i\Theta_i \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & I_{r-1} \end{bmatrix} = \begin{bmatrix} \delta_i & \mathbf{0} \end{bmatrix} \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & I_{r-1} \end{bmatrix} = \mathbf{0} \, .$$

If we further partition the row vector g_i and the transfer matrix $\Theta_i(z)$ accordingly with J, we conclude that

$$\begin{bmatrix} a_i & b_i \end{bmatrix} \begin{bmatrix} \Theta_{i,11} & \Theta_{i,12} \\ \Theta_{i,21} & \Theta_{i,22} \end{bmatrix} (f_i) \; = \; \mathbf{0} \; ,$$

which makes evident the following interpolation property: $a_i\Theta_{i,12}\Theta_{i,22}^{-1}(f_i) = -b_i$. (The invertibility of $\Theta_{i,22}(z)$ inside the unit disc is guaranteed by the $J-$losslessness of $\Theta_i(z)$.). Therefore, each first-order section satisfies a local blocking and/or interpolation property. This fact plays a central role in our approach to interpolation problems. While one can use purely algebraic arguments, we think it is useful to present a physical (network-theoretic) interpretation as well. The following example illustrates the main points in our construction (for more involved examples and for a detailed discussion on the approach described herein, the reader may consult [171, 182]).

We consider the well-known tangential Nevanlinna-Pick problem where one is interested in finding one (or even all) $p \times q$ Schur matrix functions $S(z)$ (i.e., analytic and strictly bounded by unity in $|z| < 1$) that satisfy the tangential conditions

$$u_i S(f_i) = v_i \; , \quad \text{for} \quad i = 0, 1, \ldots, n-1 \quad \text{and} \quad |f_i| < 1 \, . \tag{6.5a}$$

Here, u_i and v_i are $1 \times p$ and $1 \times q$ row vectors, respectively. To solve this problem we introduce the matrices F, G, and J: $F = \text{diagonal}\,\{f_0, \ldots, f_{n-1}\}$,

$$G = \begin{bmatrix} u_0 & v_0 \\ u_1 & v_1 \\ \vdots & \vdots \\ u_{n-1} & v_{n-1} \end{bmatrix} \quad \text{and} \quad J = \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} \; , \tag{6.5b}$$

and apply the recursive procedure (6.4a) to F and G. The "Blaschke" matrix Φ_i is now a diagonal matrix since F is also diagonal, viz.,

$$\Phi_i = \text{diagonal}\,\{0, \frac{f_{i+1}-f_i}{1-f_{i+1}f_i^*}, \ldots, \frac{f_{n-1}-f_i}{1-f_{n-1}f_i^*}\}\,.$$

This leads to a cascade $\mathbf{\Theta}(z)$ of n first-order $J-$lossless sections,

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

It is now instructive to see how the local blocking properties of the individual sections combine together to impose a global blocking property on the entire cascade. We start with the first section and invoke its blocking property: $g_0\Theta_0(f_0) = \mathbf{0}$, where g_0 is the first row of G. It thus follows that

$$g_0\mathbf{\Theta}(f_0) = \underbrace{g_0\Theta_0(f_0)}_{\mathbf{0}}\,\Theta_1(f_0)\ldots\Theta_{n-1}(f_0) = \mathbf{0}\,.$$

In system-theoretic terms this means that when the first row of G is fed into the cascade $\mathbf{\Theta}(z)$ we get a zero-output at the 'frequency' f_0,

$$\begin{bmatrix} u_0 & v_0 \end{bmatrix}\mathbf{\Theta}(f_0) = \mathbf{0}\,.$$

But what happens when the second row of G is fed into the cascade? To answer this question, let us first check how does the first section of the cascade react to the second row of G. That is, let us evaluate the quantity $\begin{bmatrix} u_1 & v_1 \end{bmatrix}\Theta_0(f_1)$. Using the definition of $\Theta_0(z)$ we see that

$$\Theta_0(f_1) = \Theta_0\begin{bmatrix} B_0(f_1) & 0 \\ 0 & I_{r-1}\end{bmatrix} = \Theta_0\begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & I_{r-1}\end{bmatrix} + B_0(f_1)\Theta_0\begin{bmatrix} 1 & 0 \\ 0 & \mathbf{0}\end{bmatrix}\,.$$

Therefore, $\begin{bmatrix} u_1 & v_1 \end{bmatrix}\Theta_0(f_1)$ is equal to

$$\begin{bmatrix} u_1 & v_1 \end{bmatrix}\Theta_0\begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & I_{r-1}\end{bmatrix} + \begin{bmatrix} u_1 & v_1 \end{bmatrix}B_0(f_1)\Theta_0\begin{bmatrix} 1 & 0 \\ 0 & \mathbf{0}\end{bmatrix}\,.$$

But if we compare the second rows on both sides of the generator recursion (6.4a) we see that the above expression should be equal to the top row of G_1. That is,

$$g_1 = \begin{bmatrix} u_1 & v_1 \end{bmatrix}\Theta_0\begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & I\end{bmatrix} + B_0(f_1)\begin{bmatrix} u_1 & v_1 \end{bmatrix}\Theta_0\begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix} = \begin{bmatrix} u_1 & v_1 \end{bmatrix}\Theta_0(f_1)\,.$$

This shows that when the second row of G enters the cascade we get the top row of G_1 at the output of the first section at the 'frequency' f_1; thus leading to

$$\begin{aligned}\begin{bmatrix} u_1 & v_1 \end{bmatrix}\mathbf{\Theta}(f_1) &= \underbrace{\begin{bmatrix} u_1 & v_1 \end{bmatrix}\Theta_0(f_1)}_{g_1}\,\Theta_1(f_1)\ldots\Theta_{n-1}(f_1), \\ &= \underbrace{g_1\Theta_1(f_1)}_{\mathbf{0}}\,\Theta_2(f_1)\ldots\Theta_{n-1}(f_1), \\ &= \mathbf{0}\,,\end{aligned}$$

which shows that the second row of G also annihilates the entire cascade at the frequency f_1. This argument can be continued to show that the remaining rows of G are also zero directions of the cascade $\mathbf{\Theta}(z)$ at the corresponding f_i, viz.,

$$\begin{bmatrix} u_i & v_i \end{bmatrix}\mathbf{\Theta}(f_i) = \mathbf{0}\,. \tag{6.5c}$$

If we now partition the $J-$lossless cascade $\mathbf{\Theta}(z)$ accordingly with J,

$$\mathbf{\Theta}(z) = \begin{bmatrix} \mathbf{\Theta}_{11}(z) & \mathbf{\Theta}_{12}(z) \\ \mathbf{\Theta}_{21}(z) & \mathbf{\Theta}_{22}(z) \end{bmatrix} ,$$

we then conclude from (6.5c) that the $p \times q$ Schur matrix function,

$$S(z) = -\mathbf{\Theta}_{12}(z)\mathbf{\Theta}_{22}^{-1}(z) ,$$

is one solution that satisfies $u_i S(f_i) = v_i$. That is, it solves the tangential Nevanlinna-Pick interpolation problem. We again remark that the invertibility of $\mathbf{\Theta}_{22}(z)$ inside the unit disc is guaranteed by the $J-$losslessness of $\mathbf{\Theta}(z)$, which also allows us to conclude that $S(z)$ is of Schur type. Moreover, all solutions $S(z)$ to the tangential Nevanlinna-Pick problem are in fact given by a linear fractional transformation of a Schur matrix function $K(z)$ [20, 71, 182] ($\|K\|_\infty < 1$), viz.,

$$S(z) = -\left[\mathbf{\Theta}_{11}(z)K(z) + \mathbf{\Theta}_{12}(z)\right]\left[\mathbf{\Theta}_{21}(z)K(z) + \mathbf{\Theta}_{22}(z)\right]^{-1} . \tag{6.6}$$

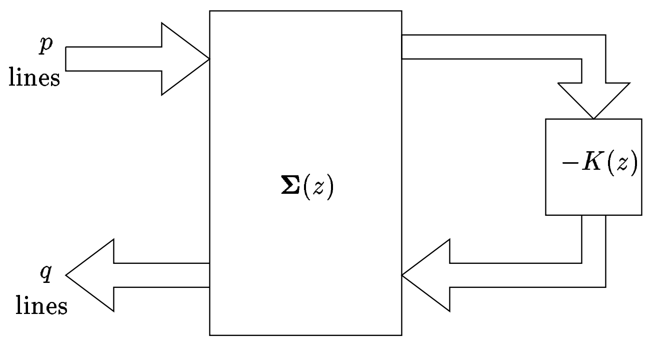

FIG. 6.2. *Scattering interpretation of the interpolating solutions.*

The solutions $S(z)$ in (6.6) have a scattering interpretation as shown in Figure 6.2, where $\mathbf{\Sigma}(z)$ is the scattering matrix defined by

$$\mathbf{\Sigma}(z) = \begin{bmatrix} \mathbf{\Theta}_{11} - \mathbf{\Theta}_{12}\mathbf{\Theta}_{22}^{-1}\mathbf{\Theta}_{21} & -\mathbf{\Theta}_{12}\mathbf{\Theta}_{22}^{-1} \\ \mathbf{\Theta}_{22}^{-1}\mathbf{\Theta}_{21} & \mathbf{\Theta}_{22}^{-1} \end{bmatrix}(z) .$$

That is, $S(z)$ is the transfer matrix from the top left ($1 \times p$) input to the bottom left ($1 \times q$) output, with a Schur-type load ($-K(z)$) at the right end. Here we only remark that the scattering matrix $\mathbf{\Sigma}(z)$ is a so-called inner dilation of $-\mathbf{\Theta}_{12}(z)\mathbf{\Theta}_{22}^{-1}(z)$ and satisfies $\mathbf{\Sigma}(z)\mathbf{\Sigma}^*(z) = I$ on $|z| = 1$. The $\mathbf{\Sigma}(z)$ can also be obtained as a cascade of elementary sections $\Sigma_i(z)$, which are defined in terms of the $\Theta_i(z)$. In the $\Sigma_i(z)$-cascade, signals flow in both directions; this yields a so-called (generalized) transmission line.

The same line of reasoning can be used to solve more involved interpolation problems of the Hermite-Fejér type, as detailed in [171, 182]. But more important perhaps is to stress that the arguments used in the solution of the above interpolation problem are essentially matrix-based arguments. This has the nice feature of being equally applicable to time-variant extensions of classical interpolation and matrix completion problems, as detailed in [174].

6.1.2. Unconstrained Interpolation Problems. Actually the arguments can also be extended to the very old class of unconstrained interpolation problems. These problems have a very long history, associated with many classical results of Lagrange, Hermite, Prony, Padé, and other famous names. In recent years, several authors have approached these problems from a system-theoretical point of view, where the main idea was to find a global transfer matrix that provides a linear fractional parametrization of all solutions (see, e.g., [15, 21] and the references therein).

In [29, 28], we showed how a generalized Schur algorithm for non Hermitian matrices [120, 171] (described in Section 7.4.2) can be used to give a recursive solution. The discussion that follows can be extended to tangential interpolation problem [28], along the same lines discussed in the previous section for the tangential Nevanlinna Pick problem. But here, for simplicity, we shall illustrate the results by considering a scalar example that reads as follows: given a set of complex pairs (f_i, β_i), find all irreducible rational interpolants $s(z) = n(z)/d(z)$ such that $s(f_i) = \beta_i$. In many applications the interpolants are also required to satisfy certain minimality constraints, where the complexity of a rational solution $s(z)$ is measured in terms of its McMillan degree. This is defined as the maximum of the degrees of the numerator and denominator. In these cases, we would also like to determine the admissible degrees of complexity of the rational interpolants, as well as the minimal degree of complexity and the minimal interpolant(s).

The approach we described earlier in the analytic case extends smoothly to the unconstrained case [29]. Here we only briefly address the major points. Let us first show how to determine an irreducible interpolant $s(z)$. For this purpose, we consider an appropriate non-Hermitian displacement structure of the form

$$R - FRA^* = GJB^*. \tag{6.7}$$

The interpolation data is collected into F, G and J as before, viz.,

$$F = \text{diagonal}\{f_0, \ldots, f_{n-1}\},$$

$$G = \begin{bmatrix} 1 & -\beta_0 \\ 1 & -\beta_1 \\ \vdots & \vdots \\ 1 & -\beta_{n-1} \end{bmatrix}, \quad J = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

The matrices A and B are chosen freely so as to guarantee the strong regularity of R. (Details on how to choose A and B are omitted here but can be found in [28]. Also, the strong regularity condition may be relaxed but that is not of major consequence to the argument that follows – see [28]). Though the non Hermitian case is only treated later (Section 7.4.2), it suffices here to note that applying the array algorithm to (F, G) and to (A, B) leads to two 2×2 cascades $\mathbf{\Theta}(z)$ and $\mathbf{\Gamma}(z)$,

$$\mathbf{\Theta}(z) = \Theta_0(z) \ldots \Theta_{n-1}(z) \;\; , \;\; \mathbf{\Gamma}(z) = \Gamma_0(z) \ldots \Gamma_{n-1}(z) \; ,$$

which satisfy $\mathbf{\Theta}(z) J \mathbf{\Gamma}^*(\omega) = J$ on $zw^* = 1$. Moreover, each first-order section possesses a local blocking property,

$$g_i \Theta_i(f_i) = \mathbf{0} \;\; , \;\; b_i \Gamma_i(a_i) = \mathbf{0} \; .$$

As before, and following the same arguments as in the Nevanlinna-Pick case, the cascades $\mathbf{\Theta}(z)$ and $\mathbf{\Gamma}(z)$ also turn out to satisfy global blocking results. More precisely, the rows of G turn out to annihilate the output of the cascade $\mathbf{\Theta}(z)$ at the corresponding frequencies f_i,

$$\begin{bmatrix} 1 & -\beta_i \end{bmatrix} \mathbf{\Theta}(f_i) = \mathbf{0} \; .$$

If we partition $\mathbf{\Theta}(z)$ accordingly with J, then all rational interpolants can be shown to be parametrized as follows (see, *e.g.*, [15, 21, 29]):

(6.8a) $$s(z) = \frac{p(z)\,\Theta_{11}(z) + q(z)\,\Theta_{12}(z)}{p(z)\,\Theta_{21}(z) + q(z)\,\Theta_{22}(z)} \; ,$$

where $p(z)$ and $q(z)$ are coprime polynomials such that $p(f_i)\,\Theta_{21}(f_i) + q(f_i)\,\Theta_{22}(f_i) \neq 0$. In particular, if $\Theta_{12}(z)$ and $\Theta_{22}(z)$ are irreducible, which is always guaranteed in the analytic case but not here, then $s(z) = \Theta_{12}(z)\Theta_{22}^{-1}(z)$ is an irreducible interpolant.

The above unconstrained interpolation problem can also be solved by expressing all interpolants in terms of the Lagrange interpolating polynomial as was done, for example, in [15]. As a matter of interest, we now show how to obtain this solution via the generalized Schur algorithm (of Section 7.4). Recall that we are essentially free to choose A and B so as to guarantee the strong regularity of R. Each such choice would lead to a cascade $\mathbf{\Theta}(z)$ that parametrizes all solutions of the unconstrained problem as in (6.8a). We now exhibit a particular choice for A and B that guarantees that the particular solution $s(z) = \Theta_{12}(z)\Theta_{22}^{-1}(z)$ will coincide with the Lagrange polynomial.

We start by defining the Lagrange interpolating polynomial. Consider the n-dimensional linear space of polynomials of degree at most $n-1$, in which a basis $\{L_0(z), L_1(z), \ldots, L_{n-1}(z)\}$ can be defined as follows:

$$L_i(f_j) = \begin{cases} 1 & \text{if} \;\; j = i \\ 0 & \text{if} \;\; j \neq i \end{cases} \; .$$

That is, each basis polynomial $L_i(z)$ assumes the value 1 at f_i and the value zero at the other points $f_j, \; j \neq i$. Now, a polynomial solution to the unconstrained problem can be obtained as a linear combination of $L_0(z), L_1(z), \ldots, L_{n-1}(z)$ with coefficients $\beta_0, \beta_1, \ldots, \beta_{n-1}$, viz., $L(z) = \sum_{i=0}^{n-1} \beta_i\, L_i(z)$. The polynomial $L(z)$ is called the Lagrange interpolating polynomial and it constitutes the unique interpolating solution in the space of polynomials of order at most $n-1$.

We can write down an explicit expression for each $L_i(z)$ as follows: $L_i(z)$ has zeros at $\{f_0, f_1, \ldots, f_{i-1}, f_{i+1}, \ldots, f_{n-1}\}$. Therefore,

$$L_i(z) = \frac{\prod_{j\neq i}(z - f_j)}{\prod_{j \neq i}(f_i - f_j)} = \frac{1}{\pi'(f_i)}\, \frac{\pi(z)}{(z - f_i)} \; ,$$

where $\pi(z) = \prod_{j=0}^{n-1}(z - f_j)$. Thus, we obtain the celebrated formula

$$L(z) = \sum_{i=0}^{n-1} \beta_i\, \frac{1}{\pi'(f_i)}\, \frac{\pi(z)}{(z - f_i)} \; .$$

It is also known that all rational solutions can be parametrized in terms of $L(z)$, viz.,

(6.8b) $$s(z) = L(z) + \pi(z)\, \frac{p(z)}{q(z)} \; ,$$

for coprime polynomials $p(z)$ and $q(z)$ such that $q(f_i) \neq 0$.

The question is how do we choose A and B so as to obtain the Lagrange solution. For this, we define A as the lower triangular shift matrix ($A = Z$) and choose B to be equal to

$$B = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix}^{\mathbf{T}} .$$

It can be shown [29] that the resulting R is strongly regular and that the resulting cascade $\mathbf{\Theta}(z)$ is a polynomial matrix of the form

$$\mathbf{\Theta}(z) = \begin{bmatrix} \pi(z) & L(z) \\ 0 & 1 \end{bmatrix} ,$$

which explicitly contains $L(z)$. Observe that substituting the above expression for $\mathbf{\Theta}(z)$ into the linear fractional parametrization formula (6.8a) leads to (6.8b).

But what if we want to obtain a minimal degree interpolant? Following [14, 15, 21] we have that, provided $\mathbf{\Theta}(z)$ is a column reduced polynomial matrix [108, p.387], the McMillan degree of an interpolant is given by

$$\max\{\deg n(z), \deg d(z)\} = \max\{\kappa_1 + \deg p(z), \kappa_2 + \deg q(z)\} ,$$

where κ_1 and $\kappa_2 \geq \kappa_1$ are the column indices of $\mathbf{\Theta}(z)$. Moreover, the minimal solution of the unconstrained rational interpolation problem can be obtained by either

$$s_{\min}(z) = \frac{\Theta_{11}(z)}{\Theta_{21}(z)} ,$$

provided that $\Theta_{11}(z)$ and $\Theta_{21}(z)$ are coprime, or

$$s_{\min}(z) = \frac{p(z)\,\Theta_{11}(z) + \Theta_{12}(z)}{p(z)\,\Theta_{21}(z) + \Theta_{22}(z)} ,$$

where $\deg p(z) \leq \kappa_2 - \kappa_1$ and $p(f_i)\,\Theta_{21}(f_i) + \Theta_{22}(f_i) \neq 0$, when $\Theta_{11}(z)$ and $\Theta_{21}(z)$ share some common roots. In the first case, there is a minimal solution with complexity κ_1, while in the second case there exists a family of minimal interpolants whose complexities are equal to κ_2. In both cases, there exist infinitely many interpolants with complexity larger than κ_2.

To obtain a column reduced transfer matrix $\mathbf{\Theta}(z)$ via the non-Hermitian generalized Schur algorithm we restrict A to an arbitrary lower triangular matrix whose diagonal and first subdiagonal are zero and choose an arbitrary B so as to guarantee the strong regularity of R. Then it can be verified [29] that the generalized Schur algorithm yields a column reduced polynomial matrix $\mathbf{\Theta}(z)$.

In concluding this section, we should remark that with the proper formulation, many problems in circuit, control, computation, communications, and signal processing can be reformulated as interpolation problems. One such application to the so-called four-block problem in H^∞-control can be found in [58]. On the other hand, it is also noteworthy that the solution of interpolation problems can be reduced to the determination of an appropriate fast matrix triangularization algorithm. This constructive view provides a nice complement to the many abstract formulations (esp. those connected with lifting theory) of the important topic of interpolation theory.

7. Other Forms of Generalized Schur Algorithms. The generator recursions that we exhibited in Section 4 were all based on first reducing a generator matrix to proper form by employing appropriate $J-$unitary operators. The significance of this step was that it allowed us to provide simple array interpretations for the varied generator recursions. But it is not difficult to see that the array recursions only provide one possibility for the update of the generators.

A greater level of generality is desirable for several reasons, among which we highlight the following. First, it is important in its own right to exhibit the most general form of the relation that might exist between generators of two successive Schur complements of a structured matrix. Secondly, once this general relation is established, it will allow us, in several instances, to verify that what may appear at first glance to be a new generator recursion is in fact nothing but a special case of the general result. Thirdly, a general recursion will also allow us to handle cases where the propagation of the generators in proper (or array) form may not be necessarily the most convenient. This occurs, for example, in the study of Bezoutian matrices [132] (see Section 8.2), in extensions of the displacement structure theory to the time-variant setting [174] (see Section 10), and in the solution of a general maximum entropy problem [57]. It also occurs when the displacement equation does not uniquely determine the matrix itself (see, e.g., the discussion after Theorem 7.4). Finally, and perhaps a most important reason, is that the general result will further allow us to make explicit connections with results in other places in the literature, especially in the study of lossless embedding (or dilation) and of lossless systems (see, e.g., [20, 67, 71, 83, 124, 130, 135, 138]), where the results are often displayed in forms that fit into our general expressions.

For all the above reasons, we now move to exhibit the general form of the generator recursion, but always keeping in mind the desire to highlight connections with other results, especially with the so-called embedding or dilation relations that are essential in the study of lossless and $J-$lossless systems both in discrete as well as continuous time. The derivation that follows is essentially based on the same ideas as before: starting with a structured matrix R, we write down a displacement equation for its first Schur complement R_1, and then verify that it can be expressed, by completion of squares, in the form $G_1JG_1^*$, for some generator matrix G_1 that is related to G, and so on. But we shall proceed gradually: we shall first consider Toeplitz-like matrices, which lead to a discrete-time embedding result, followed by Hankel-like matrices, which lead to a continuous-time embedding result, and we shall finish with a generalized displacement that includes both Toeplitz- and Hankel-like matrices, as well as other matrices, as special cases.

7.1. Discrete-Time Embedding Relations: Completion of Squares. The reasoning given in this section invokes only matrix-based arguments (see, *e.g.,* [171, 182, 184]), and is especially convenient for extensions of the displacement structure concept to the time-variant setting, as discussed in [171, 174, 184].

We start with a Hermitian strongly regular $n \times n$ matrix $R = [r_{mj}]_{m,j=0}^{n-1}$ that satisfies a displacement equation of the form

$$R - FRF^* = GJG^* \,, \tag{7.1a}$$

where F is an *arbitrary* lower triangular matrix with *arbitrary* diagonal entries that are denoted by $\{f_i\}$, G is an $n \times r$ generator matrix, and $J = (I_p \oplus -I_q)$, $p + q = r$.

We say that R is a Toeplitz-like matrix with respect to (F, G, J); our purpose is to characterize all (same size) generators of the Schur complements R_i.

First note that if l_0 and g_0 denote the first column of R and the top row of G, respectively, we can see from (7.1a) that l_0 and the top left-corner element r_{00} of R obey the identities

$$l_0 = Fl_0 f_0^* + GJg_0^* \ , \quad d_0(1-|f_0|^2) = g_0 J g_0^* \ . \tag{7.1b}$$

We can now form the Schur complement R_1 from $\tilde{R}_1 = R - l_0 d_0^{-1} l_0^*$. Let F_1 be the submatrix obtained after deleting the first row and column of F. Using (4.1a) and (7.1a) we can prove the following lemma.

LEMMA 7.1. *Consider an $n \times n$ Hermitian strongly-regular matrix that satisfies a displacement equation of the form*

$$R - FRF^* = GJG^*,$$

where F is lower triangular with diagonal entries $\{f_i\}$. Then the first Schur complement R_1 satisfies

$$R_1 - F_1 R_1 F_1^* = G_1 J G_1^*,$$

where G_1 is an $(n-1) \times r$ matrix that is computed from G as follows

$$\begin{bmatrix} \mathbf{0} \\ G_1 \end{bmatrix} = Fl_0 h_0^* J + GJk_0^* J \ , \tag{7.2a}$$

and h_0 and k_0 are, respectively, $r \times 1$ and $r \times r$ arbitrary matrices chosen so as to satisfy the following $(d_0 \oplus J)-$unitary embedding (or dilation) relation

$$\begin{bmatrix} f_0 & g_0 \\ h_0 & k_0 \end{bmatrix} \begin{bmatrix} d_0 & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} \begin{bmatrix} f_0 & g_0 \\ h_0 & k_0 \end{bmatrix}^* = \begin{bmatrix} d_0 & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} . \tag{7.2b}$$

Proof. The proof can be obtained via straightforward manipulations as follows: using (4.1a) and (7.1b) we find that

$$\tilde{R}_1 - F\tilde{R}_1 F^* = -Fl_0 \underbrace{\frac{f_0^* g_0}{d_0}}_{-h_0^* J k_0} JG^* - GJ \underbrace{\frac{g_0^* f_0}{d_0}}_{-k_0^* J h_0} l_0^* F^* + Fl_0 \underbrace{\frac{g_0 J g_0^*}{d_0^2}}_{h_0^* J h_0} l_0^* F^* +$$

$$GJ \underbrace{\left\{ J - \frac{g_0^* g_0}{d_0} \right\}}_{k_0^* J k_0} JG^* \ . \tag{7.3a}$$

We now verify that the right-hand side of the above expression can be made into a *perfect square* by introducing some auxiliary quantities. Consider an $r \times 1$ column vector h_0 and an $r \times r$ matrix 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 show later that this is always possible):

$$h_0^* J h_0 = \frac{g_0 J g_0^*}{d_0^2} \ , \quad k_0^* J k_0 = J - \frac{g_0^* g_0}{d_0} \ , \quad k_0^* J h_0 = -\frac{f_0 g_0^*}{d_0} \ . \tag{7.3b}$$

Using $\{h_0, k_0\}$, we can rewrite the right-hand side of (7.3a) in the form

$$GJk_0^* J k_0 JG^* + GJk_0^* J h_0 l_0^* F^* + Fl_0 h_0^* J k_0 JG^* + Fl_0 h_0^* J h_0 l_0^* F^*,$$

which can be factored as $\tilde{G}_1 J \tilde{G}_1^*$, where $\tilde{G}_1 = F l_0 h_0^* J + G J k_0^* J$. Recall that the first row and column of $\tilde{R}_1$ are zero. Hence, the first row of $\tilde{G}_1$ is zero,

$$\tilde{G}_1 = \begin{bmatrix} \mathbf{0} \\ G_1 \end{bmatrix} .$$

Moreover, it follows from (7.3b) (and the expression for d_0 in (7.1b)) that $\{f_0, g_0, h_0, k_0\}$ satisfy the relation

$$\begin{bmatrix} f_0 & g_0 \\ h_0 & k_0 \end{bmatrix}^* \begin{bmatrix} d_0^{-1} & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} \begin{bmatrix} f_0 & g_0 \\ h_0 & k_0 \end{bmatrix} = \begin{bmatrix} d_0^{-1} & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} ,$$

which is equivalent to (7.2b). □

Several remarks are due here. To begin with, the statement of the lemma requires only the strong-regularity of R, thus assuring $d_0 \neq 0$. But no condition is imposed on the diagonal entries of F, as opposed to the array form of Theorem 4.2 where $(1 - f_i f_j^*) \neq 0$ is required in order to guarantee the existence of the successive Blaschke matrices Φ_i. The Φ_i matrices are avoided here because the above completion-of-squares argument allows us to update the generator matrix as in (7.2a), which is expressed in terms of l_i rather than Φ_i.

Note also how naturally the so-called embedding (or dilation) relation (7.2b) arises in our framework. It will be seen in the next section that relations of this type play a fundamental role in characterizing $J-$lossless transfer functions. Finally, observe that the generator recursion (7.2a) and the identity (7.1b) for l_0 can be combined and rewritten compactly into the following revealing expression

$$\begin{bmatrix} l_0 & \mathbf{0} \\ & G_1 \end{bmatrix} = \begin{bmatrix} F_0 l_0 & G \end{bmatrix} \begin{bmatrix} f_0^* & h_0^* J \\ J g_0^* & J k_0^* J \end{bmatrix} . \tag{7.4}$$

This identifies a first-order system that arises in state-space form specified by the system matrix

$$\begin{bmatrix} f_0^* & h_0^* J \\ J g_0^* & J k_0^* J \end{bmatrix} .$$

Hence, the rows of G and G_1 can be regarded as inputs and outputs of this system, respectively. The entries of $F_0 l_0$ and l_0 can be regarded as the corresponding current and future states. This explains the terms "discrete-time embedding" that appear in the title of this section. Further clarification will be given in the next section.

Now the Schur complementation process can be repeated via the defining relation (4.1b) where l_i and d_i denote the first column and the $(0,0)$ entry of R_i, respectively. This leads to the following generalization of the result of Lemma 7.1 [171, 175].

LEMMA 7.2. *Consider an $n \times n$ strongly-regular Hermitian matrix R with Toeplitz-like displacement as in Lemma 7.1. The successive Schur complements of R with respect to its leading $i \times i$ submatrices are also structured,*

$$R_i - F_i R_i F_i^* = G_i J G_i^*,$$

where F_i is the submatrix obtained by deleting the first row and column of F_{i-1}, and G_i is an $(n-i) \times r$ generator matrix that satisfies, along with l_i (the 1^{st} column of R_i), the following recursion

$$\begin{bmatrix} l_i & \mathbf{0} \\ & G_{i+1} \end{bmatrix} = \begin{bmatrix} F_i l_i & G_i \end{bmatrix} \begin{bmatrix} f_i^* & h_i^* J \\ J g_i^* & J k_i^* J \end{bmatrix} , \tag{7.5a}$$

where g_i is the top row of G_i, and h_i and k_i are arbitrary $r \times 1$ and $r \times r$ matrices, respectively, chosen so as to satisfy the embedding relation

$$\begin{bmatrix} f_i & g_i \\ h_i & k_i \end{bmatrix} \begin{bmatrix} d_i & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} \begin{bmatrix} f_i & g_i \\ h_i & k_i \end{bmatrix}^* = \begin{bmatrix} d_i & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} . \tag{7.5b}$$

Also, d_i and l_i satisfy the relations

$$d_i(1 - |f_i|^2) = g_i J g_i^* \ , \quad (I_{n-i} - f_i^* F_i) l_i = G_i J g_i^* \ . \tag{7.5c}$$

In order to apply the generator recursion (7.5a), we still need to show how to choose the arbitrary parameters h_i and k_i so as to satisfy (7.5b). We shall see later that all possible values of h_i and k_i are (almost) completely determined by the known quantities $\{f_i, g_i, d_i\}$.

7.1.1. Elementary Discrete-Time State-Space Sections. As noted earlier, the significance of (7.5a) is that it can be regarded as specifying an $r \times r$ first-order discrete-time system (in state-space form) that performs a state-space transformation of the form

$$\begin{bmatrix} \mathbf{x}_{j+1} & \mathbf{y}_j \end{bmatrix} = \begin{bmatrix} \mathbf{x}_j & \mathbf{w}_j \end{bmatrix} \begin{bmatrix} f_i^* & h_i^* J \\ J g_i^* & J k_i^* J \end{bmatrix} , \tag{7.6a}$$

where $\mathbf{x}_j$ denotes the current state, $\mathbf{y}_j$ the *row* output vector, and $\mathbf{w}_j$ the $1 \times r$ *row* input vector at time j. In system theory, the *transfer function* is an important system invariant and it can be seen to be given by the expression

$$\Theta_i(z) = J k_i^* J + J g_i^* \left[z^{-1} - f_i^* \right]^{-1} h_i^* J \ . \tag{7.6b}$$

Using the embedding relation (7.5b) (or the expressions similar to (7.3b) for h_i and k_i) we readily conclude that

$$\Theta_i(z) J \Theta_i^*(z) = J + \frac{J g_i^* g_i J}{d_i} \, \frac{z z^* - 1}{(1 - z f_i^*)(1 - z^* f_i)} , \tag{7.6c}$$

which shows that the transfer function $\Theta_i(z)$ satisfies the normalization condition $\Theta_i(z) J \Theta_i^*(z) = J$ on $|z| = 1$.

7.1.2. Generalized Schur Algorithm for Toeplitz-Like Matrices. The generator recursion (7.5a) in Lemma 7.2 is still incomplete since we have not yet shown how to choose the free parameters h_i and k_i.

LEMMA 7.3. *All possible choices of h_i and k_i that satisfy the embedding relation (7.5b), with $d_i \neq 0$, can be expressed in terms of f_i, g_i and d_i as follows:*

$$h_i = \Theta_i^{-1} \left\{ \frac{1}{d_i} \frac{\tau_i - f_i}{1 - \tau_i f_i^*} J g_i^* \right\} \quad \textit{and} \quad k_i = \Theta_i^{-1} \left\{ I_r - \frac{1}{d_i} \frac{J g_i^* g_i}{1 - \tau_i f_i^*} \right\} , \tag{7.7}$$

for an arbitrary $J-$unitary matrix Θ_i and for an arbitrary scalar τ_i on the unit circle ($|\tau_i| = 1$).

Proof. Let τ_i be an arbitrary point on the unit circle, and let us try to determine a pair $(\hat{h}_i, \hat{k}_i)$ such that the corresponding transfer matrix, $\hat{\Theta}_i(z)$ (as in (7.6b)), satisfies $\hat{\Theta}_i(\tau_i^{-*}) = I_r$. Note that this is consistent with the requirement that $\hat{\Theta}_i(z)$ has to satisfy the normalization condition $\hat{\Theta}_i(z) J \hat{\Theta}_i^*(z) = J$ for all $|z| = 1$.

The condition $\hat{\Theta}_i(\tau_i^{-*}) = I_r$ implies that $\hat{k}_i + \hat{h}_i(\tau_i - f_i)^{-1}g_i = I_r$. But the embedding relation (7.5b) implies that $\hat{h}_i d_i f_i^* + \hat{k}_i J g_i^* = \mathbf{0}$. Therefore, we can solve for $\hat{h}_i$ and $\hat{k}_i$, leading to

$$\hat{h}_i = \frac{1}{d_i}\frac{\tau_i - f_i}{1-\tau_i f_i^*}Jg_i^* \quad \text{and} \quad \hat{k}_i = I_r - \frac{1}{d_i}\frac{Jg_i^* g_i}{1-\tau_i f_i^*} .$$

The claim is that all other choices of h_i and k_i are related to $\hat{h}_i$ and $\hat{k}_i$ via $h_i = \Theta_i^{-1}\hat{h}_i$ and $k_i = \Theta_i^{-1}\hat{k}_i$, for an arbitrary $J-$unitary matrix Θ_i. To check this, let $\Theta_i(z)$ be the transfer matrix of any other valid choice (h_i, k_i). Clearly, $\Theta_i(\tau_i^{-*})J\Theta_i^*(\tau_i^{-*}) = J$, since $|\tau_i| = 1$. Hence, $\Theta_i(\tau_i^{-*})$ is invertible and we define a transfer matrix $\hat{\Theta}_i(z)$ by $\hat{\Theta}_i(z) = \Theta_i(z)\Theta_i^{-1}(\tau_i^{-*})$. This function satisfies $\hat{\Theta}_i(\tau_i^{-*}) = I_r$. Using the fact that this condition is satisfied by $(\hat{h}_i, \hat{k}_i)$ as above, we readily conclude that

$$h_i = \Theta_i^{-1}(\tau_i^{-*})\hat{h}_i, \quad k_i = \Theta_i^{-1}(\tau_i^{-*})\hat{k}_i.$$

This proof is patterned on one in [135]. □

We are now in a position to state the generalized Schur algorithm for Toeplitz-like matrices. For this purpose, we substitute the above expressions for h_i and k_i into Lemma 7.2. This leads to the next theorem, which is a generalization of Theorem 4.2 in two important respects. First, it does not involve the intermediate step of reducing generators to proper form and, secondly, it avoids the requirement $(1 - f_i f_j^*) \neq 0$ for all i, j. But right after the statement of the theorem we shall see that if this requirement is further imposed, then we can rewrite the result of the theorem in a more compact form.

THEOREM 7.4. *Consider an $n \times n$ strongly-regular Hermitian matrix R such that*

$$R - FRF^* = GJG^* , \tag{7.8a}$$

where F is lower triangular, G is $n \times r$ and $J = (I_p \oplus -I_q)$, $p + q = r$, is a signature matrix. The (arbitrary) diagonal entries of F are further denoted by $\{f_i\}$.

The successive Schur complements of R with respect to its leading $i \times i$ submatrices are also structured,

$$R_i - F_i R_i F_i^* = G_i J G_i^*, \tag{7.8b}$$

where F_i is the submatrix obtained by deleting the first row and column of F_{i-1}, and G_i is an $(n-i) \times r$ generator matrix that satisfies the following recursive construction: start with $G_0 = G$, $F_0 = F$, and repeat for $i \geq 0$:

$$\begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = \left\{G_i - \Lambda_i l_i d_i^{-1} g_i\right\}\Theta_i , \tag{7.8c}$$

where Θ_i is an arbitrary $r \times r$ $J-$unitary matrix, τ_i is an arbitrary unit-modulus complex scalar ($|\tau_i| = 1$), g_i is the top row of G_i, and

$$\Lambda_i = \frac{1}{(1-\tau_i^* f_i)}(I_{n-i} - \tau_i^* F_i).$$

Moreover, the d_i and l_i are the $(0,0)$ entry and the first column of the i^{th} Schur complement R_i, respectively, and they satisfy the equations

$$d_i(1 - |f_i|^2) = g_i J g_i^* , \quad (I_{n-i} - f_i^* F_i)l_i = G_i J g_i^* . \tag{7.8d}$$

The auxiliary parameters $\{h_i, k_i\}$ were eliminated from the generator recursion (7.8c). But it is still a function of the first column of i^{th} Schur complement R_i, l_i, and which is a solution of the linear system of equations

$$(I_{n-i} - f_i^* F_i) l_i = G_i J g_i^*.$$

If no restrictions are imposed on the diagonal entries of F, then the displacement equation (7.8b) may not specify R uniquely and, consequently, the l_i in the above equation may not be uniquely defined. In other words, the recursion (7.8c) is adequate as long as the l_i and d_i can be uniquely determined from (7.8d) or from other available information. More details on this issue are provided in Section 7.3.

7.1.3. The Special Case of a Unique R. But an important special case that is of interest is when the displacement equation (7.8b) uniquely defines R. This happens when the eigenvalues of F (or equivalently its diagonal entries, since F is triangular) satisfy the condition

$$1 - f_i f_j^* \neq 0 \quad \text{for all } \ i, j.$$

In this case, we can uniquely solve for l_i and express it in the form

$$l_i = (I_{n-i} - f_i^* F_i)^{-1} G_i J g_i^*.$$

If we now substitute this expression for l_i into the generator recursion (7.8c), we obtain the following alternative update for the generator matrices [120, 171, 175].

COROLLARY 7.5. *Assume that, in Theorem 7.4, the diagonal entries of F satisfy*

$$1 - f_i f_j^* \neq 0 \quad \textit{for all } \ i, j. \tag{7.9a}$$

Then the recursion (7.8c) simplifies to the following: start with $G_0 = G$, $F_0 = F$, and repeat for $i \geq 0$:

$$\begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = \left\{ G_i + (\Phi_i - I_{n-i}) G_i \frac{J g_i^* g_i}{g_i J g_i^*} \right\} \Theta_i \ , \tag{7.9b}$$

where Φ_i is a "Blaschke" matrix of the form,

$$\Phi_i = \frac{1 - \tau_i f_i^*}{\tau_i - f_i} (F_i - f_i I_{n-i})(I_{n-i} - f_i^* F_i)^{-1} \ , \tag{7.9c}$$

Θ_i is an arbitrary $J-$unitary matrix, and τ_i is an arbitrary unit-modulus scalar. The triangular factors of R are determined by

$$l_i = (I_{n-i} - f_i^* F_i)^{-1} G_i J g_i^*, \ \ d_i = \frac{g_i J g_i^*}{1 - |f_i|^2}. \tag{7.9d}$$

Furthermore, each step of the algorithm gives rise to a first-order r-input $r-$output section with transfer matrix

$$\Theta_i(z) = \left\{ I + [B_i(z) - 1] \frac{J g_i^* g_i}{g_i J g_i^*} \right\} \Theta_i \ , \tag{7.9e}$$

where $B_i(z)$ is a Blaschke factor of the form

$$B_i(z) = \frac{z - f_i}{1 - z f_i^*} \frac{1 - \tau_i f_i^*}{\tau_i - f_i}. \tag{7.9f}$$

We may remark that the completion-of-squares argument is not absolutely essential to the derivation of the generator recursions (7.8c) or (7.9b). Indeed, starting with expression (7.3a) for the displacement of $\tilde{R}_1$, one can check that its right-hand side is equal to

$$\begin{bmatrix} \mathbf{0} \\ G_1 \end{bmatrix} J \begin{bmatrix} \mathbf{0} \\ G_1 \end{bmatrix}^* ,$$

where G_1 is the matrix obtained via (7.8c) or (7.9b), without the need to explicitly introduce the auxiliary parameters $\{h_0, k_0\}$ and then eliminate them via the expressions of Lemma 7.3. This is clearly a possible alternative to the arguments of the previous sections, but one has to somehow foresee (7.8c) or (7.9b), or any other related special case. We have chosen to present the completion-of-squares argument in order to highlight connections with other important results that arise in the study of $J-$lossless systems, such as the embedding relation (7.5b), as well as the first-order state-space representation (7.5a). In any case, the point is that the generator recursions (7.8c) or (7.9b) can be derived and motivated in several different ways, each of which has its own merits and highlights different connections and results. It can even be written or presented in many different forms, such as the general form (7.8c) (or (7.9b)), the array form, which is rederived below in (7.10d), and the earlier form (7.2a) in terms of the $\{h_i, k_i\}$ that are specified by the expressions of Lemma 7.3.

Specialization to Array Form

To show that the array form is indeed a special case of (7.9b), we first note that expression (7.9b) includes two free parameters, viz., Θ_i and τ_i. Proper choices of these quantities lead to the array interpretation of Theorem 4.2. A simple choice for τ_i is $\tau_i = (1+f_i)/(1+f_i^*)$. In this case, the expressions for Φ_i and $B_i(z)$ collapse to

$$\text{(7.10a)} \qquad \Phi_i = (F_i - f_i I_{n-i})(I_{n-i} - f_i^* F_i)^{-1} \quad \text{and} \quad B_i(z) = \frac{z - f_i}{1 - z f_i^*} .$$

A convenient choice for Θ_i is to choose it so as to reduce the top row of G_i to the special form

$$\text{(7.10b)} \qquad g_i \Theta_i = \begin{bmatrix} \mathbf{0} & \delta_i & \mathbf{0} \end{bmatrix} ,$$

with a single nonzero entry, δ_i, say in the j^{th} column. In this case, $\Theta_i(z)$ in (7.9e) and the generator recursion (7.9b) reduce to

$$\text{(7.10c)} \qquad \Theta_i(z) = \Theta_i \begin{bmatrix} I_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & B_i(z) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{r-j-1} \end{bmatrix} , \quad B_i(z) = \frac{z - f_i}{1 - z f_i^*} ,$$

$$\text{(7.10d)} \begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = \Phi_i G_i \Theta_i \begin{bmatrix} \mathbf{0}_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{r-j-1} \end{bmatrix} + G_i \Theta_i \begin{bmatrix} I_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{r-j-1} \end{bmatrix} ,$$

which is the generalized array form that we presented earlier in Theorem 4.2. But other choices are also possible and would lead to different forms for the generator recursion, such as the simple choice $\Theta_i = I$.

The previous results can be extended rather immediately to the case of non-Hermitian structured matrices as in (6.7). We shall not elaborate on this extension here, only because it is a special case of the general discussion in Section 7.4, and can thus be immediately obtained from the results in that section.

7.2. Continuous-Time Embedding Relations: Completion of Squares. We have focused so far on Toeplitz-like matrices as in (6.7) and (7.1a). We now discuss another important subclass of structured matrices, the so-called Hankel-like matrices. It will become clear, as we proceed with our discussion in this section, that the derivation of the associated Schur algorithm is considerably more immediate in this case. The reason is that our completion-of-squares argument will also lead to an embedding relation in terms of auxiliary quantities $\{h_i, k_i\}$; but it happens that this relation is very immediate to solve in the Hankel-like case, thus avoiding the route we had to take before in the Toeplitz-like case by solving for h_i and k_i as in Lemma 7.3.

But first let us exhibit several examples of Hankel and Hankel-like matrices; here we should mention that though we had introduced the concept of Hankel-like matrices independently and in fact somewhat earlier (see [114, 115, 130]), it was Heinig and Rost [101] who first made a detailed study of such structure, e.g., explicitly identifying Vandermonde and Cauchy matrices as being in this class. However, as will be illustrated soon (in Section 7.2.1), these matrices can also be regarded as being Toeplitz-like; as mentioned earlier, it is possible to transform results for displacement operators of the form $R - FRA^*$ to those for $FR + RA^*$. Nevertheless, in special cases, it may be simpler to work in one or the other domain – see, e.g., the root-distribution problem studied in Section 8.2 ahead.

7.2.1. Hankel, Vandermonde, Pick, Cauchy, Loewner Matrices. We start with a Hankel matrix $H = [h_{i+j}]_{i,j=0}^{n-1}$ and note that

$$(7.11)\ \nabla H = ZH - HZ^* = \begin{bmatrix} 0 & -h_0 & -h_1 & \dots & -h_{n-2} \\ h_0 & & & & \\ h_1 & & \text{O} & & \\ \vdots & & & & \\ h_{n-2} & & & & \end{bmatrix} \quad \text{has rank 2.}$$

We thus say that H has displacement rank 2 with respect to the displacement $ZH - HZ^*$. A minor problem here is that H cannot be recovered from its displacement ∇H, because the entries $\{h_{n-1}, \dots, h_{2n-2}\}$ do not appear in ∇H; this "difficulty" can be fixed in various ways (see, e.g., [50, 99] and also Section 7.3). One is to border H with zeros and then form the displacement, which will now have rank 4. Another method is to form the $2n \times 2n$ (triangular) Hankel matrix with top row $\{h_0, \dots, h_{2n-1}\}$; now the displacement rank will be two. Note that in both cases the generators have the same number of entries.

Several examples of Hankel-like matrices occur in applications. For instance, the so-called Loewner and Cauchy matrices arise in the study of unconstrained interpolation problems and in system theory (see, e.g., [13, 74, 194]). The entries of an $n \times n$ Cauchy-like matrix R have the form

$$R = \left[\frac{\mathbf{u}_i \mathbf{v}_j^*}{f_i - a_j^*} \right]_{i,j=0}^{n-1},$$

where $\mathbf{u}_i$ and $\mathbf{v}_j$ denote $1 \times r$ row vectors, and the $\{f_i, a_i\}$ are complex scalars that satisfy the conditions $f_i - a_j^* \neq 0$ for all i, j. The Loewner matrix is a special Cauchy matrix that corresponds to the choices $r = 2$, $\mathbf{u}_i = \begin{bmatrix} \beta_i & 1 \end{bmatrix}$, and $\mathbf{v}_i = \begin{bmatrix} 1 & -\kappa_i \end{bmatrix}$, and consequently, $\mathbf{u}_i\mathbf{v}_j^* = \beta_i - \kappa_j^*$,

$$R = \left[\frac{\beta_i - \kappa_j^*}{f_i - a_j^*}\right]_{i,j=0}^{n-1} .$$

We also note that Cauchy matrices arise from the choices $r = 1$ and $\mathbf{u}_i = 1 = \mathbf{v}_i$.

It is easy to verify that the Cauchy-like matrix R satisfies

$$FR - RA^* = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{n-1} \end{bmatrix} \begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_{n-1} \end{bmatrix}^* , \tag{7.12a}$$

where F and A are diagonal matrices,

$$F = \text{diagonal}\{f_0, \ldots, f_{n-1}\}, \quad A = \text{diagonal}\{a_0, \ldots, a_{n-1}\} .$$

We say that a Cauchy-like matrix R has displacement rank r with respect to the displacement defined by $FR - RA^*$.

Another example is the so-called Pick matrix that arises in interpolation problems for the right-half plane, for example,

$$R = \left[\frac{1 - \beta_i\beta_j^*}{f_i + f_j^*}\right]_{i,j=0}^{n-1} ,$$

where the β_i are complex scalars and the f_i are points in the right-half plane ($Re\ (f_i) > 0$). For the same diagonal F as above, we can check that

$$FR + RF^* = \begin{bmatrix} 1 & \beta_0 \\ 1 & \beta_1 \\ \vdots & \vdots \\ 1 & \beta_{n-1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & \beta_0 \\ 1 & \beta_1 \\ \vdots & \vdots \\ 1 & \beta_{n-1} \end{bmatrix}^* . \tag{7.12b}$$

A final example is that of a Vandermonde matrix, say

$$V = \begin{bmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \ldots & x_1^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \ldots & x_{n-1}^{n-1} \end{bmatrix} ,$$

which turns out to have displacement rank 1 since

$$AV - VZ^* = \begin{bmatrix} \frac{1}{x_0} & \\ \frac{1}{x_1} & \\ \vdots & O \\ \frac{1}{x_{n-1}} & \end{bmatrix} , \tag{7.12c}$$

where A is the diagonal matrix (assuming $x_i \neq 0$)

$$A = \text{diagonal}\,\{\frac{1}{x_0}, \ldots, \frac{1}{x_{n-1}}\}\,.$$

But note that we can as well write

$$V - FVZ^* = \begin{bmatrix} 1 & \\ 1 & \\ \vdots & \text{O} \\ 1 & \end{bmatrix},$$

where F is now the diagonal matrix

$$F = \text{diagonal}\,\{x_0, \ldots, x_{n-1}\}\,.$$

This shows that the Vandermonde matrix also has displacement rank 1 with respect to the structure defined by $V - FVZ^*$, which avoids the restriction that the x_i be nonzero. Similar statements can be made for the Pick and Cauchy matrices, as well, and in fact for all classes of structured matrices, a point emphasized by V. Pan in his interesting paper [156] and in his recent book with Bini [24], which contains much additional material on displacement operators. Pan proposed to improve Hilbert- and Vandermonde-type computations by reducing them to the more-studied Toeplitz/Hankel-type computations. On the other hand, Heinig [97] proposed going in the other direction in order to exploit the fact that the Cauchy structure is unaffected by row or column interchanges, so that pivoting strategies can be employed [97]. In [87, 89, 119] it was observed that partial pivoting could also be incorporated into generalized Schur algorithms for larger classes of structured matrices – see also Sec. 11.5.

7.2.2. Generalized Schur Algorithm for Hankel-Like Matrices. We now consider strongly-regular Hermitian matrices R that satisfy displacement equations of the form

$$FR + RF^* + GJG^* = \mathbf{0}\,, \tag{7.13}$$

where F is an *arbitrary* lower triangular matrix with diagonal entries $\{f_i\}_{i=0}^{n-1}$, G is an $n \times r$ generator matrix, and J is any nonsingular matrix satisfying $J^2 = I$, e.g., a signature matrix $J = (I_p \oplus -I_q)$.

A short remark is due here. Note that in defining the Hankel-displacement (7.13) we chose to write $FR + RF^*$ rather than $FR - RF^*$. One motivation for this is that Pick matrices, as in (7.12b), are structured with respect to $FR + RF^*$. Another is that (7.13) is the standard form (at least when $J = I$) of the famous Lyapunov equation for checking whether the eigenvalues of F have negative real parts. The form (7.13) is also more convenient for comparing the associated Schur algorithm with that of the Toeplitz-like case (7.1a), as will become clear further ahead. In any case, all these forms, as well as the non-Hermitian counterpart, are special cases of the general structure (7.24) that is considered in a future section.

We say that R in (7.13) is a Hankel-like matrix with respect to (F, G, J), and our purpose is to show that the Hankel-like structure is preserved under Schur complementation. That is, if R_1 is the Schur complement of r_{00} in R then R_1 is also Hankel-like. To check this, we let l_0 and g_0 denote the first column of R and the top row of G, respectively. We then conclude from (7.13) that the first column l_0 and the top left-corner element r_{00} of R obey the identities

$$Fl_0 + l_0 f_0^* + GJg_0^* = \mathbf{0} \quad , \quad d_0(f_0 + f_0^*) + g_0 J g_0^* = 0 \; . \tag{7.14}$$

Using (4.1a) and (7.13) we can prove the following lemma.

LEMMA 7.6. *Consider an $n \times n$ strongly-regular Hermitian matrix R that satisfies the displacement equation*

$$FR + RF^* + GJG^* = \mathbf{0} \; , \tag{7.15a}$$

where the diagonal entries of F are denoted by $\{f_i\}$. The first Schur complement R_1 satisfies

$$F_1 R_1 + R_1 F_1^* + G_1 J G_1^* = \mathbf{0},$$

where F_1 is the submatrix obtained after deleting the first row and column of F, and G_1 is computed from G as follows

$$\begin{bmatrix} \mathbf{0} \\ G_1 \end{bmatrix} = G - l_0 d_0^{-1} g_0 \; . \tag{7.15b}$$

Moreover, d_0 and l_0 are the $(0,0)$ entry and the first column of R, respectively, and they satisfy the equations

$$Fl_0 + l_0 f_0^* + GJg_0^* = \mathbf{0} \quad , \quad d_0(f_0 + f_0^*) + g_0 J g_0^* = 0 \; . \tag{7.15c}$$

Proof. The proof can be obtained via straightforward manipulations as follows: using (4.1a) and (7.14) we find that

$$F\tilde{R}_1 + \tilde{R}_1 F^* = -GJG^* + GJg_0^* d_0^{-1} l_0^* + l_0 d_0^{-1} g_0 J G^* + l_0 d_0^{-1} l_0^* (f_0 + f_0^*) \; .$$

Now the right-hand side of the above expression is easily seen to be a *perfect square* since we can express it as

$$-\left[G - l_0 d_0^{-1} g_0\right] J \left[G - l_0 d_0^{-1} g_0\right]^* \; ,$$

and the result is established. □

If we define the quantities $h_0 = -Jg_0^* d_0^{-1}$ and $k_0 = I$ (the identity matrix), then (7.15b) takes a form similar to the generator recursion (7.2a) in the Toeplitz-like case. More precisely, we can rewrite (7.15b) as

$$\begin{bmatrix} \mathbf{0} \\ G_1 \end{bmatrix} = l_0 h_0^* J + GJk_0^* J \; .$$

It can also be easily seen that for any other $J-$unitary choice k_0, viz., $k_0 J k_0^* = J$, and for $h_0 = -k_0 J g_0^* d_0^{-1}$, we also get

$$-\left[GJk_0^* J + l_0 h_0^* J\right] J \left[GJk_0^* J + l_0 h_0^* J\right]^* = F\tilde{R}_1 + \tilde{R}_1 F^* \; .$$

We are, therefore, led to the following generalization of Lemma 7.6.

LEMMA 7.7. *Consider the same setting as Lemma 7.6. The Schur complement* R_1 *satisfies*

$$F_1 R_1 + R_1 F_1^* + G_1 J G_1^* = \mathbf{0},$$

where

$$\begin{bmatrix} \mathbf{0} \\ G_1 \end{bmatrix} = l_0 h_0^* J + G J k_0^* J \ , \tag{7.16}$$

and h_0 *and* k_0 *are, respectively,* $r \times 1$ *and* $r \times r$ *arbitrary matrices chosen so as to satisfy, along with* $\{d_0, f_0, g_0\}$*, the embedding relations*

$$\begin{aligned} f_0 d_0 + d_0 f_0^* + g_0 J g_0^* &= 0 \\ h_0 d_0 + k_0 J g_0^* &= \mathbf{0} \\ k_0 J k_0^* &= J \ . \end{aligned}$$

The argument can now be repeated for the successive Schur complements and leads to the following theorem, which is the counterpart in the Hankel-like case of Lemma 7.2 and Theorem 7.4. [We may note here that solving the embedding relations for (h_0, k_0) in the Hankel-like case is rather trivial, thus allowing us to avoid the intermediate route of first determining h_0 and k_0 as in Lemma 7.3 in the Toeplitz-like case].

THEOREM 7.8. *Consider an* $n \times n$ *strongly-regular Hermitian matrix* R *that obeys the displacement equation*

$$FR + RF^* + GJG^* = \mathbf{0} \ , \tag{7.17a}$$

where F *is lower triangular,* G *is* $n \times r$*,* J *is an* $r \times r$ *signature matrix satisfying* $J^2 = I$*, and the diagonal entries of* F *are denoted by* $\{f_i\}$*. The Schur complements* R_i *are also structured with generator matrices* G_i*, viz.,* $F_i R_i + R_i F_i^* + G_i J G_i^* = \mathbf{0}$*, where* F_i *is the submatrix obtained after deleting the first row and column of* F_{i-1}*, and* G_i *is an* $(n-i) \times r$ *generator matrix that satisfies, along with* l_i *(the* 1^{st} *column of* R_i*), the following recursion*

$$\begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = l_i h_i^* J + G_i J k_i^* J \ , \tag{7.17b}$$

where h_i *and* k_i *are, respectively,* $r \times 1$ *and* $r \times r$ *arbitrary matrices that satisfy, along with* $\{d_i, f_i, g_i\}$*, the embedding relation*

$$\begin{aligned} f_i d_i + d_i f_i^* + g_i J g_i^* &= 0 \\ h_i d_i + k_i J g_i^* &= \mathbf{0} \\ k_i J k_i^* &= J \ . \end{aligned} \tag{7.17c}$$

Moreover, d_i *and* l_i *are the* $(0,0)$ *entry and the first column of* R_i*, respectively, and they satisfy the relations*

$$d_i(f_i + f_i^*) + g_i J g_i^* = 0, \ \ F_i l_i + l_i f_i^* + G_i J g_i^* = \mathbf{0}. \tag{7.17d}$$

A simple choice for h_i *and* k_i *is*

$$k_i = I \ \ , \ \ h_i = -J g_i^* d_i^{-1} \ ,$$

which leads to the simplified generator recursion

$$\begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = G_i - l_i d_i^{-1} g_i \,. \tag{7.18}$$

For any other choice k_i, and defining $\Theta_i = k_i^{-1}$, the generator recursion can be re-expressed as

$$\begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = \left\{ G_i - l_i d_i^{-1} g_i \right\} \Theta_i \,, \tag{7.19}$$

where Θ_i is an arbitrary $r \times r$ $J-$unitary matrix.

Note again that the auxiliary parameters $\{h_i, k_i\}$ were eliminated from the generator recursion (7.19). But it is still a function of the first column of i^{th} Schur complement R_i, which is a solution of the linear system of equations

$$(F_i + f_i^* I_{n-i}) l_i = -G_i J g_i^*.$$

If no restrictions are imposed on the diagonal entries of F, then the displacement equation (7.17a) may not specify R uniquely and, consequently, the l_i in the above equation may not be uniquely defined. In other words, the recursion (7.19) is adequate as long as the l_i and d_i can be uniquely determined from (7.17d) or from other available information. More details on this issue are provided in Section 7.3.

7.2.3. The Special Case of a Unique R. But an important special case that is of interest is when the displacement equation (7.17a) uniquely defines R. This happens when the eigenvalues of F (or equivalently its diagonal entries, since F is triangular) satisfy the condition

$$f_i + f_j^* \neq 0 \quad \text{for all} \quad i, j.$$

In this case, we can uniquely solve for l_i and express it in the form

$$l_i = -(F_i + f_i^* I_{n-i})^{-1} G_i J g_i^*.$$

If we now substitute this expression for l_i into the generator recursion (7.19), we obtain the following alternative update for the generator matrices [120, 171, 175].

COROLLARY 7.9. *Assume that, in Theorem 7.8, the diagonal entries of F satisfy*

$$f_i + f_j^* \neq 0 \quad \text{for all} \quad i, j. \tag{7.20a}$$

Then the generator recursion (7.19) reduces to the following: start with $G_0 = G$, $F_0 = F$, and repeat for $i \geq 0$:

$$\begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = \left\{ G_i + (\Phi_i - I_{n-i}) G_i \frac{J g_i^* g_i}{g_i J g_i^*} \right\} \Theta_i \ , \tag{7.20b}$$

where Φ_i is a "Blaschke" matrix of the form,

$$\Phi_i = (F_i - f_i I_{n-i})(F_i + f_i^* I_{n-i})^{-1} \ , \tag{7.20c}$$

and Θ_i is an arbitrary $J-$unitary matrix. The triangular factors are given by

$$l_i = -(F_i + f_i^* I_{n-i})^{-1} G_i J g_i^*, \;\; d_i = -\frac{g_i J g_i^*}{f_i + f_i^*}. \tag{7.20d}$$

Specialization to Array Form

Assume, without loss of generality, that the signature matrix J has the form $J = (I_p \oplus -I_q)$. If we now choose Θ_i so as to reduce g_i to the form

$$g_i\Theta_i = \begin{bmatrix} \mathbf{0} & \delta_i & \mathbf{0} \end{bmatrix} , \tag{7.21}$$

with a single nonzero entry, δ_i, say in the j^{th} column, it is then easy to verify that the above generator recursion reduces to the following array form

$$\begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = \Phi_i G_i \Theta_i \begin{bmatrix} \mathbf{0}_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{r-j-1} \end{bmatrix} + G_i\Theta_i \begin{bmatrix} I_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{r-j-1} \end{bmatrix} .$$

We may add that in several instances, the simple choices $\Theta_i = I$ or $\Theta_i = J$ (rather than as hyperbolic rotations to achieve (7.21)) are also of interest and often lead to alternative convenient forms for the generator recursion – see, e.g., the calculations in Section 8.2.1.

7.2.4. Elementary Continuous-Time State-Space Sections. It is again interesting to note how the so-called embedding (dilation) relation (7.17c) arises naturally in our framework. It plays a fundamental role in characterizing $J-$lossless transfer functions over the left-half plane. Indeed, if we define the first-order section

$$\Theta_i^{-1}(s) = k_i + h_i(s - f_i)^{-1} g_i = \left[I + h_i(s - f_i)^{-1} g_i \right] k_i^{-1} ,$$

and use the proper choice for $\Theta_i = k_i^{-1}$, as in (7.21), we get

$$\Theta_i(s) = \Theta_i \begin{bmatrix} I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{s-f_i}{s+f_i^*} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I \end{bmatrix} .$$

This transfer function clearly satisfies

$$\Theta_i(s) J \Theta_i^*(s) = J \text{ on } \operatorname{Re}(s) = 0 .$$

If we further assume that $\operatorname{Re}(f_i) < 0$, it then follows that: (i) $\Theta_i(s)$ is analytic in the left-half plane ($\operatorname{Re}(s) < 0$), (ii) $J-$unitary on the imaginary axis, $\Theta_i(s)J\Theta_i^*(s) = J$ on $\operatorname{Re}(s) = 0$, and (iii) $\Theta_i(s)J\Theta_i^*(s) < J$ in $\operatorname{Re}(s) < 0$. We thus say that $\Theta_i(s)$ is a $J-$lossless transfer function.

7.2.5. A Note on the Non-Hermitian Case. The same arguments extend to the non-Hermitian case,

$$FR + RA^* + GJB^* = \mathbf{0} , \tag{7.22}$$

where F and A are lower triangular matrices with diagonal entries $\{f_i, a_i\}_{i=0}^{n-1}$, G and B are an $n \times r$ generator matrices, and J is a signature matrix. Such matrices admit a triangular factorization of the form $R = LD^{-1}U$, where L is lower triangular and U is upper triangular with identical diagonal entries, and which are equal to those of D. In what follows, we denote the (nonzero parts of the) columns and rows of L

and U by $\{l_i, u_i\}$, respectively. The arguments of the previous section extend rather immediately to the non-Hermitian case, thus leading to the following result.

THEOREM 7.10. *Consider an $n \times n$ strongly-regular non-Hermitian matrix R that satisfies*

$$FR + RA^* + GJB^* = \mathbf{0}\ , \tag{7.23a}$$

where the diagonal entries of the lower triangular matrices F and A are arbitrary and denoted by $\{f_i, a_i\}$, respectively. Then the successive Schur complements R_i satisfy

$$F_iR_i + R_iA_i^* + G_iJB_i^* = \mathbf{0}, \tag{7.23b}$$

where F_i and A_i are the submatrices obtained after deleting the first row and column of F_{i-1} and A_{i-1}, respectively, and G_i and B_i are $(n-i)\times r$ generator matrices that satisfy, along with l_i and u_i (the first column and row of R_i), the following recursions

$$\begin{aligned} \begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} &= l_i h_i^* J + G_i J k_i^* J\ , \\ \begin{bmatrix} \mathbf{0} \\ B_{i+1} \end{bmatrix} &= u_i^* c_i^* J + B_i J s_i^* J\ , \end{aligned} \tag{7.23c}$$

where $\{h_i, c_i\}$ and $\{k_i, s_i\}$ are, respectively, $r\times 1$, $r\times 1$, $r\times r$, and $r\times r$ arbitrary matrices that satisfy, along with $\{d_i, f_i, a_i, g_i, b_i\}$, the embedding relations

$$\begin{aligned} f_id_i + d_ia_i^* + g_iJb_i^* &= 0 \\ h_id_i^* + k_iJg_i^* &= \mathbf{0} \\ c_id_i + s_iJb_i^* &= \mathbf{0} \\ k_iJs_i^* &= J\ , \end{aligned} \tag{7.23d}$$

A simple choice for $\{h_i, c_i\}$ and $\{k_i, s_i\}$ is

$$k_i = s_i = I\ \ ,\ \ h_i = -Jg_i^*d_i^{-*}\ ,\ \ c_i = -Jb_i^*d_i^{-1}\ ,$$

which leads to the simplified generator recursions

$$\begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = G_i - l_id_i^{-1}g_i\ ,\ \ \begin{bmatrix} \mathbf{0} \\ B_{i+1} \end{bmatrix} = B_i - u_i^*d_i^{-*}b_i\ . \tag{7.23e}$$

Moreover, $d_i = (-g_iJb_i^*)/(f_i + a_i^*)$,

$$l_i = -(F_i + a_i^*I_{n-i})^{-1}G_iJb_i^*\ ,\ \ u_i = -g_iJB_i^*(A_i^* + f_iI_{n-i})^{-1}\ .$$

For any other choices k_i and s_i, and defining $\Theta_i = k_i^{-1}$ and $\Gamma_i = s_i^{-1}$, the generator recursions can be re-expressed in the form

$$\begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = \left\{G_i - l_id_i^{-1}g_i\right\}\Theta_i\ ,\ \ \begin{bmatrix} \mathbf{0} \\ B_{i+1} \end{bmatrix} = \left\{B_i - u_i^*d_i^{-*}b_i\right\}\Gamma_i\ , \tag{7.23f}$$

where $\Theta_iJ\Gamma_i^* = J$.

These recursions, as well as all the recursions presented so far in the paper, are special cases of a general algorithm for matrices with generalized displacement structure introduced by Kailath and Sayed [120, 171, 175], as detailed in Section 7.4.

7.3. More on Nonunique R. We mentioned earlier that the generator recursion (7.19) is adequate as long as the l_i and d_i can be uniquely determined from (7.17d) or from other available information. A sufficient condition is clearly to assume that the diagonal entries f_i satisfy $(f_i + f_j^*) \neq 0$ for all i, j. This guarantees that we can uniquely solve for l_i and d_i and leads to the generator recursion (7.20b).

While the uniqueness condition $(f_i + f_j^*) \neq 0$ is met in many applications, there are various instances where it is not; as mentioned above, we now will need additional information to carry on the recursion. No useful universal prescription seems to be available, but in important cases special methods can be used to address the problems arising from nonuniqueness. We illustrate this by considering two examples.

Our first example follows [50] in studying the Hankel matrix in (7.11). 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. More fundamentally, the issue is that the columns of the triangular factors of H, satisfy linear equations of the form $Z_i l_i = G_i J g_i^*$, with a singular coefficient matrix Z_i. The elements in the nullspace of Z_i are column vectors of the form $\begin{bmatrix} 0 & \ldots & 0 & \alpha \end{bmatrix}^{\mathbf{T}}$, with a single nonzero entry in the last position and $(i-1)$ leading zeros (Z_i stands for the $(i \times i)$ shift matrix).

Hence, the nonuniqueness only affects the last entry of l_i. This suggests the following method around this difficulty: border H by a row and column of zeros, say

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

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

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

Now, the nullspace of Z_{n+1} is the set of all $(n+1) \times 1$ column vectors of the form

$$\begin{bmatrix} 0 & \ldots & 0 & \alpha \end{bmatrix}^{\mathbf{T}}, \quad \text{for arbitrary scalar } \alpha,$$

and the columns of the triangular factors of M are orthogonal to the above nullspace, since their last entries are zero. We can thus uniquely determine the columns of the triangular factor of M, and by deleting their last entries we obtain those of the triangular factors of H.

Another way out of the nonuniqueness problem is to simply assume that we are also given the entries that are not uniquely specified by the displacement equation. While this assumption is not always natural in matrix factorization problems (e.g., in the Hankel case), it happens in certain interpolation problems that this extra information is available in the form of what are called coupling numbers (see, e.g., [20, 91]). To illustrate this, we consider a 3×3 Cauchy-like strongly-regular matrix $R = [r_{ij}]_{i,j=0}^2$ that satisfies the displacement equation

$$FR - RA^* = GJB^*,$$

with $F = \text{diagonal}\{f_0, f_1, f_2\}$ and $A = \text{diagonal}\{a_0, a_1, a_2\}$. Assume $a_2^* = f_1$. It then follows that the r_{12} entry of R can not be recovered from the displacement equation.

Therefore, the matrix R is completely described by the generator matrices $\{G, B\}$, the displacement matrices $\{F, A\}$, and by its $(1,2)$ entry, r_{12}. This entry is assumed to be known and is referred to as a coupling number, $\rho_{12}^{(0)} = r_{12}$. The generator recursion for R can now be updated as follows:

(i) Recover from the generator matrices and from the coupling data the first column l_0 and the first row u_0 of R;

(ii) Compute G_1 and B_1 using (7.23f);

(iii) Update the coupling numbers via a standard Schur reduction procedure, viz., $\rho_{12}^{(1)} = \rho_{12}^{(0)} - r_{10}r_{00}^{-1}r_{02}$.

These steps can now be repeated for the higher-order Schur complements. We may remark that, unlike the bordering technique for the Hankel example, which successfully exploits the underlying Hankel structure, the third step in the above coupling procedure updates the coupling numbers via a standard Schur reduction procedure. More study of the nonunique case is possible.

7.4. Generalized Schur Algorithm for Generalized Displacement Structure. To clarify our claim that the varied algorithms that were presented in earlier sections are special cases of a more general definition of displacement structure, we now consider matrices that are structured with respect to the following definition [120, 171, 175]

$$\Omega R \Delta^* - FRA^* = GJB^* \ , \tag{7.24}$$

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 $n \times r$ generator matrices, and J is an $r \times r$ signature matrix. We further assume here that, for each i, at least one of the following conditions is satisfied

$$\omega_i \delta_i^* \neq 0 \quad \text{or} \quad f_i a_i^* \neq 0 \quad \text{or} \quad f_i \delta_i^* \neq 0 \quad \text{or} \quad \omega_i a_i^* \neq 0. \tag{7.25a}$$

This means that, for each i, ω_i and f_i are not zero simultaneously, and also a_i and δ_i are not zero simultaneously. These conditions are weaker than requiring (7.24) to have a unique solution R, which would be the case had we instead required the diagonal entries $\{\omega_i, \delta_i, f_i, a_i\}$ to satisfy the condition

$$\omega_j \delta_i^* - f_j a_i^* \neq 0 \quad \text{for all} \quad i, j. \tag{7.25b}$$

Special cases of (7.24) include Hankel-like matrices such as,

$$\Omega R + RA^* = GJB^* \ ,$$

which corresponds to the choice $\Delta = -F = I$ and, therefore, satisfies (7.25a) since $f_i \delta_i^* = -1 \neq 0$. Expression (7.24) also includes Toeplitz-like matrices such as,

$$R - FRA^* = GJB^* \ ,$$

which corresponds to $\Omega = \Delta = I$; thus satisfying (7.25a) since $\omega_i \delta_i^* = 1 \neq 0$. We now proceed to verify that the generalized structure is also preserved under Schur complementation. The argument depends on which condition in (7.25a) is satisfied. So we proceed gradually, and the final result is stated in Theorem 7.14.

To begin with, assume $\omega_0 \delta_0^* \neq 0$, which is in accordance with (7.25a). (Due to the symmetry of equation (7.24), the arguments that follow are equally applicable if we instead had $f_0 a_0^* \neq 0$). It follows from the displacement equation (7.24) that the $(0,0)$ entry of R, its first column l_0, and its first row u_0, satisfy the following relations

$$\begin{aligned} \Omega l_0 \delta_0^* &= F l_0 a_0^* + G J b_0^*, \\ \omega_0 u_0 \Delta^* &= f_0 u_0 A^* + g_0 J B^*, \\ d_0(\omega_0 \delta_0^* - f_0 a_0^*) &= g_0 J b_0^*, \end{aligned} \tag{7.26}$$

where g_0 and b_0 denote the first rows of G and B, respectively. Let $\{\Omega_1, \Delta_1, F_1, A_1\}$ denote the submatrices obtained after deleting the first row and column of $\{\Omega, \Delta, F, A\}$, respectively. The following result is the immediate extension of Lemma 7.1.

LEMMA 7.11. *Consider an $n \times n$ strongly regular matrix that satisfies (7.24) and assume $\omega_0 \neq 0$ and $\delta_0 \neq 0$. The first Schur complement R_1 satisfies*

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

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

$$\begin{aligned} \begin{bmatrix} \mathbf{0} \\ G_1 \end{bmatrix} &= F l_0 c_0^* J + G J s_0^* J \ , \\ \begin{bmatrix} \mathbf{0} \\ B_1 \end{bmatrix} &= A u_0^* h_0^* J + B J k_0^* J \ , \end{aligned} \tag{7.27a}$$

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

$$\begin{bmatrix} f_0 & g_0 \\ h_0 & k_0 \end{bmatrix} \begin{bmatrix} d_0 & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix} \begin{bmatrix} a_0 & b_0 \\ c_0 & s_0 \end{bmatrix}^* = \begin{bmatrix} \omega_0 d_0 \delta_0^* & \mathbf{0} \\ \mathbf{0} & J \end{bmatrix}. \tag{7.27b}$$

Also, d_0, l_0, and u_0 are the $(0,0)$ entry, the first column and the first row of R, respectively, and they satisfy the equations

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

Proof. The proof can be obtained via direct manipulations that are similar to the proof of Lemma 7.1. We write

$$\begin{aligned} \Omega \tilde{R}_1 \Delta^* - F \tilde{R}_1 A^* &= G J B^* - \Omega l_0 d_0^{-1} u_0 \Delta^* + F l_0 d_0^{-1} u_0 A^*, \\ &= G J \left\{ J - \frac{b_0^* g_0}{\omega_0 d_0 \delta_0^*} \right\} J B^* - \\ &\quad G J \ \frac{f_0 b_0^*}{\omega_0 d_0 \delta_0^*} \ u_0^* A^* - F l_0 \ \frac{a_0^* g_0}{\omega_0 d_0 \delta_0^*} \ J B^* + \\ &\quad F l_0 \ \frac{g_0 J b_0^*}{\omega_0 d_0^2 \delta_0^*} \ u_0^* A^* \ , \end{aligned} \tag{7.28}$$

where we have replaced Ωl_0 and $u_0 \Delta^*$ by the relations

$$\begin{aligned} \Omega l_0 &= \frac{1}{\delta_0^*} \left[F l_0 a_0^* + G J b_0^* \right], \\ u_0 \Delta^* &= \frac{1}{\omega_0} \left[f_0 u_0 A^* + g_0 J B^* \right]. \end{aligned}$$

Here we invoked the nonsingularity of δ_0 and ω_0. [If we had instead $f_0 \neq 0$ and $a_0 \neq 0$, then we would have replaced Fl_0 and $u_0 A^*$ by $(a_0^{-*}[\Omega l_0 \delta_0^* - GJb_0^*])$ and $(f_0^{-1}[\omega_0 u_0 \Delta^* - g_0 JB^*])$, respectively; thus leading to a similar result but with an appropriately modified embedding relation. In any case, the general result is stated in the next lemma.]. It follows, as in the proof of Lemma 7.1, that the right-hand side of (7.28) can be put into a perfect-square because of (7.27b). □

We may now proceed as in the Toeplitz-like case and determine expressions for $\{h_0, k_0, c_0, s_0\}$ so as to satisfy the embedding relation (7.27b) (see, e.g., [171, 175]), thus leading to the following extension of Theorem 7.4.

LEMMA 7.12. *Consider an $n \times n$ strongly regular matrix R as in (7.24) and assume $\omega_0 \neq 0$ and $\delta_0 \neq 0$. The first Schur complement R_1 satisfies*

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

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

$$\begin{bmatrix} \mathbf{0} \\ G_1 \end{bmatrix} = \left\{G - \Lambda_0 l_0 d_0^{-1} g_0\right\} \Theta_0 \ ,$$

$$\begin{bmatrix} \mathbf{0} \\ B_1 \end{bmatrix} = \left\{B - \Sigma_0 u_0^* d_0^{-*} b_0\right\} \Gamma_0 \ , \tag{7.29}$$

where $\Theta_0 J \Gamma_0^* = J$,

$$\Lambda_0 = \frac{1}{(\omega_0 - \tau_0^* f_0)} (\Omega - \tau_0^* F), \quad \Sigma_0 = \frac{1}{(\delta_0 - \nu_0^* a_0)} (\Delta - \nu_0^* A),$$

with $\tau_0 \nu_0^* = 1$ and where d_0, l_0, and u_0 are the $(0,0)$ entry, the first column and the first row of R, respectively, and they satisfy the equations

$$\begin{aligned} \Omega l_0 \delta_0^* &= F l_0 a_0^* + GJb_0^*, \\ \omega_0 u_0 \Delta^* &= f_0 u_0 A^* + g_0 JB^*, \\ d_0(\omega_0 \delta_0^* - f_0 a_0^*) &= g_0 J b_0^*. \end{aligned}$$

By symmetry, the recursions are also applicable if we instead had $f_0 \neq 0$ and $a_0 \neq 0$.

Note that it is always possible to choose a τ_0 so as to guarantee $(\omega_0 - \tau_0^* f_0) \neq 0$. This is due to our earlier assumption that ω_0 and f_0 are not zero simultaneously. A similar remark holds for the term $(\delta_0 - \nu_0^* a_0)$ that appears in the expression for Σ_0.

But what if we had $f_0 \delta_0^* \neq 0$? This is again in accordance with (7.25a) and, by symmetry, the arguments that follow are equally applicable if we alternatively had $\omega_0 a_0^* \neq 0$. In these cases, it is convenient to follow the continuous-time embedding technique of the previous section especially because, as we remarked earlier, the corresponding derivation is rather immediate. This leads to the following generalization of Lemma 7.6.

LEMMA 7.13. *Consider an $n \times n$ strongly regular matrix R as in (7.24) and assume $f_0 \neq 0$ and $\delta_0 \neq 0$. The first Schur complement R_1 satisfies*

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

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

$$\begin{aligned}\begin{bmatrix}\mathbf{0}\\ G_1\end{bmatrix} &= \left\{G-\frac{1}{f_0}\,Fl_0d_0^{-1}g_0\right\}\Theta_0,\\ \begin{bmatrix}\mathbf{0}\\ B_1\end{bmatrix} &= \left\{B-\frac{1}{\delta_0}\,\Delta u_0^*d_0^{-*}b_0\right\}\Gamma_0,\end{aligned}\tag{7.30}$$

where $\Theta_0 J\Gamma_0^* = J$. If we instead had $\omega_0 \neq 0$ and $a_0 \neq 0$ then the above recursions get replaced by

$$\begin{aligned}\begin{bmatrix}\mathbf{0}\\ G_1\end{bmatrix} &= \left\{G-\frac{1}{\omega_0}\,\Omega l_0d_0^{-1}g_0\right\}\Theta_0,\\ \begin{bmatrix}\mathbf{0}\\ B_1\end{bmatrix} &= \left\{B-\frac{1}{a_0}\,Au_0^*d_0^{-*}b_0\right\}\Gamma_0.\end{aligned}\tag{7.31}$$

Again, d_0, l_0, and u_0 are the $(0,0)$ entry, the first column and the first row of R, respectively, and they satisfy the equations

$$\begin{aligned}\Omega l_0\delta_0^* &= Fl_0a_0^* + GJb_0^*,\\ \omega_0u_0\Delta^* &= f_0u_0A^* + g_0JB^*,\\ d_0(\omega_0\delta_0^* - f_0a_0^*) &= g_0Jb_0^*.\end{aligned}$$

Proof. The proof is similar to that of Lemma 7.6. We prove it for the case $f_0 \neq 0$ and $\delta_0 \neq 0$. So we write

$$\begin{aligned}\Omega\tilde{R}_1\Delta^* - F\tilde{R}_1A^* &= GJB^* - \Omega l_0d_0^{-1}u_0\Delta^* + Fl_0d_0^{-1}u_0A^*,\\ &= GJB^* - \frac{1}{d_0\delta_0^*}GJb_0^*u_0\Delta^* - \frac{1}{d_0f_0}Fl_0g_0JB^* + Fl_0\frac{g_0Jb_0^*}{d_0^2f_0\delta_0^*}u_0\Delta^*\ ,\end{aligned}$$

which can be put into the perfect square form

$$\left[G-\frac{1}{f_0}\,Fl_0d_0^{-1}g_0\right]J\left[B-\frac{1}{\delta_0}\,\Delta u_0^*d_0^{-*}b_0\right]^*\ .$$

□

The previous results clearly extend to successive Schur complements, thus leading to the following general theorem.

THEOREM 7.14. *Consider an $n\times n$ strongly regular matrix R as in (7.24) and assume that, for every i, at least one of the following conditions holds:*

$$\omega_i\delta_i^* \neq 0 \quad or \quad f_ia_i^* \neq 0 \quad or \quad f_i\delta_i^* \neq 0 \quad or \quad \omega_ia_i^* \neq 0.$$

Then, the successive Schur complements of R satisfy

$$\Omega_iR_i\Delta_i^* - F_iR_iA_i^* = G_iJB_i^*,$$

where $\{\Omega_i, \Delta_i, F_i, A_i\}$ are the submatrices obtained after deleting the first row and column of the corresponding $\{\Omega_{i-1}, \Delta_{i-1}, F_{i-1}, A_{i-1}\}$, and G_i and B_i are $(n-i)\times r$ generator matrices that can be recursively constructed as follows: start with $\Omega_0 = \Omega, \Delta_0 = \Delta$, $F_0 = F$, $A_0 = A$, $G_0 = G$, and $B_0 = B$, and repeat for $i \geq 0$:

1. *If* $\omega_i\delta_i^* \neq 0$ *or* $f_i a_i^* \neq 0$ *then*

$$\begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = \left\{G_i - \Lambda_i l_i d_i^{-1} g_i\right\}\Theta_i \,,$$
$$\begin{bmatrix} \mathbf{0} \\ B_{i+1} \end{bmatrix} = \left\{B_i - \Sigma_i u_i^* d_i^{-*} b_i\right\}\Gamma_i \,, \quad \Theta_i J \Gamma_i^* = J,$$

$$\Lambda_i = \frac{1}{(\omega_i - \tau_i^* f_i)}\,(\Omega_i - \tau_i^* F_i), \quad \Sigma_i = \frac{1}{(\delta_i - \nu_i^* a_i)}\,(\Delta_i - \nu_i^* A_i), \quad \tau_i \nu_i^* = 1.$$

2. *If* $f_i\delta_i^* \neq 0$ *then*

$$\begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = \left\{G_i - \frac{1}{f_i}\,F_i l_i d_i^{-1} g_i\right\}\Theta_i,$$
$$\begin{bmatrix} \mathbf{0} \\ B_{i+1} \end{bmatrix} = \left\{B_i - \frac{1}{\delta_i}\,\Delta_i u_i^* d_i^{-*} b_i\right\}\Gamma_i, \quad \Theta_i J \Gamma_i^* = J.$$

3. *If* $\omega_i a_i \neq 0$ *then*

$$\begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = \left\{G_i - \frac{1}{\omega_i}\,\Omega_i l_i d_i^{-1} g_i\right\}\Theta_i,$$
$$\begin{bmatrix} \mathbf{0} \\ B_{i+1} \end{bmatrix} = \left\{B_i - \frac{1}{a_i}\,A_i u_i^* d_i^{-*} b_i\right\}\Gamma_i, \quad \Theta_i J \Gamma_i^* = J.$$

Also, g_i *and* b_i *are the top rows of* G_i *and* B_i*, respectively, and* $d_i, l_i,$ *and* u_i *are the* $(0,0)$ *entry, the first column, and the first row of* R_i*, respectively, and they satisfy the equations*

$$\begin{aligned} \Omega_i l_i \delta_i^* &= F_i l_i a_i^* + G_i J b_i^*, \\ \omega_i u_i \Delta_i^* &= f_i u_i A_i^* + g_i J B_i^*, \\ d_i(\omega_i\delta_i^* - f_i a_i^*) &= g_i J b_i^*. \end{aligned} \tag{7.32}$$

7.4.1. The Special Case of a Unique R. The generator recursions in the above theorem are adequate as long as d_i, l_i, and u_i can be uniquely determined from (7.32) or from other available information. If it is further assumed that the displacement equation (7.24) has a unique solution R, which happens when

$$\omega_j\delta_i^* - f_j a_i^* \neq 0 \quad \text{for all } \ i, j, \tag{7.33}$$

then we can solve explicitly for d_i, l_i and u_i, and it also allows us to conclude that we either have $\omega_j\delta_i^* \neq 0$ or $f_j a_i^* \neq 0$. This leads to the following result.

COROLLARY 7.15. *Consider an* $n \times n$ *strongly regular matrix* R *with generalized displacement structure*

$$\Omega R \Delta^* - F R A^* = G J B^* \,,$$

where $\Omega, \Delta, F,$ *and* A *are* $n \times n$ *lower triangular matrices whose diagonal entries are 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* $n \times r$ *generator matrices and* J *is an* $r \times r$ *signature matrix. It is further assumed that*

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

Then the successive Schur complements of R satisfy

$$\Omega_i R_i \Delta_i^* - F_i R_i A_i^* = G_i J B_i^*,$$

where $\{\Omega_i, \Delta_i, F_i, A_i\}$ are the submatrices obtained after deleting the first row and column of the corresponding $\{\Omega_{i-1}, \Delta_{i-1}, F_{i-1}, A_{i-1}\}$, and G_i and B_i are $(n-i)\times r$ generator matrices that satisfy the following recursions: start with $G_0 = G, B_0 = B$, $F_0 = F, A_0 = A$, $\Omega_0 = \Omega$ and $\Delta_0 = \Delta$ and repeat for $i \geq 0$:

(7.34a)
$$\begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = \left[G_i + (\Phi_i - I_{n-i})G_i \frac{Jb_i^* g_i}{g_i J b_i^*}\right]\Theta_i \ ,$$
$$\begin{bmatrix} \mathbf{0} \\ B_{i+1} \end{bmatrix} = \left[B_i + (\Psi_i - I_{n-i})B_i \frac{Jg_i^* b_i}{b_i J g_i^*}\right]\Gamma_i \ ,$$

where Θ_i and Γ_i are arbitrary matrices that satisfy $\Theta_i J \Gamma_i^* = J$,

(7.34b)
$$\Phi_i = \left(\frac{\delta_i^* - \nu_i a_i^*}{\nu_i \omega_i - f_i}\right)(\omega_i F_i - f_i \Omega_i)(\delta_i^* \Omega_i - a_i^* F_i)^{-1} \ ,$$
$$\Psi_i = \left(\frac{\omega_i^* - \tau_i f_i^*}{\tau_i \delta_i - a_i}\right)(\delta_i A_i - a_i \Delta_i)(\omega_i^* \Delta_i - f_i^* A_i)^{-1} \ ,$$

and ν_i and τ_i are complex scalars that satisfy $\nu_i \tau_i^* = 1$. The triangular factors are given by

$$l_i = (\Omega_i \delta_i^* - F_i a_i^*)^{-1} G_i J b_i^*,$$
$$u_i = g_i J B_i^* (\omega_i \Delta_i^* - f_i A_i^*)^{-1},$$
$$d_i = \frac{g_i J b_i^*}{\omega_i \delta_i^* - f_i a_i^*}.$$

7.4.2. Specialization to Array Form. The recursions can also be rewritten in array form, under the uniqueness assumption (7.33), by properly choosing the parameters (τ_i, ν_i) and the rotation matrices (Θ_i, Γ_i). We assume $J = (I_p \oplus -I_q)$, set

(7.35)
$$\nu_i = \frac{f_i + \delta_i^*}{a_i^* + \omega_i}, \quad \tau_i = \frac{a_i + \omega_i^*}{\delta_i + f_i^*},$$

and choose the rotation matrices so as to reduce the rows rows g_i and b_i to the forms

$$g_i \Theta_i = \begin{bmatrix} \mathbf{0} & \bar{x}_i & \mathbf{0} \end{bmatrix} \quad \text{and} \quad b_i \Gamma_i = \begin{bmatrix} \mathbf{0} & \bar{y}_i & \mathbf{0} \end{bmatrix} \ ,$$

respectively, where the nonzero entries $\bar{x}_i$ and $\bar{y}_i$ are in the same column position, say the j^{th} position. This leads to the following result.

COROLLARY 7.16. *Consider an $n \times n$ strongly regular matrix R with generalized displacement structure*

(7.36a)
$$\Omega R \Delta^* - F R A^* = G J B^* \ ,$$

where Ω, Δ, F, and A are $n \times n$ lower triangular matrices whose diagonal entries are 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 $n \times r$ generator matrices, and J is an $r \times r$ signature matrix. It is further assumed that

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

The successive Schur complements of R are also structured, viz.,

$$\Omega_i R_i \Delta_i^* - F_i R_i A_i^* = G_i J B_i^*, \tag{7.36b}$$

where $\{\Omega_i, \Delta_i, F_i, A_i\}$ are the submatrices obtained after deleting the first row and column of the corresponding $\{\Omega_{i-1}, \Delta_{i-1}, F_{i-1}, A_{i-1}\}$, and G_i and B_i are $(n-i)\times r$ generator matrices that satisfy the following recursions: start with $G_0 = G, B_0 = B$, $F_0 = F, A_0 = A$, $\Omega_0 = \Omega$ and $\Delta_0 = \Delta$ and repeat for $i \geq 0$:

1. *At step i we have G_i, B_i, F_i, A_i Ω_i, Δ_i. Let g_i and b_i denote the top rows of G_i and B_i, respectively;*
2. *Choose rotation matrices Θ_i and Γ_i that satisfy $\Theta_i J \Gamma_i^* = J$ and such that the rows g_i and b_i are reduced to the forms*

$$g_i \Theta_i = \begin{bmatrix} \mathbf{0} & \bar{x}_i & \mathbf{0} \end{bmatrix} \quad and \quad b_i \Gamma_i = \begin{bmatrix} \mathbf{0} & \bar{y}_i & \mathbf{0} \end{bmatrix} , \tag{7.36c}$$

respectively, where the nonzero entries $\bar{x}_i$ and $\bar{y}_i$ are in the same column position, say the j^{th} position;

3. *The generators G_{i+1} and B_{i+1} are then given by*

$$\begin{aligned} \begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} &= \Phi_i G_i \Theta_i \begin{bmatrix} \mathbf{0}_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{r-j-1} \end{bmatrix} + G_i \Theta_i \begin{bmatrix} I_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{r-j-1} \end{bmatrix} , \\ \begin{bmatrix} \mathbf{0} \\ B_{i+1} \end{bmatrix} &= \Psi_i B_i \Gamma_i \begin{bmatrix} \mathbf{0}_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{r-j-1} \end{bmatrix} + B_i \Gamma_i \begin{bmatrix} I_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{r-j-1} \end{bmatrix} , \end{aligned} \tag{7.36d}$$

where

$$\Phi_i = (\omega_i F_i - f_i \Omega_i)(\delta_i^* \Omega_i - a_i^* F_i)^{-1} ,$$
$$\Psi_i = (\delta_i A_i - a_i \Delta_i)(\omega_i^* \Delta_i - f_i^* A_i)^{-1} , \quad .$$

4. *The triangular factors are given by $d_i = (\bar{x}_i J_{jj} \bar{y}_i^*)/(\omega_i \delta_i^* - f_i a_i^*)$,*

$$l_i = (\Omega_i \delta_i^* - a_i^* F_i)^{-1} G_i \Theta_i J \begin{bmatrix} \mathbf{0} \\ \bar{y}_i^* \\ \mathbf{0} \end{bmatrix}, \quad u_i = \begin{bmatrix} \mathbf{0} & \bar{x}_i & \mathbf{0} \end{bmatrix} J \Gamma_i^* B_i^* (\omega_i \Delta_i^* - f_i A_i^*)^{-1}.$$

5. *Each step also gives rise to two $r-$input $r-$output first-order sections of the form*

$$B_{\Theta,i}(z) = \Theta_i \begin{bmatrix} I_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{\omega_i z - f_i}{\delta_i^* - z a_i^*} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{r-j-1} \end{bmatrix} , \quad B_{\Gamma,i}(z) = \Gamma_i \begin{bmatrix} I_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{\delta_i z - a_i}{\omega_i^* - z f_i^*} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{r-j-1} \end{bmatrix} .$$

7.4.3. Specialization to Hankel-Like Structure. The recursion (7.34a) also includes the algorithms of earlier sections as special cases. To demonstrate this, let us show that the recursions (7.34a) include, for example, the Hankel-type recursions (7.23f). For this purpose, we consider matrices that satisfy (*cf.* (7.22))

$$\Omega R + R A^* + G \bar{J} B^* = \mathbf{0}, \quad \bar{J} \text{ is a signature matrix,}$$

which is clearly a special case of (7.24) with

$$\Delta = I, \quad F = -I, \quad J = -\bar{J}.$$

To specialize the recursions (7.34a) to this case, we choose $\Theta_i = \Gamma_i = I$ and (ν_i, τ_i) as in (7.35). This leads to

$$\Phi_i = (\omega_i F_i - f_i \Omega_i)(\delta_i^* \Omega_i - a_i^* F_i)^{-1}, \quad \Psi_i = (\delta_i A_i - a_i \Delta_i)(\omega_i^* \Delta_i - f_i^* A_i)^{-1},$$

and the recursion for G_{i+1} now becomes (here we want to re-express the terms as a function of l_i)

$$\begin{aligned} \begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} &= G_i + \underbrace{\Phi_i G_i J b_i^*}_{(\omega_i F_i - f_i \Omega_i) l_i} \frac{g_i}{g_i J b_i^*} - \underbrace{G_i J b_i^*}_{(\delta_i^* \Omega_i - a_i^* F_i) l_i} \frac{g_i}{g_i J b_i^*}, \\ &= G_i + (-\omega_i I_{n-i} + \Omega_i - \Omega_i - a_i^* I_{n-i}) \frac{1}{g_i J b_i^*} l_i g_i, \\ &= G_i + \frac{\omega_i + a_i^*}{g_i \bar{J} b_i^*} l_i g_i = G_i - l_i d_i^{-1} g_i. \end{aligned}$$

Likewise for B_{i+1}.

Many other illustrations can be given of how, not surprisingly, the above general recursions, which were first presented in [120], include various algorithms derived in the literature for special choices of $\{\Omega, \Delta, F, A\}$. Here, however, we would like to describe an alternative function-theoretic formulation.

8. The Generating Function Approach. The generalizations studied so far of Schur's original recursion (3.1) (or (3.6a)) were obtained via matrix-based arguments, and one might wonder at this point whether a similar extension is possible in the function domain. The answer is (essentially) affirmative and it takes us to the earlier work of Lev-Ari and Kailath [130, 133, 134] on the generating function approach to structured matrices. In this framework, a matrix R is described in terms of a bivariate function $R(z, w)$: if R is structured then $R(z, w)$ will be a structured bivariate function as clarified below. More important perhaps is the fact that the bivariate functions associated with the successive Schur complements of R will also inherit the same structure as $R(z, w)$. The final recursions, in function language, hint to connections with results in complex analysis and to several applications. In particular, we elaborate further ahead on a connection with Bezoutians and stability tests.

We shall, for simplicity, limit ourselves in this section to Hermitian strongly regular matrices, although the approach is readily applicable to non-Hermitian matrices as well [183]. Our goal will be triangular factorization, which is a nested operation. Therefore we shall assume, and without loss of generality, that all matrices R are extended to be semi-infinite.

The generating function of a semi-infinite matrix $R = [r_{mj}]_{m,j=0}^{\infty}$ is the bivariate function $R(z, w)$ defined by

$$R(z, w) = \begin{bmatrix} 1 & z & z^2 & \ldots \end{bmatrix} R \begin{bmatrix} 1 & w & w^2 & \ldots \end{bmatrix}^* .$$

Lev-Ari and Kailath [130, 134] studied structured matrices R whose generating functions can be expressed in the general form

$$R(z, w) = \frac{G(z) J G^*(w)}{d(z, w)}, \tag{8.1a}$$

where J is any Hermitian constant nonsingular matrix, $d(z,w)$ is the generating function of a constant (possibly singular) Hermitian matrix $\mathbf{d} = [d_{mj}]_{m,j=0}^{\infty}$ viz.,

$$d(z,w) = \begin{bmatrix} 1 & z & z^2 & \dots \end{bmatrix} \mathbf{d} \begin{bmatrix} 1 & w & w^2 & \dots \end{bmatrix}^* ,$$

and $G(z)$ is a $1 \times r$ row vector function, where r is called the displacement rank of R. Equation (8.1a) has a matrix domain equivalent that shows that it can be regarded as a generalization of the earlier definition (2.1b). If we write $d(z,w) = \sum_{m,j=0}^{\infty} d_{mj} z^m w^{*j}$, then (8.1a) can be alternatively expressed in the form

$$\sum_{m,j=0}^{\infty} d_{mj} Z^m R Z^{*j} = GJG^* \quad , \tag{8.1b}$$

where each column of the matrix G is formed by stacking the coefficients of the corresponding function in $G(z)$, *i.e.*, $G(z) = \begin{bmatrix} 1 & z & z^2 \dots \end{bmatrix} G$.

As simple examples, consider the case of Hermitian Toeplitz and Hankel matrices ($T = [c_{i-j}]_{i,j=0}^{\infty}$, $c_i = c_{-i}^*$, and $H = [h_{i+j}]_{i,j=0}^{\infty}$), for which it can be seen by direct calculation that

$$T(z,w) = \frac{c(z) + c^*(w)}{2(1 - zw^*)} \quad \text{and} \quad H(z,w) = \frac{[zh(z) - w^*h^*(w)]}{(z - w^*)} \quad , \tag{8.2}$$

respectively, where $c(z)$ and $h(z)$ are given by

$$c(z) = c_0 + 2zc_1 + 2z^2c_2 + \dots \quad \text{and} \quad h(z) = h_0 + zh_1 + z^2h_2 + \dots$$

The expression for $T(z,w)$ can also be written in the alternative form

$$T(z,w) = \frac{x(z)x^*(w) - y(z)y^*(w)}{1 - zw^*} ,$$

where

$$2x(z) = c(z) + 1 \quad \text{and} \quad 2y(z) = c(z) - 1 .$$

By writing the above as $(1 - zw^*)T(z,w) = x(z)x^*(w) - y(z)y^*(w)$, we can easily recognize the matrix equivalent (compare with (2.3a))

$$T - \mathcal{Z}T\mathcal{Z}^* = \begin{bmatrix} \mathbf{x}_0 & \mathbf{y}_0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 & \mathbf{y}_0 \end{bmatrix}^* ,$$

where the semi-infinite columns $\mathbf{x}_0$ and $\mathbf{y}_0$ are formed by stacking the coefficients of $x(z)$ and $y(z)$, respectively. Expression (8.1b) is clearly more general than (2.1b) and it includes a larger class of structured matrices, such as sums of Toeplitz and Hankel matrices [183]: assume R is the sum of a Hermitian Toeplitz matrix and a symmetric Hankel matrix. Using (8.2) we readily verify that the associated generating function is equal to

$$R(z,w) = \frac{G(z)JG^*(w)}{j(1 - zw^*)(z - w^*)} ,$$

where $G(z) = \begin{bmatrix} 1 & z & c(z) + z^2h(z) & z\,[c(z) + h(z)] \end{bmatrix}$ and

$$J = \begin{bmatrix} & & & -j \\ & & j & \\ & -j & & \\ j & & & \end{bmatrix} , \quad j = \sqrt{-1}.$$

Further examples and discussion of the more general class of sums of quasi-Toeplitz and quasi-Hankel matrices can be found in [183] – see also [80, 98, 143, 197, 198] for alternative appoaches to the factorization of sums of Toeplitz and Hankel matrices.

We may remark that the function defined by the equation $d(z,z) = 0$ demarcates various regions in the complex plane (see Section 8.2). When $d(z,w) = 1 - zw^*$, the boundary is the unit circle, when $d(z,w) = z - w^*$, the boundary is the real axis. This remark sheds further light on our earlier (perhaps cryptic remarks to readers not versed in linear system theory) that Toeplitz-like and Hankel-like displacements corresponded to unit circle (discrete-time) and half-plane (continuous-time) systems, respectively.

We now show how to extend Schur's algorithm, viz., recursion (3.6a), to the class of matrices described by (8.1a). We shall see ahead that for such an extension to be possible, the freedom in choosing $d(z,w)$ has to be limited (see (8.5a)).

Returning to the general definition (8.1a), and using the notation of generating functions, it is easy to check that the Schur reduction procedure (4.1b) can be written in the equivalent form:

$$d_i = R_i(0,0), \quad l_i(z) = R_i(z,0) \ , \tag{8.3a}$$

$$zw^* R_{i+1}(z,w) = R_i(z,w) - R_i(z,0)R_i^{-1}(0,0)R_i(0,w) \ , \tag{8.3b}$$

where $l_i(z) = \begin{bmatrix} 1 & z & z^2 & \ldots \end{bmatrix} l_i$, and $R_i(z,w)$ denotes the generating function of the i^{th} Schur complement. Notice that the (nonzero parts of the) i^{th} column of the triangular factor L of the given finite matrix R is obtained by considering the leading $(n-i)$ coefficients of $l_i(z)$.

Our purpose is to show that if we start with a generating function $R(z,w)$ that has the form given by (8.1a) then, under an additional condition to be specified ahead, the successive Schur complements of R will also have a similar form. To verify this, we proceed by induction: suppose we can write $R_i(z,w)$ in the form

$$R_i(z,w) = \frac{G_i(z)JG_i^*(w)}{d(z,w)} \ ,$$

(this is certainly true for $i = 0$) then substitution into (8.3b) yields

$$zw^* R_{i+1}(z,w) = \frac{G_i(z)\{J - \frac{d(z,w)}{d(z,0)d(0,w)} JM_iJ\}G_i^*(w)}{d(z,w)} \ ,$$

where we defined $M_i = G_i^*(0)R_i^{-1}(0,0)G_i(0)$. [The strong regularity of R guarantees that $R_i(0,0) \neq 0$. Also note that $G_i(0)$ is equal to the top row of G_i and that $R_i(0,0)$ is equal to the $(0,0)$ entry of R_i.]. If we can now find a matrix function $\Theta_i(z)$ such that

$$\Theta_i(z)J\Theta_i^*(w) = J - \frac{d(z,w)}{d(z,0)d(0,w)} JM_iJ \ , \tag{8.4a}$$

then we can write

$$zw^* R_{i+1}(z,w) = \frac{G_i(z)\Theta_i(z)J\Theta_i^*(w)G_i^*(w)}{d(z,w)} \ . \tag{8.4b}$$

This shows that in order to reduce the right-hand side of the above expression to a form similar to (8.1a) all we need to do is to define the following row function (compare with (3.6a))

$$zG_{i+1}(z) = G_i(z)\Theta_i(z) \ , \quad G_0(z) = G(z) \ , \tag{8.4c}$$

which tells us how to update $G_i(z)$. In this case, we can rewrite (8.4b) as

$$R_{i+1}(z,w) = \frac{G_{i+1}(z)JG_{i+1}^*(w)}{d(z,w)} \ , \tag{8.4d}$$

which is the desired form. It also follows that we can determine d_{i+1} and $l_{i+1}(z)$ from $G_{i+1}(z)$ without the need to explicitly evaluate $R_{i+1}(z,w)$, viz.,

$$d_{i+1} = \lim_{z\to 0} \frac{G_{i+1}(z)JG_{i+1}^*(z)}{d(z,z)} \ , \quad l_{i+1}(z) = \frac{G_{i+1}(z)JG_{i+1}^*(0)}{d(z,0)} \ . \tag{8.4e}$$

All that remains to be shown is that a $\Theta_i(z)$ exists that satisfies (8.4a), and if so to find an explicit expression for it. It turns out that to do this, $d(z,w)$ needs to be further constrained as in (8.5a) below.

THEOREM 8.1. *Let R be a Hermitian structured matrix whose generating function is given by*

$$R(z,w) = \frac{G(z)JG^*(w)}{d(z,w)}.$$

If $d(z,w)$ is of the form

$$d(z,w) = a(z)a^*(w) - f(z)f^*(w) \ , \tag{8.5a}$$

for some functions $a(z)$ and $f(z)$, then the triangular factorization of R can be carried out recursively using (8.4c) and (8.4e) with

$$\Theta_i(z) = \left\{ I - \frac{d(z,\tau)}{d(z,0)d(0,\tau)} JM_i \right\} \Theta_i \ , \tag{8.5b}$$

where Θ_i is an arbitrary $J-$unitary matrix ($\Theta_i J \Theta_i^* = J$) and τ is any scalar such that $d(\tau,\tau) = 0$.

Proof. We refer to [131, 134] for a proof that (8.5a) ensures that some $\Theta_i(z)$ exists, because it is a bit long (and could probably be improved). Here, however it will be useful to indicate how (8.5b) follows under the assumption that (8.4a) has a solution $\Theta_i(z)$. First we note that if $\Theta_i(z)$ exists, it is not unique because it can be replaced by $\Theta_i(z)\Theta_i$, for any Θ_i such that $\Theta_i J \Theta_i^* = J$. If we choose any scalar such that $d(\tau,\tau) = 0$, it then follows from (8.4a) that $\Theta_i(\tau)$ is a $J-$unitary matrix, and that we can express $\Theta_i(z)$ as in (8.5b). □

We may remark that in matrix form the descriptions (8.1a) and (8.5a) imply that R satisfies a displacement equation of the form

$$ARA^* - FRF^* = GJG^* \ ,$$

where A and F are lower triangular Toeplitz matrices that are formed from the coefficients of $a(z)$ and $f(z)$, respectively. This is of course only a special case of the generalized displacement (7.24); however we may note that by appropriate use of

Newton series and divided-difference matrices (rather than Taylor series and Toeplitz matrices) Lev-Ari [130] was able to obtain generator recursions equivalent to those of Corollary 7.5 for the case of nonderogatory matrices F that are not necessarily Toeplitz.

The generating function approach allows an interesting generalization of the Schur algorithm to factorizations of the form $R = QDQ^*$, Q not necessarily lower triangular [130]. This can be useful in various applications, e.g., in root distribution theory (see Section 8.2 below). The generalization is obtained by replacing (8.4c) by

$$(z-\xi_i)G_{i+1}(z) = G_i(z)\Theta_i(z), \quad G_0(z) = G(z),$$

where ξ_i is any point in the complex plane (though the most common choices are $\xi_i = 0, \pm 1$). Now $\Theta_i(z)$ is correspondingly generalized from (8.5b) to read

$$\Theta_i(z) = \left\{ I - \frac{d(z,\tau)}{d(z,\xi_i)d(\xi_i,\tau)} JG_i^*(\xi_i)R^{-1}(\xi_i,\xi_i)G_i(\xi_i) \right\} \Theta_i ,$$

where τ is any scalar such that $d(\tau,\tau) = 0$ and Θ_i is an arbitrary $J-$unitary matrix.

8.1. An Example: Quasi-Toeplitz Matrices. We now specialize, for the sake of illustration, the recursions of Theorem 8.1 to a Hermitian positive-definite quasi-Toeplitz matrix R, viz., a matrix whose generating function has the form

$$R(z,w) = \frac{G(z)JG^*(w)}{1-zw^*} ,$$

where $G(z)$ is a 1×2 row vector function and $J = (1 \oplus -1)$. Notice that $d(z,w) = (1-zw^*)$ satisfies condition (8.5a). Expression (8.5b) has two degrees of freedom, viz., the $J-$unitary matrix Θ_i and the scalar τ. A simple choice for τ is $\tau = 1$. A convenient choice for Θ_i is to choose it as a 2×2 hyperbolic rotation that annihilates the second entry of the top row of G_i, which is equal to $G_i(0)$. The positive-definiteness of R guarantees that the corresponding reflection coefficient γ_i will be strictly bounded by one. Substituting this choice for Θ_i, along with $\tau = 1$, into expression (8.5b), it follows, upon simplification, that it collapses to the form

$$\Theta_i(z) = \frac{1}{\sqrt{1-|\gamma_i|^2}} \begin{bmatrix} 1 & -\gamma_i \\ -\gamma_i^* & 1 \end{bmatrix} \begin{bmatrix} z & 0 \\ 0 & 1 \end{bmatrix} , \tag{8.6}$$

which is precisely the same expression for $\Theta_i(z)$ in Schur's linearized form (3.6a).

8.2. An Application: Bezoutians and Root Distribution Problems. There is a long history of studies on the problem of tests for determining the distribution with respect to different regions (in particular, half planes and discs) of the roots of a given polynomial. Most of the literature on this topic falls into one of two major classes: (i) various matrix criteria (e.g., those of Hermite (1856), Lyapunov (1893), and more recent results of Kalman (1969) and Jury-Gutman (1981)) and (ii) efficient computational algorithms (*e.g.*, those of Routh (1857), Hurwitz (1895), Schur (1917), Cohn (1922)). Most of the literature treats the first class of problems, presumably because the classical efficient algorithms can hardly be improved upon. Nevertheless, there is room for dissatisfaction with the current state of knowledge, even if only from a pedagogical point of view. Among the reasons for this are: 1) the miscellany of derivations for each test, 2) the differences between the forms and the derivations of the imaginary axis and unit circle tests, 3) the generally *ad hoc* procedures for handling singular cases, 4) the continuing "surprising" discoveries of new tests. Now it turns out that one of the oldest criteria for root distribution with respect to the left-half plane – the one of Hermite (1856) – involves a particular kind of structured matrix (a Bezoutian), and so in fact does perhaps the earliest unit circle test – the one of Schur (1917).

Therefore, it was interesting to explore the results of applying our general fast algorithms for matrices with displacement structure to the special Bezoutian matrices for root distribution problems. The results were encouraging. A unified approach could be found to half-plane and unit-circle problems. The natural algorithms arising from the analysis were essentially the same as the Routh-Hurwitz and Schur-Cohn tests - slightly different in form but identical in the number of computations. Moreover, a whole family of algorithms with identical computational burden could be identified and parametrized. A classification scheme was evident for grouping various algorithms - old ones and any yet-to-be proposed. Finally a systematic method for handling singularities becomes available. A more detailed account of our results in this direction can be found in [25, 132, 152, 154].

To give a flavor of our approach, recall the function $d(z,w)$ of (8.5a),

$$d(z,w) = a(z)a^*(w) - f(z)f^*(w) ,$$

and define[1]

$$\Omega_+ = \{z|d(z,z) > 0\}, \quad \Omega_- = \{z|d(z,z) < 0\}, \quad \partial\Omega = \{z|d(z,z) = 0\}.$$

One is generally interested in three special cases.

1. Real Axis: $d_R(z,w) = j(z - w^*)$. In this case

$$\partial R = \{z|j(z - z^*) = Im\, z = 0\},$$

so that $R_\pm$ are, respectively, the open upper and lower half planes.

2. Imaginary Axis: $d_I(z,w) = (z + w^*)$. In this case

$$\partial I = \{z|(z + z^*) = Re\, z = 0\},$$

so that $I_\pm$ are, respectively, the open left and right half planes.

3. Unit Circle: $d_T(z,w) = 1 - zw^*$. In this case

$$\partial T = \{z|1 - |z|^2 = 0\},$$

so that $T_\pm$ are, respectively, the exterior and interior of the unit circle.

Given a polynomial

$$p(z) = p_{n,0} + p_{n,1}z + \ldots + p_{n,n}z^n,$$

we shall define $p_\Omega^\#(z)$ as the *polynomial reflection* of $p(z)$ with respect to $\partial\Omega$. Instead of exploring a general definition, we shall content ourselves with defining it in the three special cases mentioned above:

[1]We remark that the symbol Ω is used in this section to denote regions of interest in the plane, and it should not be confused with the matrix Ω that we used earlier in (7.24). The distinction is obvious from the context. Likewise, the symbols R, T, and I are used here to denote specific regions in the plane rather than matrices.

1. $p_R^{\#}(z) = [p(z^*)]^*$, where the notation $[\cdot]^*$ denotes conjugation with respect to both the coefficients and the variable. Note that $p_R^{\#}(z)$ corresponds to reflecting the roots of $p(z)$ into their images with respect to the real axis.

2. $p_I^{\#}(z) = [p(-z^*)]^*$, which corresponds to reflecting the roots to their images in the imaginary axis. If $p(z)$ has only real coefficients, then $p_I^{\#}(z) = p(-z)$.

3. $p_T^{\#}(z) = z^{\deg p(z)}[p(1/z^*)]^*$, which corresponds to reflection with respect to the unit circle. Alternatively, $p_T^{\#}(z)$ is the so-called conjugate reverse polynomial,

$$p_T^{\#}(z) = p_{n,0}^* z^n + p_{n,1}^* z^{n-1} + \ldots + p_{n,n}^*.$$

Next we define a Bezoutian form

$$B_\Omega(z,w) = \frac{p(z)[p^*(w)]^* - p^{\#}(z)[p^{\#}(w)]^*}{d_\Omega(z,w)},$$

where again we shall restrict Ω to be R, I, or T. The choices $\Omega = R$ corresponds to the classical Bezoutian form introduced by Hermite (1856); $\Omega = I$ was studied by Fujiwara (1926); $\Omega = T$ by Schur (1917) and Cohn (1922).

It is not hard to see that the Bezoutian form is a polynomial of degree n in both z and w^*, so that it is a quadratic form associated with a matrix B_Ω,

$$B_\Omega(z,w) = \sum_{i=0}^{n}\sum_{j=0}^{n} z^i w^{*j} b_{ij}^{\Omega}, \quad B_\Omega = [b_{ij}^{\Omega}].$$

It can be shown that the Bezoutian matrix B_Ω is Hermitian. Now we can state (a slight generalization of) Hermite's original result; various proofs can be given – see, e.g., [132].

THEOREM 8.2. *Let $\{\pi, \eta, \nu\}$ be the inertia of B_Ω,* i.e. , *B_Ω has π strictly positive eigenvalues, η zero eigenvalues, and ν strictly negative eigenvalues.*

Then $p(z)$ and $p^{\#}(z)$ will have η roots in common; of the remaining roots of $p(z)$, π will lie in Ω_- and ν will lie in Ω_+.

Therefore, the root distribution problem can be reduced to one of finding the inertia of B_Ω. One method is to seek a factorization

$$B_\Omega = QDQ^*,$$

where D is diagonal (or at least block diagonal). Then by a famous theorem of Sylvester (that congruence preserves inertia), $In\ B_\Omega = In\ D$, and now the inertia is easy to find. One approach is to try to find Q as a lower-triangular matrix, which can be achieved with a diagonal D if and only if all the leading minors of B_Ω are nonzero. In this case, B_Ω is said to be strongly regular, and it is associated with nonsingular root distribution problems (e.g., those with no degeneracies in the Routh table). When some minors are zero, the best we can get with triangular Q is a block diagonal D, with blocks whose sizes correspond in a certain way to the patterns of zero minors in B_Ω. Here we shall assume the strongly regular case; the singular case is studied in [155, 180].

The triangular factorization of an $n \times n$ matrix can be achieved in $O(n^3)$ flops; however the Bezoutian matrix clearly has displacement structure, which can be exploited to find the factorization and the inertia with $O(n^2)$ computations, and moreover to do this using only the coefficients of $p(z)$ without explicitly forming the Bezoutian matrix B_Ω. Here we illustrate how the famous Routh test, usually derived via complex function arguments (e.g., Rouche's theorem), falls out from the Schur algorithm applied to a Bezoutian matrix.

8.2.1. Tests of Routh-Hurwitz Type. For reasons of simplicity, we shall assume that the polynomial $p(z)$ has real coefficients. Then $p^{\#}(z) = p(-z)$ and

$$B_I(z,w) = \frac{G(z)JG^*(w)}{d_I(z,w)}$$

with

$$d(z,w) = z + w^*, \quad G(z) = \begin{bmatrix} p(z) & p(-z) \end{bmatrix}, \quad J = (1 \oplus -1).$$

Then

$$\frac{d(z,\tau)}{d(z,0)d(0,\tau)} = \frac{1}{z} + \frac{1}{\tau^*}.$$

Here τ is an arbitrary point on the imaginary axis; we could take it as j, but the choice $\tau = j\infty$ seems even better. Now $B(0,0) = 0/0$, and so we use L'Hospital's rule to evaluate it as

$$B(0,0) = 2p(0)p'(0) = 2p_{n,0}p_{n,1}.$$

Therefore,

$$M_0 = \begin{bmatrix} p(0) \\ p(0) \end{bmatrix} B^{-1}(0,0) \begin{bmatrix} p(0) & p(0) \end{bmatrix} = \rho_0 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

where

$$\rho_0 = \frac{p^2(0)}{2p(0)p'(0)} = \frac{p_{n,0}}{2p_{n,1}},$$

and

$$\Theta_0(z) = \begin{bmatrix} 1 - \rho_0 z^{-1} & -\rho_0 z^{-1} \\ \rho_0 z^{-1} & 1 + \rho_0 z^{-1} \end{bmatrix} \Theta_0.$$

If we choose $\Theta_0 = J$, then we can check that $G_1(z)$ will have the form $\begin{bmatrix} p_1(z) & p_1(-z) \end{bmatrix}$ so that the recursion can be continued. In fact, if we write

$$p_i(z) = p_{i,0} + p_{i,1}z + \ldots + p_{i,n-i}z^{n-i},$$

then it is easy to see that

$$d_i = B_i(0,0) = 2p_i(0)p_i'(0), \quad \rho_i = p_i^2(0)/d_i,$$

$$\Theta_i(z) = \begin{bmatrix} 1 - \rho_i z^{-1} & -\rho_i z^{-1} \\ \rho_i z^{-1} & 1 + \rho_i z^{-1} \end{bmatrix} \Theta_i$$

and the recursion $zG_{i+1}(z) = G_i(z)\Theta_i(z)$ becomes

$$zp_{i+1}(z) = p_i(z) - \rho_i z^{-1}[p_i(z) - p_i(-z)]. \tag{8.7}$$

Note that $[p_i(z) - p_i(-z)]$ is odd, and therefore divisible by z; also that when $z = 0$ the right-hand side is zero. Therefore $p_{i+1}(z)$ has degree less than $p_i(z)$, and the recursions will terminate in n steps or less. Note also that the inertia depends upon the signs of the $\{d_i\}$ or equivalently of the $\{\rho_i = p_i^2(0)/d_i\}$.

Finally, as for the computational effort, note that at the i^{th} stage we have $(n - i)/2$ multiplications (even coefficients $\times \rho_i$) and $(n-i)/2$ additions (modify the even coefficients of $p_i(z)$). This is exactly the same as for the celebrated Routh recursions, which however are slightly different in form. However they are easily obtained by adding the recursions for $p_i(z)$ and $p_i(-z)$. In fact if

$$2m_i(z) = p_i(z) + p_i(-z), \quad 2n_i(z) = p_i(z) - p_i(-z),$$

the recursion (8.7) becomes exactly the Routh recursion

$$zm_{i+1}(z) = n_i(z), \quad zn_{i+1}(z) = m_i(z) - 2\rho_i z^{-1} n_i(z).$$

The reader can check that these expressions will be obtained directly if we started with

$$G_0(z)U = \begin{bmatrix} p_0(z) & p_0(-z) \end{bmatrix} U, \quad \text{with } U = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

However our derivation shows that there are many tests of complexity equivalent to that of the Routh test – we gave one, but many others can be obtained by choosing differently the free parameters associated with the recursions of Theorem 8.1.

8.3. Remark on Reproducing Kernel Hilbert Spaces. Lev-Ari and Kailath [133] pointed out that the generating functions $R(z, w)$ in Theorem 8.1 could be regarded as the reproducing kernels for certain Hilbert spaces of analytic functions (see, e.g., [16], [62]). This provides some new ways of looking at these theories, e.g., the famous Szegö kernel, $R(z, w) = 1/(1 - zw^*)$ is the generating function of the (identity) covariance matrix of a white noise process. So also, the kernel $T(z, w) = (c(z) + c^*(w))/(2(1 - zw^*))$ in (8.2), which arises from a Toeplitz covariance, occurs frequently in the work of deBranges, who characterizes $T(z, w)$ much more abstractly via certain resolvent properties. Dym and his colleagues, esp. Alpay and Dewilde, noted the connection with the work of deBranges; they have developed it extensively and also applied it to the study of displacement structure (see, e.g., [6, 7, 8, 71, 72]). It would take us too far afield to describe their approach here, esp. since their main focus is not the factorization problem which we regard as central. We may note that our specification of the form (8.5a) for the (domain) function $d(z, w)$ has led to the study of a new class of reproducing kernel spaces – see [10].

Here we shall reinforce the value of making connections with stochastic processes by showing how displacement structure can be enhanced by combining it with state-space structure, and especially the computationally-oriented Kalman-filter theory, which has been extensively developed by engineers over the last 30 years.

9. Incorporating State-Space Structure. We have studied displacement structure at some length. A very widely studied structure in system theory is state-space structure. It turns out that we can often combine the two quite effectively. In particular, while fast triangular factorization algorithms for $N \times N$ matrices R_Y with displacement structure reduce the computational complexity from $O(N^3)$ to $O(N^2)$, if the matrices arise from a system with n states, the computational effort can be reduced to $O(Nn^3)$ and when displacement structure is also present, to $O(Nn^2)$. Since the number of states n, is often much less than the number of measurements, N, the presence of state-space structure enables significant computational savings.

To be more specific, we shall focus largely on the case where the matrix R_Y is the covariance matrix of a set of zero-mean random variables,

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \end{bmatrix}, \quad R_Y = E\mathbf{Y}\mathbf{Y}^*,$$

where the letter E denotes expected value. The $\{\mathbf{y}_i\}$ are further assumed to be obtained from the following state-space model:

$$\begin{aligned} \mathbf{x}_{i+1} &= F_i\mathbf{x}_i + G_i\mathbf{u}_i \\ \mathbf{y}_i &= H_i\mathbf{x}_i + \mathbf{v}_i \ , \end{aligned} \tag{9.1a}$$

where F_i, G_i and H_i are $n\times n, n\times m$, and $p\times n$ known matrices, respectively, while $\mathbf{x}_0$, $\{\mathbf{u}_i\}$ and $\{\mathbf{v}_i\}$ are zero-mean random variables with specified second-order statistics

$$E\begin{bmatrix} \mathbf{u}_i \\ \mathbf{v}_i \\ \mathbf{x}_0 \\ 1 \end{bmatrix}\begin{bmatrix} \mathbf{u}_j \\ \mathbf{v}_j \\ \mathbf{x}_0 \end{bmatrix}^* = \begin{bmatrix} Q_i\delta_{ij} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & R_i\delta_{ij} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Pi_0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{9.1b}$$

Here δ_{ij} is the Kronecker delta function equal to unity when $i = j$ and zero elsewhere. Of course, the $\{\mathbf{x}_i\}$ and $\{\mathbf{y}_i\}$ will also be random variables with statistics that can be computed as follows: let

$$\Pi_i = E\mathbf{x}_i\mathbf{x}_i^*, \quad \text{the state covariance matrix.}$$

Then given the initial values Π_0 we can find the Π_i via the (Lyapunov) recursion

$$\Pi_{i+1} = F_i\Pi_iF_i^* + G_iQ_iG_i^*, \quad i \geq 0. \tag{9.2a}$$

Next, we have $R_Y = E\mathbf{Y}\mathbf{Y}^* = [E\mathbf{y}_i\mathbf{y}_j^*]_{i,j=0}^{N-1}$ where

$$E\mathbf{y}_i\mathbf{y}_j^* = \begin{cases} H_iF_iF_{i-1}\dots F_j\Pi_iH_j^* & \text{if } i > j \\ H_i\Pi_iH_i^* + R_i & \text{if } i = j \\ H_i\Pi_iF_i^*F_{i+1}^*\dots F_j^*H_j^* & \text{if } i < j \end{cases} \tag{9.2b}$$

We shall assume that the matrix R_Y is strictly positive-definite, which can be assured by various reasonable assumptions on the system parameters (e.g., $R_i > \mathbf{0}$).

We see that the entries of R_Y are essentially determined by size n matrices, which is what allows the possibility of computational reduction. It can be shown, though we shall not prove this here, that the triangular factorization of R_Y can now be carried out with $O(Nn^3)$ elementary operations. The key calculation turns out to be that of certain $n \times p$ matrices $\{K_i\}$, which can be done in various ways. The best known is one introduced in the celebrated Kalman filter [123] and involves the so-called discrete Riccati recursion for certain $n \times n$ matrices $\{P_i\}$,

$$P_{i+1} = F_iP_iF_i^* + G_iQ_iG_i^* - F_iP_iH_i^*R_{e,i}^{-1}H_iP_iF_i^*, \quad P_0 = \Pi_0, \tag{9.3}$$

where $R_{e,i} = R_i + H_i P_i H_i^*$. The invertibility of the $\{R_{e,i}\}$ is assured by (in fact, is equivalent to) the assumed positive-definiteness of the matrix R_Y, whose (block) triangular factorization,

$$R_Y = LD^{-1}L^*,$$

turns out to be specified by $D =$ diagonal $\{R_{e,i}\}$, and

$$\begin{aligned} l_i &= \text{the nonzero part of the } i^{th} \text{ column of } L \\ &= \begin{bmatrix} R_{e,i} \\ HK_i \\ HF_{i+1}K_i \\ \vdots \\ HF_{N-1}\ldots F_{i+1}K_i \end{bmatrix}, \quad \text{where } K_i = F_i P_i H_i^*. \end{aligned}$$

We can see that each iteration (*i.e.*, computing P_{i+1} from P_i) of the Riccati recursion takes $O(n^3)$ elementary operations, so that the complete triangular factorization of R_Y takes $O(Nn^3)$ elementary operations, rather than $O(N^3)$.

To make a connection with the material in earlier sections, it is easiest to begin by restricting attention to the special case where all the model parameters are time-invariant (e.g., $F_i = F$). Assume also that the eigenvalues of F have magnitude less than unity (*i.e.*, F is a stable matrix) and $\Pi_0 = \bar{\Pi}$, where $\bar{\Pi}$ is the unique nonnegative definite solution of the (discrete-time Lyapunov) equation (*cf.* (9.2a))

$$\bar{\Pi} = F\bar{\Pi}F^* + GQG^*.$$

Then it is not hard to see from the expressions (9.2b) that now R_Y will be Toeplitz,

$$E\mathbf{y}_i\mathbf{y}_j^* = \begin{cases} HF^{i-j}\bar{\Pi}H^* & \text{if } i > j \\ H\bar{\Pi}H^* + R & \text{if } i = j \\ H\bar{\Pi}F^{*(j-i)}H^* & \text{if } i < j \end{cases}$$

If we did not account for the state-space structure, the inversion of the Toeplitz matrix R_Y would require $O(N^2)$ operations; as we have just seen, state-space structure allows this to be reduced to $O(Nn^3)$. But the additional assumption of constant parameters should buy us more. In fact, we can manage in the Toeplitz case with $O(Nn^2)$ operations rather than $O(Nn^3)$. And in fact, for any set of constant parameters and any choice of Π_0 (e.g., F not necessarily stable), we can manage with $O(Nn^2\alpha)$ operations, where

$$\alpha = \text{rank}(F\Pi_0F^* + GQG^* - K_0R_{e,0}^{-1}K_0^* - \Pi_0) = \text{rank }(P_1 - P_0).$$

This was shown by Kailath et al. [107, 118, 122] in their studies of recursive state-space estimation. They showed that one could alternatively compute $\{K_i, R_{e,i}\}$ via the so-called Chandrasekhar-type recursions

$$\begin{aligned} K_{i+1} &= K_i - FL_iR_{r,i}^{-1}L_i^*H^* \\ L_{i+1} &= FL_i - K_iR_{e,i}^{-1}HL_i \\ R_{e,i+1} &= R_{e,i} - HL_iR_{r,i}^{-1}L_i^*H^* \\ R_{r,i+1} &= R_{r,i} - L_i^*H^*R_{e,i}^{-1}L_i^*H^*, \end{aligned} \tag{9.4}$$

where L_0 and $R_{r,0}$ are defined (nonuniquely) via the factorization

$$F\Pi_0 F^* + GQG^* - K_0 R_{e,0}^{-1} K_0^* - \Pi_0 = -L_0 R_{r,0}^{-1} L_0^*.$$

The reason for the name is that the above recursions can be seen to be discretizations of certain partial differential equations (the famous X and Y equations) introduced by Chandrasekhar [45] to solve certain equations of the Wiener-Hopf type.

9.1. Chandrasekhar-Type Recursions in Array Form. It has been useful, as with the Schur algorithm, to rewrite the above equations in array form, which was done as follows [122]: introduce the (nonunique) factorization

$$P_1 - P_0 = F\Pi_0 F^* + GQG^* - K_0 R_{e,0}^{-1} K_0^* - \Pi_0 = L_0 S L_0^*,$$

where S is an $\alpha \times \alpha$ signature matrix and apply any $(I \oplus S)-$unitary matrix $\mathbf{\Theta}_1$ (or a sequence of elementary matrices) that triangularizes the prearray shown below

$$\begin{bmatrix} R_{e,0}^{1/2} & HL_0 \\ K_0 R_{e,0}^{-1/2} & FL_0 \end{bmatrix} \mathbf{\Theta}_1 = \begin{bmatrix} X & \mathbf{0} \\ Y & Z \end{bmatrix}, \tag{9.5a}$$

where for a positive-definite matrix A, a square-root factor is defined as any matrix, say $A^{1/2}$, such that $A = (A^{1/2})(A^{1/2})^*$. Such square-root factors are clearly not unique. They can be made unique, e.g., by insisting that the factors be Hermitian or that they be triangular (with positive diagonal elements). In most applications, the triangular form is preferred.

By "squaring" both sides of (9.5a) in the $(I \oplus S)$-norm, we can see that

$$XX^* = R_{e,1}, \quad YX^* = K_1 \text{ and } ZSZ^* = P_2 - P_1,$$

which implies that we can make the identifications $X = R_{e,1}^{1/2}$ and $Y = K_1 R_{e,1}^{-1/2} \equiv \bar{K}_{p,1}$, and conclude that the difference $(P_2 - P_1)$ has the same inertia matrix, S, as $(P_1 - P_0)$,

$$P_2 - P_1 = L_1 S L_1^*, \text{ say.}$$

We can thus take Z as L_1 and conclude that the first step of the array recursion is given by

$$\begin{bmatrix} R_{e,0}^{1/2} & HL_0 \\ K_0 R_{e,0}^{-1/2} & FL_0 \end{bmatrix} \mathbf{\Theta}_1 = \begin{bmatrix} R_{e,1}^{1/2} & \mathbf{0} \\ \bar{K}_{p,1} & L_1 \end{bmatrix}. \tag{9.5b}$$

More generally, the array form of the Chandrasekhar recursions are given by

$$\begin{bmatrix} R_{e,i}^{1/2} & HL_i \\ K_i R_{e,i}^{-1/2} & FL_i \end{bmatrix} \mathbf{\Theta}_{i+1} = \begin{bmatrix} R_{e,i+1}^{1/2} & \mathbf{0} \\ \bar{K}_{p,i+1} & L_{i+1} \end{bmatrix}, \tag{9.6}$$

where $\mathbf{\Theta}_{i+1}$ is any $(I \oplus S)$-unitary matrix that triangularizes the prearray.

9.2. Connections to Generalized Schur Algorithms. Soon after their introduction, the Chandrasekhar recursions were recognized to be closely related to the well-known Levinson algorithm [137] for solving the discrete-time analog of the finite-time Wiener-Hopf equation, viz., a linear equation with a Toeplitz coefficient matrix, and more precisely to certain generalizations of the Levinson algorithm [79] devised to account for the fact that the appropriate coefficient matrix for constant-parameter state-space systems is not Toeplitz but is Toeplitz-like. But as the displacement theory began to be better understood, it was realized that the proper connection was to the Schur algorithm. What we shall show now is that when the extra structure provided by an underlying state-space model is properly incorporated into the generalized Schur algorithm, it reduces to the Chandrasekhar recursions [181]. Moreover, this interpretation allowed us to generalize the Chandrasekhar recursions to certain types of time-variant systems [178], which fortunately also arise in important applications [179] (see (9.9a) and Section 9.3).

First we need some notation to incorporate the fact that the covariance matrix R_Y has $p \times p$ matrix entries $E\mathbf{y}_i\mathbf{y}_j^*$. Thus note that Z^p is a lower triangular shift matrix with ones on the p^{th} subdiagonal and zeros elsewhere, so that multiplying a column vector by Z^p corresponds to shifting its entries down by p positions and introducing p zeros on top. The special structure that will be relevant to our discussions here is the case where R_Y satisfies a displacement equation of the form

$$R_Y - Z^p R_Y \left[Z^p\right]^* = \mathcal{G} J \mathcal{G}^* \ , \tag{9.7a}$$

where J is a $p \times \alpha$ signature matrix and $\mathcal{G}$ is a generator matrix with $p+\alpha$ columns. By suitably extending Theorem 4.2, we can see that the triangular factorization of such an R_Y can be recursively computed as follows: start with $\mathcal{G}_0 = \mathcal{G}$ and repeat for $i \geq 0$:

1. Determine a $J-$unitary matrix $\mathbf{\Theta}_i$ that reduces the top p rows of $\mathcal{G}_i$ (denoted by g_i) to the form $g_i\mathbf{\Theta}_i = \begin{bmatrix} X & \mathbf{0} \end{bmatrix}$, where X is a $p \times p$ matrix. That is, a $p \times \alpha$ zero-block is introduced in $g_i\mathbf{\Theta}_i$;

2. Shift down the first p columns of $\mathcal{G}_i\mathbf{\Theta}_i$ by p steps and keep the last α columns unaltered,

$$\begin{bmatrix} \mathbf{0}_{p\times(p+\alpha)} \\ \mathcal{G}_{i+1} \end{bmatrix} = Z^p \mathcal{G}_i \mathbf{\Theta}_i \begin{bmatrix} I_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_\alpha \end{bmatrix} + \mathcal{G}_i \mathbf{\Theta}_i \begin{bmatrix} \mathbf{0}_p & \mathbf{0} \\ \mathbf{0} & I_\alpha \end{bmatrix}, \quad \mathcal{G}_0 = \mathcal{G}. \tag{9.7b}$$

To compute the displacement of R_Y, let us first note from the basic formulas (9.2a)–(9.2b) for its entries that

$$\Pi_{i+1} - \Pi_i = F^i \Delta F^{*i} \ , \ \ \Delta = \Pi_1 - \Pi_0,$$

and that

$$\begin{aligned} E\mathbf{y}_i\mathbf{y}_i^* - E\mathbf{y}_{i-1}\mathbf{y}_{i-1}^* &= HF^{(i-1)}\Delta F^{*(i-1)}H^* \\ E\mathbf{y}_i\mathbf{y}_{i+1}^* - E\mathbf{y}_{i-1}\mathbf{y}_i^* &= HF^{(i-1)}\Delta F^{*i}H^*. \end{aligned}$$

From these identities, it follows readily that

$$\nabla R_Y = R_Y - Z^p R_Y \left[Z^p\right]^* = \begin{bmatrix} R_{e,0} & K_0^* H^* & K_0^* F^* H^* & K_0^* F^{*2} H^* & \cdots \\ HK_0 & H\Delta H^* & H\Delta F^* H^* & H\Delta F^{*2} H^* & \\ HFK_0 & HF\Delta H^* & HF\Delta F^* H^* & HF\Delta F^{*2} H^* & \\ HF^2K_0 & HF^2\Delta H^* & HF^2\Delta F^* H^* & HF^2\Delta F^{*2} H^* & \\ \vdots & & & & \ddots \end{bmatrix}.$$

There is clearly a significant redundancy in the elements of $R_Y - Z^p R_Y \left[Z^p\right]^*$, since the second and later rows differ only by multiples of F from the rows above. One suspects that the block displacement rank is low, and this can be verified by going through the first few (in fact, two) steps of Schur reduction. Let us begin with the Schur complement of the $(0,0)$ block entry of ∇R_Y, which is

$$\nabla R_Y - \begin{bmatrix} R_{e,0} \\ HK_0 \\ HFK_0 \\ \vdots \end{bmatrix} R_{e,0}^{-1} \begin{bmatrix} R_{e,0} & K_0^* H^* & K_0^* F^* H^* & \ldots \end{bmatrix} =$$

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & H\delta P_1 H^* & H\delta P_1 F^* H^* & H\delta P_1 F^{*2} H^* & \\ \mathbf{0} & HF\delta P_1 H^* & HF\delta P_1 F^* H^* & HF\delta P_1 F^{*2} H^* & \\ \mathbf{0} & HF^2\delta P_1 H^* & HF^2\delta P_1 F^* H^* & HF^2\delta P_1 F^{*2} H^* & \\ \vdots & & & & \ddots \end{bmatrix} =$$

$$\begin{bmatrix} \mathbf{0} \\ H \\ HF \\ \vdots \end{bmatrix} \delta P_1 \begin{bmatrix} \mathbf{0} & H^* & F^* H^* & \ldots \end{bmatrix} ,$$

where we used the following relation (recalling the Riccati recursion (9.3))

$$H\Delta H^* - H\bar{K}_{p,0}\bar{K}_{p,0}^* H^* = H(\Pi_1 - \Pi_0 - \bar{K}_{p,0}\bar{K}_{p,0}^*)H^* = H(P_1 - P_0)H^* = H\delta P_1 H^*.$$

It now follows easily that the the displacement ∇R_Y has block rank 2,

$$\nabla R_Y = \begin{bmatrix} R_{e,0} \\ HK_0 \\ HFK_0 \\ \vdots \end{bmatrix} R_{e,0}^{-1} \begin{bmatrix} R_{e,0} \\ HK_0 \\ HFK_0 \\ \vdots \end{bmatrix}^* + \begin{bmatrix} \mathbf{0} \\ H \\ HF \\ \vdots \end{bmatrix} \delta P_1 \begin{bmatrix} \mathbf{0} \\ H \\ HF \\ \vdots \end{bmatrix}^* .$$

To find a generator for R_Y, we factor $\delta P_1 \equiv P_1 - P_0 = L_0 S L_0^*$, where L_0 is $n \times \alpha$ and S is the $\alpha \times \alpha$ signature matrix of $(P_1 - P_0)$. We can then write

$$\nabla R_Y = R_Y - Z^p R_Y [Z^p]^* = \mathcal{G} J \mathcal{G}^*,$$

where

$$J = \begin{bmatrix} I_p & \mathbf{0} \\ \mathbf{0} & S \end{bmatrix} \quad \text{and} \quad \mathcal{G} = \begin{bmatrix} R_{e,0}^{1/2} & \mathbf{0} \\ H\bar{K}_{p,0} & HL_0 \\ HF\bar{K}_{p,0} & HFL_0 \\ \vdots & \vdots \end{bmatrix} .$$

This establishes the fact that R_Y is indeed a structured matrix as defined in (9.7a), and hence we can compute its Cholesky factor via the array form of the generalized Schur algorithm.

Note that the rows of the generator matrix $\mathcal{G}$ are closely related: going from one row to another (except for the first row) just changes the power of the F matrix. This is a consequence of the underlying state-space model for the covariance matrix R_Y. We now verify that because of this additional structure in the generator matrix, the generalized Schur algorithm (9.7b) collapses to the Chandrasekhar recursions.

The first step in the Schur algorithm involves multiplying by $\mathbf{\Theta}_0$, which is the identity matrix since the first block-row of $\mathcal{G}$ already has a $p \times \alpha$ zero block, and then shifting down the first block-column to get

$$\mathcal{G}_1 = \begin{bmatrix} R_{e,0}^{1/2} & HL_0 \\ H\bar{K}_{p,0} & HFL_0 \\ HF\bar{K}_{p,0} & HF^2L_0 \\ \vdots & \vdots \end{bmatrix}.$$

Let $\mathbf{\Theta}_1$ be a $J-$unitary matrix such that $\begin{bmatrix} R_{e,0}^{1/2} & HL_0 \end{bmatrix} \mathbf{\Theta}_1 = \begin{bmatrix} X & \mathbf{0} \end{bmatrix}$. Applying $\mathbf{\Theta}_1$ to the first two (block) rows of $\mathcal{G}_1$ (denoted by A) we obtain a (block-triangular) postarray of the form

$$A\mathbf{\Theta}_1 = \begin{bmatrix} R_{e,0}^{1/2} & HL_0 \\ H\bar{K}_{p,0} & HFL_0 \end{bmatrix} \mathbf{\Theta}_1 = \begin{bmatrix} X & \mathbf{0} \\ Y & Z \end{bmatrix},$$

where we can identify the unknowns $\{X, Y, Z\}$ in terms of known quantities. For this purpose, we compare entries on both sides of the equality $AJA^* = A\mathbf{\Theta}_1 J \mathbf{\Theta}_1^* A^*$ leading to $XX^* = R_{e,0} + HL_0SL_0^*H^* = R_{e,1}$. So we can choose $X = R_{e,1}^{1/2}$. Moreover, $YX^* = K_0 + FL_0SL_0^*H^* = K_1$ and, hence, we can identify $Y = K_1R_{e,i}^{-1/2} = \bar{K}_{p,1}$. Finally, $YY^* + ZSZ^* = K_0R_{e,0}^{-1}K_0^* + FL_0SL_0^*F^* = P_2 - P_1 \equiv L_1SL_1^*$, which shows that we can identify Z as L_1. We thus conclude that

$$\begin{bmatrix} R_{e,0}^{1/2} & HL_0 \\ \bar{K}_{p,0} & FL_0 \end{bmatrix} \mathbf{\Theta}_1 = \begin{bmatrix} R_{e,1}^{1/2} & \mathbf{0} \\ \bar{K}_{p,1} & L_1 \end{bmatrix}.$$

Therefore, $\mathcal{G}_1\mathbf{\Theta}_1$ is equal to (we now invoke the special structure of the rows of $\mathcal{G}_1$)

$$\mathcal{G}_1\mathbf{\Theta}_1 = \begin{bmatrix} R_{e,1}^{1/2} & \mathbf{0} \\ H\bar{K}_{p,1} & HL_1 \\ HF\bar{K}_{p,1} & HFL_1 \\ \vdots & \vdots \end{bmatrix}.$$

Next we shift down the first p columns to get

$$\mathcal{G}_2 = \begin{bmatrix} R_{e,1}^{1/2} & HL_1 \\ H\bar{K}_{p,1} & HFL_1 \\ HF\bar{K}_{p,1} & HF^2L_1 \\ \vdots & \vdots \end{bmatrix},$$

choose a $J-$unitary matrix $\mathbf{\Theta}_2$, shift down, form $\mathbf{\Theta}_3$, and so on. We see that because of the special state-space structure of the elements of the generator of $\mathcal{R}$, there is again a significant redundancy in the factorization arrays: the equality of the first two nonzero rows tells enough to fill out all other rows. So the basic recursion is just the following, which coincides with the array form (9.6) of the Chandrasekhar recursions [122],

$$\begin{bmatrix} R_{e,i}^{1/2} & HL_i \\ \bar{K}_{p,i} & FL_i \end{bmatrix} \mathbf{\Theta}_{i+1} = \begin{bmatrix} R_{e,i+1}^{1/2} & \mathbf{0} \\ \bar{K}_{p,i+1} & L_{i+1} \end{bmatrix}, \tag{9.8}$$

where $\mathbf{\Theta}_{i+1}$ is *any* $J = (I \oplus S)-$unitary matrix that introduces the block zero entry on the right-hand side, and

$$P_{i+2} - P_{i+1} = L_{i+1} S L_{i+1}^*.$$

It is satisfying to see how the Chandrasekhar equations, which were instrumental in the initiation of the displacement theory [110, 111], turn out to be a special case of the resulting theory. However, the displacement structure framework also provides further insights into the nature of the Chandrasekhar recursions. Thus Sayed and Kailath [170, 171, 178] showed that the Chandrasekhar recursions can be extended to time-variant state-space models that exhibit a certain structure in their time-variation. Such models often arise in adaptive filtering.

To clarify this point, we first note that the computational advantage of the Chandrasekhar recursions (9.8) stems from the fact that it propagates a low rank factor L_i instead of P_{i+1}, where $P_{i+1} - P_i = L_i S_i L_i^*$. A direct generalization would be to consider differences of the form $P_{i+1} - \Psi_i P_i \Psi_i^*$, where Ψ_i are convenient time-variant matrices that also result in a low rank difference, say of rank α. That is,

$$P_{i+1} - \Psi_i P_i \Psi_i^* \;\equiv\; L_i S_i L_i^*,$$

for some $n \times \alpha$ matrix L_i (it also follows that for the special time-variant models to be discussed ahead we have $S_i = S, \;\; \forall \, i$).

We consider a state-space model with time-variant matrices $\{F_i, G_i, H_i\}$ as in (9.1a), and we shall say that it is a *structured* time-variant model if there exist $n \times n$ matrices Ψ_i such that F_i, G_i, and H_i vary according to the following rules:

$$\text{(9.9a)} \qquad H_i = H_{i+1}\Psi_i, \quad F_{i+1}\Psi_i = \Psi_{i+1}F_i, \quad G_{i+1} = \Psi_{i+1}G_i \;\; .$$

It is clear that constant-parameter systems satisfy (9.9a) with $\Psi_i = I$. We shall assume that the covariance matrices R_i and Q_i are time-invariant whereas F_i, H_i, G_i vary in time according to (9.9a) (the restrictions on $\{R_i, Q_i\}$ can be relaxed as discussed in [171, 178]).

The point is that the conditions specified in (9.9a) guarantee that the covariance matrix R_Y of the output process $\{\mathbf{y}_i\}$ will still have a time-invariant displacement structure of the form $R_Y - Z^p R_Y [Z^p]^* = \mathcal{G} J \mathcal{G}^*$, and, consequently, its Cholesky factorization can still be carried out via the same generalized Schur algorithm. Thus, following the same reasoning as before, we can easily verify that for structured time-variant models as in (9.9a) we get

$$R_Y - Z^p R_Y [Z^p]^* = \begin{bmatrix} R_{e,0}^{1/2} & \mathbf{0} \\ H_1 \bar{K}_{p,0} & H_1 L_0 \\ H_2 F^{[1]} \bar{K}_{p,0} & H_2 F^{[1]} L_0 \\ H_3 F^{[2]} \bar{K}_{p,0} & H_3 F^{[2]} L_0 \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & S \end{bmatrix} \begin{bmatrix} R_{e,0}^{1/2} & \mathbf{0} \\ H_1 \bar{K}_{p,0} & H_1 L_0 \\ H_2 F^{[1]} \bar{K}_{p,0} & H_2 F^{[1]} L_0 \\ H_3 F^{[2]} \bar{K}_{p,0} & H_3 F^{[2]} L_0 \\ \vdots & \vdots \end{bmatrix}^* ,$$

where we defined $F^{[i]} = F_i F_{i-1} \ldots F_1, \quad F^{[0]} = I$, and L_0 and S are defined via the (nonunique) factorization $P_1 - \Psi_0 P_0 \Psi_0^* = L_0 S L_0^*$. Applying the generalized Schur algorithm to the above generator we readily verify that it collapses to the following extended Chandrasekhar recursions [170, 171, 178]:

$$\text{(9.9b)} \qquad \begin{bmatrix} R_{e,i}^{1/2} & H_{i+1} L_i \\ \Psi_{i+1} \bar{K}_{p,i} & F_{i+1} L_i \end{bmatrix} \mathbf{\Theta}_{i+1} = \begin{bmatrix} R_{e,i+1}^{1/2} & \mathbf{0} \\ \bar{K}_{p,i+1} & L_{i+1} \end{bmatrix} .$$

9.3. An Application: Fast RLS Adaptive Filtering. An important application of the extended Chandrasekhar recursions arises in the much-studied recursive least-squares problem of adaptive filtering (see, *e.g.*, [96, 160]). The presentation given here is a brief exposition of a state-space approach to adaptive RLS filtering that was recently proposed by Sayed and Kailath [171, 177, 179]. It shows how to derive the different versions of RLS adaptive schemes as special cases of a unifying estimation theory. Here we only focus on the fast RLS version; for other versions such as the QR algorithms and the lattice algorithms the interested reader may consult [179].

A basic problem in adaptive filtering reads as follows: given pairs of data points, $\{\mathbf{u}_i, d(i)\},\ \ i = 0, 1, \ldots, N$, where $\mathbf{u}_i$ is a $1 \times M$ row vector that consists of the values of M input channels at time i,

$$\mathbf{u}_i = \begin{bmatrix} u_1(i) & u_2(i) & \ldots & u_M(i) \end{bmatrix}, \tag{9.10a}$$

($d(i)$ and $u_j(i),\ j = 1, \ldots, M$, are assumed scalar for simplicity), we are required to determine the linear least-squares estimate of an $M \times 1$ column vector of unknown tap weights $\mathbf{w} = \begin{bmatrix} w_1 & w_2 & \ldots & w_M \end{bmatrix}^{\mathbf{T}}$, so as to minimize the exponentially weighted error sum

$$\mathcal{E} = (\mathbf{w} - \bar{\mathbf{w}})^* \left[\lambda^{-(N+1)}\Pi_0\right]^{-1} (\mathbf{w} - \bar{\mathbf{w}}) + \sum_{i=0}^{N} \lambda^{N-i} |d(i) - \mathbf{u}_i \mathbf{w}|^2 , \tag{9.10b}$$

where Π_0 is a given positive-definite matrix and the parameter λ, $0 \ll \lambda \leq 1$, is the so-called forgetting factor, since past inputs are (exponentially) weighted less than the more recent values. We can rewrite the expression for $\mathcal{E}$ as follows

$$\mathcal{E} = \lambda^N \left[(\mathbf{w} - \bar{\mathbf{w}})^* \left[\lambda^{-1}\Pi_0\right]^{-1} (\mathbf{w} - \bar{\mathbf{w}}) + \sum_{i=0}^{N} \left| \frac{d(i)}{(\sqrt{\lambda})^i} - \frac{\mathbf{u}_i \mathbf{w}}{(\sqrt{\lambda})^i} \right|^2 \right],$$

which shows that minimizing $\mathcal{E}$ is equivalent to the following minimization problem:

$$\min_{\mathbf{x}_0 = \mathbf{w}} \left[(\mathbf{x} - \bar{\mathbf{x}}_0)^* \left[\lambda^{-1}\Pi_0\right]^{-1} (\mathbf{x}_0 - \bar{\mathbf{x}}_0) + \sum_{i=0}^{N} |y(i) - \mathbf{u}_i \mathbf{x}_i|^2 \right], \tag{9.10c}$$

where we have defined the normalized quantities $y(i)$ and $\mathbf{x}_i$ by

$$y(i) = \frac{d(i)}{(\sqrt{\lambda})^i}, \quad \mathbf{x}_i = \frac{\mathbf{w}}{(\sqrt{\lambda})^i}, \quad \mathbf{x}_0 = \mathbf{w}, \quad \bar{\mathbf{x}}_0 = \bar{\mathbf{w}} . \tag{9.10d}$$

The minimization in (9.10c) can be easily recast into a Kalman filtering or recursive state-space estimation problem [179] by considering the following $M-$dimensional state-space model

$$\begin{aligned} \mathbf{x}_{i+1} &= \lambda^{-1/2}\mathbf{x}_i, \quad \mathbf{x}_0 = \mathbf{w}, \ \ \bar{\mathbf{x}}_0 = \bar{\mathbf{w}}, \ \ \Pi_0 , \\ y(i) &= \mathbf{u}_i \mathbf{x}_i + v(i), \quad Ev(i)v^*(j) = \delta_{ij} . \end{aligned} \tag{9.11}$$

This state-space model has special structure: $F = \lambda^{-1/2} I, G = 0$, and $R = 1$ are constant, while $H_i = \mathbf{u}_i$ is not. We now further assume that the input channels $\{u_1(.), u_2(.), \ldots, u_M(.)\}$ exhibit shift structure: $u_j(i) = u_{j-1}(i-1)$. That is, if we denote the value of the first channel at time i by $u(i)$, then this corresponds to having an input row vector $\mathbf{u}_i$ of the form

$$\mathbf{u}_i = \left[\begin{array}{cccc} u(i) & u(i-1) & \ldots & u(i-M+1) \end{array} \right] . \tag{9.12a}$$

The shift structure in $\mathbf{u}_i$ suggests that we might be able to get fast RLS algorithms by using the extended Chandrasekhar recursions in place of the Riccati recursions. In fact this is true and many results in the literature can be obtained in a more transparent (square-root array) form, and many variations and extensions derived in this way [179]. Observe that the input data $\mathbf{u}_i$ has a shift structure, viz.,

$$\mathbf{u}_i = \mathbf{u}_{i+1} Z + u(i-M+1) \left[\begin{array}{cccc} 0 & \ldots & 0 & 1 \end{array} \right] , \tag{9.12b}$$

where Z is the lower triangular shift matrix. We can however, modify the auxiliary $M-$dimensional state-space model (9.11) of the previous section in order to better exploit the shift structure and obtain a simpler relation than (9.12b). This will allow us to reduce the computational complexity of the RLS algorithm from $O(M^2)$ down to $O(M)$ operations (multiplications and additions) per time-step.

For this purpose, we consider the following $(N+1)-$ (not M) dimensional state-space model

$$\begin{array}{l} \mathbf{x}_{i+1} = \lambda^{-1/2} \mathbf{x}_i, \quad \mathbf{x}_0 = \left[\begin{array}{c} \mathbf{w} \\ \mathbf{0} \end{array} \right] , \\ y(i) = \mathbf{h}_i \mathbf{x}_i + v(i), \quad E v(i) v^*(j) = \delta_{ij} , \end{array} \tag{9.12c}$$

where $\mathbf{x}_i$ is now an $(N+1) \times 1$ state-vector with trailing zeros (added for convenience), and

$$\mathbf{h}_i = \left[\begin{array}{ccccc} u(i) & u(i-1) & \ldots & u(0) & \mathbf{0}_{N-i} \end{array} \right]$$

is a $1 \times (N+1)$ row vector. An initial state covariance matrix (with trailing zeros) is assumed, viz.,

$$E(\mathbf{x}_0 - \bar{\mathbf{x}}_0)(\mathbf{x}_0 - \bar{\mathbf{x}}_0)^* = \left[\begin{array}{cc} \Pi_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{array} \right] \equiv \Pi_0 \oplus \mathbf{0} ,$$

where Π_0 is an $M \times M$ positive definite matrix. The associated gain vector $\bar{k}_{p,i} = k_i r_{e,i}^{-*/2}$ has trailing zeros, viz.,

$$\bar{\mathbf{k}}_{p,i} \equiv \left[\begin{array}{c} \bar{\mathbf{c}}_i \\ \mathbf{0} \end{array} \right] ,$$

and the Riccati recursion is given by

$$P_{i+1} = \lambda^{-1} \left[P_i - P_i \mathbf{h}_i^* r_{e,i}^{-1} \mathbf{h}_i P_i \right] .$$

The computational complexity of the resulting (RLS) algorithm is $O(M^2)$ operations (multiplications and additions) per time step.

However, though time-variant, the special structure of $\mathbf{h}_i$, viz., $\mathbf{h}_i = \mathbf{h}_{i+1} Z$, can be further exploited to reduce the operation count to $O(M)$. Observe that the above relation is simpler than (9.12b), and shows (along with $F_{i+1} Z = Z F_i$, since $F_i = \lambda^{-1/2} I$) that the state-space model (9.12c) is a special structured time-variant model as in (9.9a). The reduction in operation count can now be achieved by using a special case of the extended Chandrasekhar recursions (9.9b) with $\Psi_i = Z,\ F_i = \lambda^{-1/2}I$. To apply these recursions, we first introduce the (nonunique) factorization

$$L_0SL_0^* = P_1 - ZP_0Z^* = \lambda^{-1}\left(\begin{bmatrix} \Pi_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} - \bar{\mathbf{k}}_{p,0}\bar{\mathbf{k}}_{p,0}^*\right) - Z\begin{bmatrix} \Pi_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}Z^* ,$$

where L_0 and S are $(N+1)\times\alpha$ and $\alpha\times\alpha$ matrices, respectively. The factor L_0 is of the form

$$L_0 = \begin{bmatrix} \tilde{L}_0 \\ \mathbf{0} \end{bmatrix} ,$$

where $\tilde{L}_0$ is $(M+1)\times\alpha$. Let $\tilde{\mathbf{h}}_i$ be the row vector of the first $M+1$ coefficients of $\mathbf{h}_i$. Writing down the extended Chandrasekhar recursions (9.9b) we obtain

(9.12d) $$\begin{bmatrix} r_{e,i}^{1/2} & \tilde{\mathbf{h}}_{i+1}\tilde{L}_i \\ \begin{bmatrix} 0 \\ \bar{\mathbf{c}}_i \end{bmatrix} & \lambda^{-1/2}\tilde{L}_i \end{bmatrix}\boldsymbol{\Theta}_{i+1} = \begin{bmatrix} r_{e,i+1}^{1/2} & \mathbf{0} \\ \begin{bmatrix} \bar{\mathbf{c}}_{i+1} \\ 0 \end{bmatrix} & \tilde{L}_{i+1} \end{bmatrix} ,$$

where $\boldsymbol{\Theta}_{i+1}$ is *any* $J = (1\oplus S)-$unitary matrix that produces the zero entry on the right hand-side of the above expression. The computational complexity of each step is $O(\alpha M)$ where the value of α depends on the choice of Π_0. This recursion is a square-root version of fast RLS algorithms discussed in the literature [39, 53]. It was also derived by Houacine [102] and Slock [187] by using alternative state-space models (see [179, 178] for more discussion). Work continues by us and others on this active research area, see, e.g., [188].

10. Time-Variant Displacement Structure. The study of time-variant Chandrasekhar recursions was based on the equation

$$P_{i+1} - \Psi_iP_i\Psi_i^* \ = \ L_iS_iL_i^*.$$

This suggests that the notion of displacement structure can also be extended to the time-variant setting, as detailed in [171, 172, 174, 184]. We say that an $n\times n$ matrix $R(t)$ has a time-variant Toeplitz-like displacement structure if the matrix difference $\nabla R(t)$ defined by

$$\nabla R(t) = R(t) - F(t)R(t-\Delta)F^*(t) \ ,$$

has low rank, say $r(t)$ (usually $r(t)\ll n$), for some lower triangular $n\times n$ matrix $F(t)$ whose diagonal elements we shall denote by $\{f_i(t)\}_{i=0}^{n-1}$. The indices t and $(t-\Delta)$ denote two discrete-time instants that are Δ units apart. It follows from the low rank property that we can factor $\nabla R(t)$ and write

(10.1) $$R(t) - F(t)R(t-\Delta)F^*(t) = G(t)J(t)G^*(t) \ ,$$

where $G(t)$ is an $n\times r(t)$ generator matrix and $J(t)$ is an $r(t)\times r(t)$ signature matrix with as many $\pm1's$ as $\nabla R(t)$ has strictly positive or negative eigenvalues. The notation $I_{p(t)}$ refers to the $p(t)\times p(t)$ identity matrix,

$$J(t) = \begin{bmatrix} I_{p(t)} & \mathbf{0} \\ \mathbf{0} & -I_{q(t)} \end{bmatrix}, \quad r(t) = p(t) + q(t).$$

We shall outline the main features of the time-variant theory here. We shall only focus, for brevity, on positive-definite Hermitian matrices $R(t)$ although the theory is equally applicable to strongly regular and non-Hermitian time-variant matrices, as well. We assume from now on, and for simplicity of notation, that $\Delta = 1$.

The main question of interest here is the following: given the Cholesky factor of $R(t-1)$, and knowing that $R(t)$ satisfies a displacement equation of the form (10.1), how to determine efficiently and recursively the Cholesky factor of $R(t)$? Situations of this type often arise in adaptive filtering [184] and instrumental-variable methods [172].

Arguments analogous to what we have used in the time-invariant case allow us to extend the recursive Schur algorithm to the time-variant setting as well (see, e.g., [184, 174, 171]), thus leading to the following result (compare with (6.3a)). Let $\bar{L}(t)$ denote the (lower-triangular) Cholesky factor of $R(t)$, viz., $R(t) = \bar{L}(t)\bar{L}^*(t)$, and denote (the nonzero parts of) its columns by $\bar{l}_i(t)$. That is, $\bar{l}_i(t) = l_i(t)d_i^{-1/2}(t)$, where $l_i(t)$ and $d_i(t)$ denote the $(0,0)$ entry and the first column of the i^{th} Schur complement $R_i(t)$, respectively.

THEOREM 10.1 (A Generalized Time-Variant Schur Recursion in Array Form). *The Cholesky factor of a positive-definite Hermitian matrix $R(t)$ that satisfies the time-variant displacement equation (10.1) can be time-updated as follows: start with $F_0(t) = F(t)$, $G_0(t) = G(t)$, $\bar{L}_0(t-1) = \bar{L}(t-1)$, and repeat for $i = 0, 1, \ldots, n-1$:*

1. *At step i we have $F_i(t)$ and $G_i(t)$. Let $g_i(t)$ denote the first row of $G_i(t)$.*
2. *Choose a convenient $(1 \oplus J(t))-$unitary transformation $\mathbf{\Gamma}_i(t)$ that performs the rotation*

$$\begin{bmatrix} f_i(t)d_i^{-1/2}(t-1) & g_i(t) \end{bmatrix} \mathbf{\Gamma}_i(t) = \begin{bmatrix} d_i^{1/2}(t) & \mathbf{0} \end{bmatrix}.$$

3. *Applying $\mathbf{\Gamma}_i(t)$ to the prearray $\begin{bmatrix} F_i(t)\bar{l}_i(t-1) & G_i(t) \end{bmatrix}$ leads to*

$$\begin{bmatrix} F_i(t)\bar{l}_i(t-1) & G_i(t) \end{bmatrix} \mathbf{\Gamma}_i(t) = \begin{bmatrix} \bar{l}_i(t) & \begin{matrix} \mathbf{0} \\ G_{i+1}(t) \end{matrix} \end{bmatrix}, \tag{10.2}$$

where $F_{i+1}(t)$ and $\bar{L}_{i+1}(t-1)$ are the submatrices obtained by deleting the first row and column of each of $F_i(t)$ and $\bar{L}_i(t-1)$, respectively. Moreover, the matrix $G_{i+1}(t)$ that appears in the postarray is a generator matrix of the $(i+1)^{th}$ Schur complement $R_{i+1}(t)$. That is,

$$R_{i+1}(t) - F_{i+1}(t)R_{i+1}(t-1)F_{i+1}(t) = G_{i+1}(t)J(t)G_{i+1}^*(t).$$

Figure 10.1 depicts one step of the algorithm in Theorem 10.1. Each such step is characterized by a $(1 \oplus J(t))-$unitary transformation $\mathbf{\Gamma}_i(t)$, and a storage element Δ that stores the present value of $\bar{l}_i(t)$ to the next time instant. The generator matrix $G_i(t)$ and the column vector $F_i(t)\bar{l}_i(t-1)$ undergo the transformation $\bar{\mathbf{\Gamma}}_i(t)$ and yield the next generator $G_{i+1}(t)$, as well as the i^{th} column of the Cholesky factor, $\bar{l}_i(t)$.

More generally, the generator recursion (10.2) admits the form [174]

$$\begin{bmatrix} l_i(t) & \begin{matrix} \mathbf{0} \\ G_{i+1}(t) \end{matrix} \end{bmatrix} = \begin{bmatrix} F_i(t)l_i(t-1) & G_i(t) \end{bmatrix} \begin{bmatrix} f_i^*(t) & h_i^*(t)J(t) \\ J(t)g_i^*(t) & J(t)k_i^*(t)J(t) \end{bmatrix},$$

where $h_i(t)$ and $k_i(t)$ are arbitrary $r(t) \times 1$ and $r(t) \times r(t)$ matrices, respectively, chosen so as to satisfy the time-variant embedding (or dilation) relation (compare with (7.2b))

$$\begin{bmatrix} f_i(t) & g_i(t) \\ h_i(t) & k_i(t) \end{bmatrix} \begin{bmatrix} d_i(t-1) & \mathbf{0} \\ \mathbf{0} & J(t) \end{bmatrix} \begin{bmatrix} f_i(t) & g_i(t) \\ h_i(t) & k_i(t) \end{bmatrix}^* = \begin{bmatrix} d_i(t) & \mathbf{0} \\ \mathbf{0} & J(t) \end{bmatrix}.$$

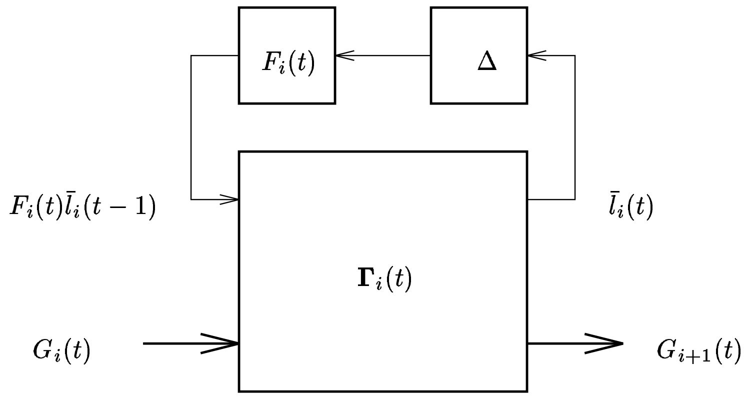

FIG. 10.1. *One step of the generalized time-variant Schur algorithm.*

This again defines a time-variant first-order system in state-space form with inputs from the left. Each such section can be further shown to exhibit certain blocking properties, which turn out to be equivalent to time-variant extensions of classical interpolation problems. For more details on these problems, and related applications to several matrix completion problems, the interested reader may consult [56, 172, 173, 174].

11. Concluding Remarks. Though a wide range of results and applications has already been addressed in this long paper, there are still several unmentioned results, and also areas where further research is possible. We give a brief outline of a few of these items here; there are indeed many others, some big, some small.

11.1. Block Schur Complementation and Look-Ahead Algorithms. A standing assumption in all the preceding has been that the matrices R are strongly regular, *i.e.,* all their leading minors are nonzero. This allowed the derivation of a recursive algorithm for the update of generator matrices of successive Schur complements. 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 a well-conditioned leading minor of appropriate dimensions, in order to proceed with a block Schur reduction step. This problem was addressed in [153] for the special class of quasi-Toeplitz matrices, where it was further shown that the block Schur complementation step can be performed via scalar operations.

More general (block or look-ahead) recursions for the class of Toeplitz-like matrices were presented in [171, 180], which employed a generalized block Schur algorithm of the form

$$\begin{bmatrix} \mathbf{0}_{\eta_i} \\ G_{i+1} \end{bmatrix} = \left\{ G_i + (\tau_i^* F_i - I_{n-\alpha_i}) L_i D_i^{-1} (I_{\eta_i} - \tau_i^* \hat{F}_i)^{-1} \hat{G}_i \right\} \Theta_i, \quad G_0 = G, \quad F_0 = F,$$

where Θ_i is an arbitrary $J-$unitary matrix, τ_i is an arbitrary unit-modulus scalar, $\eta_i \times \eta_i$ is the size of the (desired) leading block (or minor) D_i, L_i is comprised of the first η_i columns of the Schur complement at the i^{th} step, and $\hat{G}_i$ is comprised of the top η_i rows of G_i at the i^{th} step. This block recursion is applicable whether or not η_i stands for the size of the smallest nonsingular minor. The η_i can also designate the size of any desirable blocks, such as blocks that are numerically well-conditioned.

The use of block Schur complementation, or block pivoting, has recently been studied by several authors trying to devise effective numerical algorithms for various classes of (non-positive definite) structured matrices. An early paper was the one of Chan and Hansen [44]. Among many others we mention Gutknecht [95] and Freund [77], which give extensive references. Here we only wish to suggest that the explicit formulas of [171, Ch. 7] and [180], mentioned above, could provide new fast look-ahead algorithms for matrices with very general displacement structure.

We should note that apart from numerical possibilities, these fast block-Schur complementation algorithms have several theoretically interesting features as well. For example, the explicit formulas for the block diagonal matrix in the generalized LDU factorization can give simple rules for computing the inertia of general structured matrices. In particular, Pal [152] (see also [9]) showed how to obtain certain often-quoted rules of Iohvidov [105] for specifying the inertia of Toeplitz and Hankel matrices.

11.2. The Schur Theory for Matrix-Valued Meromorphic Functions. To avoid confusion, we may also note that what we have called block Schur complementation algorithms are different from the (block) Schur algorithms proposed for Toeplitz matrices with block (not necessarily Toeplitz) entries – see, e.g., [64]. Carrying out the block operations on the block entries requires forming several matrix inverses, and can be expensive. Fedcina [73] and Dewilde and Dym [67] introduced *tangential* versions in which the operations are performed in only one "direction" at a time, for example, row after row. However, by using elementary operations as described in Section 4.4, one can further reduce the operations in each direction to a set of elementary scalar operations. This scalarization procedure is widely used in the square-root (Riccati and Chandrasekhar) algorithms of Kalman filter theory described in Section 9.

The paper [2] describes this in detail, and moreover also discusses what needs to be done to extend the classical Schur algorithm, which dealt with H^∞−functions (bounded and analytic in the unit disc), to meromorphic functions. This was apparently first done by Chamfy [40] in the scalar case. Reference [1] shows how to extend the transmission line model of Section 3.2 to the scalar meromorphic case. The general case was treated in Section 6.

11.3. Doubling or Divide-and-Conquer Algorithms. The recursive doubling or divide-and-conquer ideas used to develop the FFT have been exploited to obtain asymptotically fast $O(n \log^2 n)$ algorithms for Toeplitz systems. The first such algorithm was given by Brent et al. [32]. Soon after, Bitmead and Anderson [26] and Morf [145] independently proved and used the property that displacement structure is preserved by Schur complementation to obtain similar results. For general Toeplitz-like matrices, however, the coefficient in their proposed $O(n \log^2 n)$ algorithms turned out to be extremely large. Later several other authors used an approach based on a combination of the Schur and Levinson algorithms to obtain better results; in particular Ammar and Gragg [12] made a detailed study and claimed an operation count of $8n \log^2 n$ flops for solving Toeplitz linear systems. With this count, the new algorithm (called superfast in [12]) is faster than the one based on the Levinson algorithm whenever $n > 256$. A purely Schur-based approach to Toeplitz-like matrices (e.g., those encountered in finding least-squares solutions of overdetermined linear equations) is described in [51]. If one recalls the fact that the Schur algorithm is closely related to transmission lines (cf. Section 3.2), then it may not be surprising that similar Schur-based methods for Toeplitz matrices were also discovered in the geophysics literature (see the references in [51]).

11.4. Iterative Methods. The existing methods for the solution of linear systems of equations of the form $Ax = b$ can be classified into two main categories: the so-called *direct* methods and the alternative *iterative* methods. A direct method or algorithm is primarily concerned in first obtaining the triangular or QR factors of A, and then reducing the original equations $Ax = b$ to an equivalent triangular system of equations, whose solution may be obtained, for instance, via backsubstitution [94]. In its most primitive form, the direct method may not involve more than a standard Gaussian elimination procedure with a computational complexity of the order of n^3, written $O(n^3)$, where $n \times n$ denotes the size of A. Of course, when the matrix has special structure (Toeplitz, Hankel, displacement) the computational complexity can be reduced, e.g., to $O(n^2)$ or $O(n \log^2 n)$, as detailed in earlier sections of this paper (see also [31, 94, 137, 193, 104]).

An iterative method, on the other hand, starts with an initial guess for the solution x, say x_0, and generates a sequence of approximate solutions, $\{x_k\}_{k\geq 1}$. The matrix A itself is involved in the iterations via matrix-vector products, and the major concern here is the speed of convergence of the iterations. To clarify this point, we note that we can rewrite the equation $Ax = b$ in the equivalent form

$$Cx = (C - A)x + b \ ,$$

for an arbitrary invertible matrix C. This suggests the following iterative scheme (see, e.g., [94]),

$$Cx_{i+1} = (C - A)x_i + b, \quad x_0 = \text{ initial guess.} \tag{11.1}$$

The convergence of (11.1) is clearly dependent on the spectrum of the matrix $I - C^{-1}A$. The usefulness of (11.1) from a practical point of view is very dependent on the choice for C. Extensive results in the literature are available for the special choice of circulant preconditioners C (see, e.g., [41, 43, 190], which allow the use of the fast Fourier transform (FFT) technique to carry out the computations in a numerically efficient and parallelizable manner. An interesting demonstration to this effect is the recent work by Plemmons and co-workers [42, 148], which shows how to exploit this fact, and displacement structure, in several applications. V. Pan has taken a different approach by studying the application of Newton's iteration to structured matrices [157], and also the use of parallel computation to obtain $O(\log^2 n)$ iterative solutions [23, 24, 158].

11.5. Numerical Issues. An important issue is the analysis of the numerical properties of the generalized Schur algorithms. Despite many efforts, definitive results even for Toeplitz matrices are not readily available - see the discussion in [87]. It is shown in [87] that for many structured classes such as Vandermonde-like, Chebyshev-Vandermonde-like, polynomial Vandermonde-like, Cauchy-like matrices, and in fact for any kind of displacement structure as in (7.23a) with a diagonal F, partial pivoting technique (suggested by Heinig [97]) can be incorporated into the generalized Schur algorithm [120, 175], thus giving (see Sec. 4.1) fast implementations of Gaussian elimination with partial pivoting. The point is that for the above classes of structured matrices, row interchanges do not affect the structure, e.g., a Cauchy matrix remains Cauchy (a fact first remarked and exploited by Heinig [97]). To overcome the fact that row interchanges can destroy Toeplitz structure, one can first transform it to Cauchy form, then solve the problem and finally return to the original matrix.

In fact, one can also transform Toeplitz-like, Hankel-like, Toeplitz-plus-Hankel-like, Vandermonde-like, Chebyshev-Vandermonde-like structured matrices, and probably others, into Cauchy-like matrices, essentially using only Fast Fourier, Fast Cosine or Fast Sine transforms of the columns of the generator matrices (see [24, 87, 119]).

11.6. Continuous-Time Results. The identification of displacement structure was first made in the context of integral operators and integral equations. However, attention shifted to matrices and linear equations around 1980, just as the relevance of Schur's work became clear. It seems the time may be ripe to return to the study of continuous-time problems using the insights gained from the long study of discrete-time matrix problems.

Acknowledgments

Apart from the many close colleagues and other investigators who have contributed to the development of the concept of displacement structure, we would like to thank Dr. V. Olshevsky, Dr. Y. C. Pati, and Prof. T. Constantinescu for careful reading of this paper and several useful suggestions.

REFERENCES

[1] R. Ackner, H. Lev-Ari, and T. Kailath, *Transmission-line models for the modified Schur algorithm*, IEEE Transactions on Circuits and Systems, 39 (1992), pp. 290–296.

[2] ———, *The Schur algorithm for matrix-valued meromorphic functions*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 140–150.

[3] N. I. Akhiezer, *The Classical Moment Problem and Some Related Questions in Analysis*, Hafner Publishing Company, NY, 1965. In Russian, 1961.

[4] N. Al-Dhahir, *Optimized-Transmitter Reduced-Complexity MMSE-DFE Under Finite-Length Constraints*, PhD thesis, Stanford University, Stanford, CA, June 1994.

[5] N. Al-Dhahir and J. Cioffi, *MMSE decision feedback equalizers: Finite length results*, IEEE Transactions on Information Theory, to appear.

[6] D. Alpay, P. Dewilde, and H. Dym, *Lossless inverse scattering and reproducing kernels for upper triangular operators*, Operator Theory: Advances and Applications, 47 (1990), pp. 61–135.

[7] D. Alpay and H. Dym, *On applications of reproducing kernel spaces to the Schur algorithm and rational J–unitary factorization*, Operator Theory: Advances and Applications, 18 (1986), pp. 89–159.

[8] ———, *Structured invariant spaces of vector valued functions, sesquilinear forms, and a generalization of the Iohvidov laws*, Linear Algebra and Its Applications, 137–138 (1990), pp. 413–451.

[9] ———, *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.

[10] ———, *On a new class of reproducing kernel spaces and a new generalization of Iohvidov laws*, Linear Algebra and Its Applications, 178 (1992), pp. 109–183.

[11] G. Ammar and P. D. Gader, *A variant of the Gohberg-Semencul formula involving circulant matrices*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 534–540.

[12] G. S. Ammar and W. B. Gragg, *Superfast solution of real positive definite Toeplitz systems*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 61–76.

[13] A. C. Antoulas and B. D. O. Anderson, *On the scalar rational interpolation problem*, IMA Journal of Mathematical Control and Information, 3 (1986), pp. 61–88.

[14] A. C. Antoulas, J. A. Ball, J. Kang, and J. C. Willems, *On the solution of the minimal rational interpolation problem*, Linear Algebra and its Applications, 137/138 (1990), pp. 511–573.

[15] A. C. Antoulas and J. C. Willems, *Minimal rational interpolation and Prony's method*, in Analysis and Optimization of Systems, A. Bensoussan and J. L. Lions, eds., vol. 144 of Lecture Notes in Control and Information Sciences, Springer-Verlag, 1990, pp. 297–306.

[16] N. Aronszajn, *Theory of reproducing kernels*, Trans. Amer. Math. Soc., 63 (1950), pp. 337–404.

[17] A.W.Bojanczyk, R.P.Brent, and F. de Hoog, *QR factorization of Toeplitz matrices*, Numerische Mathematik, 49 (1986), pp. 81–94.
[18] M. Bakonyi and T. Constantinescu, *Schur's Algorithm and Several Applications*, vol. 261 of Pitman Research Notes in Mathematics, Longman Scientific and Technical, 1992.
[19] J. A. Ball, I. Gohberg, and M. A. Kaashoek, *Nevanlinna-Pick interpolation for time-varying input-output maps: The discrete case*, Operator Theory: Advances and Applications, 56 (1992), pp. 1–51. Birkhäuser Verlag Basel, ed. I. Gohberg.
[20] J. A. Ball, I. Gohberg, and L. Rodman, *Interpolation of Rational Matrix Functions*, vol. 45 of Operator Theory: Advances and Applications, Birkhäuser, Basel, 1990.
[21] ———, *Two-sided Lagrange-Sylvester interpolation problems for rational matrix functions*, in Proceedings of Symposia in Pure Mathematics, Part I, vol. 51, American Mathematical Society, (1990), pp. 17–83.
[22] E. H. Bareiss, *Numerical solution of linear equations with Toeplitz and vector Toeplitz matrices*, Numer. Math., 13 (1969), pp. 404–424.
[23] D. Bini and V. Pan, *Improved parallel computations with Toeplitz-like and Hankel-like matrices*, Linear Algebra and Its Applications, 188/189 (1993), pp. 3–29.
[24] ———, *Numerical and Algebraic Computations with Matrices and Polynomials*, Birkhauser, Boston, MA, 1994.
[25] Y. Bistritz, H. Lev-Ari, and T. Kailath, *Immitance-type three-term Schur and Levinson recursions for quasi-Toeplitz complex Hermitian matrices*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 497–520.
[26] G. Bitmead and B. D. O. Anderson, *Asymptotically fast solution of Toeplitz and related systems of linear equations*, Linear Algebra and Its Applications, 34 (1980), pp. 103–116.
[27] A. W. Bojanczyk and A. O. Steinhardt, *Stabilized Hyperbolic Householder transformations*, IEEE Transactions on Acoustics, Speech, and Signal Processing, 37 (1989), pp. 1286–1288.
[28] T. Boros, A. H. Sayed, and T. Kailath, *A recursive method for solving unconstrained tangential interpolation problems*, submitted for publication.
[29] ———, *Structured matrices and unconstrained rational interpolation*, Linear Algebra and Its Applications, 203–204 (1994), pp. 155–188.
[30] E. Bozzo and C. Di Fiore, *On the use of certain matrix algebras associated with discrete trigonometric transforms in matrix displacement decomposition,* SIAM J. Matrix Anal. Appl., 16 (1995).
[31] R. P. Brent, *Parallel algorithms for Toeplitz systems*, in Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms, G. H. Golub and P. V. Dooren, eds., Springer-Verlag, 1991, pp. 75–92.
[32] R. P. Brent, F. G. Gustavson, and D. Y. Y. Yun, *Fast solution of Toeplitz systems of equations and computation of Padé approximants*, J. Algorithms, 1 (1980), pp. 259–295.
[33] A. Bruckstein, T. K. Citron, and T. Kailath, *Inverse scattering and minimal partial realization*, Int. J. Control, 48 (1988), pp. 1537–1550.
[34] A. Bruckstein and T. Kailath, *Inverse scattering for discrete transmission-line models*, SIAM Review, 29 (1987), pp. 359–389.
[35] ———, *An inverse scattering framework for several problems in signal processing*, IEEE ASSP Magazine, (1987), pp. 6–20.
[36] ———, *On discrete Schrodinger equations and their two component wave-equation equivalents*, J. Math. Physics, 28 (1987), pp. 2914–2924.
[37] O. Brune, *Synthesis of a finite two-terminal network whose driving point impedance is specified as a function of frequency*, J. Math. Phys., 10 (1931), pp. 191–236.
[38] K. B. Bube and R. Burridge, *The one-dimensional inverse problem of reflection seismology*, SIAM Review, 25 (1983), pp. 497–559.
[39] G. Carayannis, D. Manolakis, and N. Kalouptsidis, *A fast sequential algorithm for least squares filtering and prediction*, IEEE Transactions on Acoustics, Speech, and Signal Processing, 31 (1983), pp. 1394–1402.
[40] C. Chamfy, *Functions méromorphes dans le cercle-unité et leurs series de Taylor*, Ann. Inst. Fourier, 8 (1958), pp. 211–251.
[41] R. Chan, *Circulant preconditioners for Hermitian Toeplitz systems*, SIAM J. Matrix Anal. Applic., 10 (1989), pp. 503–506.
[42] R. H. Chan, J. G. Nagy, and R. J. Plemmons, *Circulant preconditioned Toeplitz least squares iterations*, SIAM J. Martix. Anal. Applic., 15 (1994), pp. 80–97.
[43] T. F. Chan, *An optimal circulant preconditioner for Toeplitz systems*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 766–771.
[44] 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.
[45] S. Chandrasekhar, *On the radiative equilibrium of a stellar atmosphere, Pt XXI*, Astrophys. J., 106 (1947), pp. 152–216. Pt XXII, vol. 107, pp. 48–72, 1948.
[46] Y. M. Cho, G. Xu, and T. Kailath, *Fast recursive identification of state-space models via exploitation of displacement structure*, Automatica, 30 (1994), pp. 45–59.
[47] Y. M. Cho, G. Xu, and T. Kailath, *Fast identification of state-space models via exploitation of displacement structure*, IEEE Transactions on Automatic Control, 39 (1994), pp. 2004–2017.
[48] J. Chun, *Fast Array Algorithms for Structured Matrices*, PhD thesis, Stanford University, Stanford, CA, 1989.
[49] J. Chun and T. Kailath, *A constructive proof of the Gohberg-Semencul formula*, Linear Algebra and Its Applications, 121 (1989), pp. 475–489.
[50] ———, *Displacement structure for Hankel, Vandermonde, and related (derived) matrices*, Linear Algebra and Its Applications, 151 (1991), pp. 199–227.
[51] ———, *Divide and conquer solutions of least squares problems for matrices with displacement structure*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 128–145.
[52] J. Chun, T. Kailath, and H. Lev-Ari, *Fast parallel algorithms for QR and triangular factorization*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. 899–913.
[53] J. Cioffi and T. Kailath, *Fast recursive-least-squares transversal filters for adaptive filtering*, IEEE Transactions on Acoustics, Speech and Signal Processing, ASSP-32 (1984), pp. 304–337.
[54] T. Citron, *Algorithms and Architectures for Error Correcting Codes*, PhD thesis, Stanford University, Stanford, CA, 1986.
[55] T. Citron, A. Bruckstein, and T. Kailath, *An inverse scattering approach to the partial realization problem*, in Proc. 23^{rd} Conference on Decision and Control, Las Vegas, 1984.
[56] T. Constantinescu, A. H. Sayed, and T. Kailath, *Displacement structure and completion problems*, SIAM J. Matrix Anal. Appl., to appear.
[57] T. Constantinescu, A. H. Sayed, and T. Kailath, *Displacement structure and maximum entropy*, submitted for publication.
[58] ———, *A recursive Schur-based approach to the four-block problem*, IEEE Transactions on Automatic Control, 39 (1994), pp. 1476–1481.
[59] G. Cybenko, *A general orthogonalization technique with applications to time series analysis and signal processing*, Math. Comp., 40 (1983), pp. 323–336.
[60] ———, *Fast Toeplitz orthogonalization using inner products*, SIAM J. Scientific Comp., (1987), pp. 734–740.
[61] G. Cybenko and M. Berry, *Hyperbolic Householder algorithms for factoring structured matrices*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 499–520.
[62] L. de Branges, *Hilbert Spaces of Entire Functions*, Prentice-Hall, Englewood Cliffs, NJ, 1968.
[63] J. M. Delosme, *Algorithms and Implementations for Linear Least-Squares Estimation*, PhD thesis, Stanford University, Stanford, CA, 1982.
[64] P. Delsarte, Y. Genin, and Y. Kamp, *The Nevanlinna-Pick problem for matrix-valued functions*, SIAM J. Appl. Math., 36 (1979), pp. 47–61.
[65] ———, *Equivalence classes of Hermitian matrices and their Schur parametrization*, SIAM J. Alg. Disc. Math., 3 (1983), pp. 279–298.
[66] P. Delsarte, Y. Genin, and Y. Kamp, *A polynomial approach to the generalized Levinson algorithm based on the Toeplitz distance*, IEEE Transactions on Information Theory, 29 (1983), pp. 268–278.
[67] P. Dewilde and H. Dym, *Lossless chain scattering matrices and optimum linear prediction: The vector case*, Circuit Theory and Applications, 9 (1981), pp. 135–175.
[68] ———, *Interpolation for upper triangular operators*, Operator Theory: Advances and Applications, 56 (1992), pp. 153–260. Birkhäuser Verlag Basel, ed. I. Gohberg.
[69] P. Dewilde, J. T. Fokkema, and I. Widya, *Inverse scattering and linear prediction: The continuous-time case*, Stochastic Systems, (1981), pp. 351–382.
[70] 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 (1978), pp. 663–675.
[71] H. Dym, *J-Contractive Matrix Functions, Reproducing Kernel Hilbert Spaces and Interpolation*, vol. 71, Conference Board of the Mathematical Sciences, American Mathematical Society, Providence, Rhode Island, 1989.
[72] ———, *On reproducing kernel spaces, J-unitary matrix functions, interpolation and displacement rank*, Operator Theory: Advances and Applications, 41 (1989), pp. 173–239.
[73] P. I. Fedčina, *Descriptions of solutions of the tangential Nevanlinna-Pick problem*, Doklady

Akad. Nauk. Arm., 60 (1975), pp. 37–42.

[74] M. Fiedler, *Hankel and Loewner matrices*, Linear Algebra and Its Applications, 58 (1985), pp. 75–95.

[75] C. Di Fiore and P. Zellini, *Matrix decompositions using displacement rank and classes of commutative matrix algebras*, Linear Algebra and Its Applications, to appear.

[76] C. Foias and A. E. Frazho, *The Commutant Lifting Approach to Interpolation Problems*, vol. 44 of Operator Theory: Advances and Applications, Birkhäuser, Basel, 1990.

[77] R. W. Freund, *A look-ahead Bareiss algorithm for general Toeplitz matrices*, Numer. Math., to appear.

[78] B. Friedlander, *Scattering Theory and Linear Least Squares Estimation*, PhD thesis, Stanford University, Stanford, CA, August 1976.

[79] B. Friedlander, T. Kailath, M. Morf, and L. Ljung, *Extended Levinson and Chandrasekhar equations for general discrete-time linear estimation problems*, IEEE Transactions on Automatic Control, 23 (1978), pp. 653–659.

[80] B. Friedlander and M. Morf, *Efficient inversion formulas for sums of products of Toeplitz and Hankel matrices*, in Proceedings 18th Annual Allerton Conference on Communication, Control and Computing, October 1980, pp. 574–583.

[81] P. D. Gader, *Displacement operator based decompositions of matrices using circulants or other group matrices*, Linear Algebra and Its Applications, 139 (1990), pp. 111–131.

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

[83] 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 (1983), pp. 251–275.

[84] M. Gentleman, *Least squares computations by Givens transformations*, J. INst. Math. Appl., 12 (1973), pp. 329–336.

[85] L. Y. Geronimus, *Orthogonal Polynomials*, Transl. Consultants Bureau, NY, 1961. In Russian, 1958.

[86] I. Gohberg and I. Fel'dman, *Convolution Equations and Projection Methods for their Solutions*, vol. 41 of Translations of Mathematical Monographs, Amer. Math. Soc., 1974.

[87] I. Gohberg, T. Kailath, and V. Olshevsky, *Fast Gaussian elimination with partial pivoting for matrices with displacement structure*, Math. Comp., to appear.

[88] I. Gohberg and V. Olshevsky, *Circulants, displacements and decompositions of matrices*, Integral Equations and Operator Theory, 15 (1992), pp. 730–743.

[89] I. Gohberg and V. Olshevsky, *Complexity of multiplication with vectors for structured matrices,* Linear Algebra and Its Applications, 202 (1994), pp. 163–192.

[90] ———, *Fast inversion of Chebyshev-Vandermonde matrices*, Numer. Math., 67 (1994), pp. 71–92.

[91] ———, *Fast state space algorithms for matrix Nehari and Nehari-Takagi interpolation problems*, Integral Equations and Operator Theory, to appear.

[92] I. Gohberg and A. Semencul, *On the inversion of finite Toeplitz matrices and their continuous analogs*, Mat. Issled., 2 (1972), pp. 201–233.

[93] G. H. Golub, *Numerical methods for solving linear least squares problems*, Numer. Math., 7 (1965), pp. 206–216.

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

[95] 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.

[96] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Englewood Cliffs, NJ, second ed., 1991.

[97] G. Heinig, *Inversion of generalized Cauchy matrices and other classes of structured matrices*, IMA volumes in Mathematics and its Applications, Springer-Verlag, 69 (1994), 95 – 114.

[98] G. Heinig, P. Jankowski, and K. Rost, *Fast inversion algorithms of Toeplitz-plus-Hankel matrices*, Numerische Mathematik, 52 (1988), pp. 665–682.

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

[100] J. W. Helton, *Operator Theory, Analytic Functions, Matrices and Electrical Engineering*, Conference Board of the Mathematical Sciences, American Mathematical Society, Providence, Rhode Island, 1987.

[101] C. Hermite, *Extrait d'une lettre de mr. ch. hermite de paris á mr. borchardt de berlin, sur le nombre de racines d'une Équation algébrique comprises entre des limites données*, Journal für die Reine und Angewandte Mathematik, 52 (1856), pp. 39–51.

[102] A. Houacine, *Regularized fast recursive least squares algorithms for finite memory filtering*,

IEEE Transactions on Signal Processing, 40 (1992), pp. 758–769.

[103] S. F. Hsieh, K. J. R. Liu, and K. Yao, *A unified square-root-free approach for QRD-based recursive least-squares estimation*, IEEE Transactions on Signal Processing, 41 (1993), pp. 1405–1409.

[104] T. Huckle, *Iterative methods for Toeplitz-like methods*, Technical Report SCCM 94-5, Stanford University (1994).

[105] I. S. Iohvidov, *Hankel and Toeplitz Matrices and Forms*, Birkhäuser, Boston, MA, 1982.

[106] T. Kailath, *Some Chandrasekhar-type algorithms for quadratic regulator problems*, in Proc. Conference on Decision and Control, New Orleans, December 1972, pp. 219–223.

[107] ———, *Some new algorithms for recursive estimation in constant linear systems*, IEEE Transactions on Information Theory, 19 (1973), pp. 750–760.

[108] ———, *Linear Systems*, Prentice Hall, Englewood Cliffs, NJ, 1980.

[109] ———, *Signal processing applications of some moment problems*, in Moments in Mathematics, H. Landau, ed., American Mathematical Society, Providence, vol. 37, 1987, pp. 71–109.

[110] ———, *From Kalman filtering to innovations, martingales, scattering and other nice things*, in Mathematical System Theory: The Influence of R.E. Kalman, A. C. Antoulas, ed., Springer-Verlag, Berlin, Providence, 1991, ch. 2, pp. 55–88.

[111] ———, *Remarks on the origin of the displacement-rank concept*, Applied Mathematics and Computation, 45 (1991), pp. 193–206.

[112] T. Kailath, A. Bruckstein, and D. Morgan, *Fast matrix factorizations via discrete transmission lines*, Linear Algebra and Its Applications, 75 (1986), pp. 1–25.

[113] 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.

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

[115] ———, *Displacement ranks of matrices and linear equations*, Journal of Mathematical Analysis and Applications, 68 (1979), pp. 395–407.

[116] T. Kailath and L. Ljung, *A scattering theory framework for fast least-squares algorithms*, in Multivariate Analysis – IV, North Holland, 1977, pp. 387–406.

[117] T. Kailath, L. Ljung, and M. Morf, *A new approach to the determination of Fredholm resolvents of nondisplacement kernels*, in Topics in Functional Analysis, I. Gohberg and M. Kac, eds., Academic, 1978, pp. 169–184.

[118] T. Kailath, M. Morf, and G. S. Sidhu, *Some new algorithms for recursive estimation in constant linear discrete-time systems*, in Proc. Seventh Princeton Conf. Inform. Sci. Systems, Princeton, NJ, 1973, pp. 344–352.

[119] T. Kailath and V. Olshevsky, *Displacement structure approach to Chebyshev-Vandermonde and related matrices*, Journal Integral Equations and Operator Theory, 1995.

[120] T. Kailath and A. H. Sayed, *Fast algorithms for generalized displacement structures*, in Proc. International Symposium on the Mathematical Theory of Networks and Systems, H. Kimura and S. Kodama, eds., Kobe, Japan, June 1991, Mita Press, pp. 27–32.

[121] T. Kailath, A. Vieira, and M. Morf, *Inverses of Toeplitz operators, innovations, and orthogonal polynomials*, SIAM Review, 20 (1978), pp. 106–119.

[122] T. Kailath, A. C. Vieira, and M. Morf, *Orthogonal transformation (square-root) implementations of the generalized Chandrasekhar and generalized Levinson algorithms*, in Lecture Notes in Control and Information Sciences, A. Bensoussan and J. Lions, eds., Springer-Verlag, NY, 1978, pp. 81–91. vol. 32.

[123] R. E. Kalman, *A new approach to linear filtering and prediction problems*, Trans. ASME J. Basic Eng., 82 (1960), pp. 34–45.

[124] H. Kimura, *Conjugation, interpolation and model matching in H^∞*, Int. J. Control, 49 (1989), pp. 269–307.

[125] I. V. Kovalishna and V. P. Potapov, *Seven Papers Translated from the Russian*, vol. 138 of 2, Amer. Math. Soc. Transl., Providence, RI, 1988.

[126] M. G. Krein and A. A. Nudelman, *The Markov Problem of Moments and Extremal Problems*, Moscow, 1973. (Russian).

[127] H. J. Landau, *Classical background of the moment problem*, in Moments in Mathematics, H. J. Landau, ed., Amer. Math. Soc., Providence, 1987, pp. 1–15.

[128] ———, *Maximum entropy and the moment problem*, Bull. Amer. Math. Soc., 16 (1987), pp. 47–77.

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

[130] H. LEV-ARI, *Nonstationary Lattice-Filter Modeling*, PhD thesis, Stanford University, Stanford, CA, December 1983.

[131] ———, *Extension problems under the displacement structure regime*, in Signal Processing, part I: Signal Processing Theory, L. Auslander, T. Kailath, and S. Mitter, eds., Springer–Verlag, NY, 1990, pp. 211–227.

[132] H. LEV-ARI, Y. BISTRITZ, AND T. KAILATH, *Generalized Bezoutians and families of efficient zero-location procedures*, IEEE Transactions on Circuits and Systems, 38 (1991), pp. 170–185.

[133] H. LEV-ARI AND T. KAILATH, *Lattice filter parametrization and modeling of nonstationary processes*, IEEE Transactions on Information Theory, 30 (1984), pp. 2–16.

[134] ———, *Triangular factorization of structured Hermitian matrices*, Operator Theory: Advances and Applications, 18 (1986), pp. 301–324.

[135] ———, *State-space approach to factorization of lossless transfer functions and structured matrices*, Linear Algebra and Its Applications, 162–164 (1992), pp. 273–295.

[136] H. LEV-ARI, T. KAILATH, AND J. CIOFFI, *Least squares adaptive lattice and transversal filters: A unified geometrical theory*, IEEE Transactions on Information Theory, IT–30 (1984), pp. 222–236.

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

[138] D. J. N. LIMEBEER AND B. D. O. ANDERSON, *An interpolation theory approach to H^∞-controller degree bounds*, Linear Algebra and Its Applications, 98 (1988), pp. 347–386.

[139] M. S. LIVSIC AND A. A. YANTSEVICH, *Operator Colligations in Hilbert Spaces*, V. H. Winston, 1979.

[140] L. LJUNG, T. KAILATH, AND B. FRIEDLANDER, *Scattering theory and linear least squares estimation, Part I: Continuous-time problems*, Proc. IEEE, 64 (1976), pp. 131–138.

[141] J. MAKHOUL, *Linear prediction: A tutorial review*, Proceedings IEEE, 63 (1975), pp. 561–580.

[142] J. D. MARKEL AND A. H. GRAY, *Linear Prediction of Speech*, Communications and Cybernetics, Springer-Verlag, Berlin, 1976.

[143] G. A. MERCHANT AND T. W. PARKS, *Efficient solution of a Toeplitz-plus-Hankel coefficient system of equations*, IEEE Transactions on Acoustics, Speech and Signal Processing, 30 (1982), pp. 40–44.

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

[145] ———, *Doubling algorithms for Toeplitz and related equations*, in Proc. IEEE Inter. Conf. on Acoustics, Speech and Signal Processing, Denver, 1980, pp. 954–959.

[146] M. MORF AND T. KAILATH, *Square root algorithms for least squares estimation*, IEEE Transactions on Automatic Control, 20 (1975), pp. 487–497.

[147] M. MORF, G. S. SIDHU, AND T. KAILATH, *Some new algorithms for recursive estimation in constant, linear, discrete-time systems*, IEEE Transactions on Automatic Control, 19 (1974), pp. 315–323.

[148] J. G. NAGY AND R. J. PLEMMONS, *Some fast Toeplitz least squares algorithms*, in Proc. SPIE Conference: Advanced Signal Processing Algorithms, Architectures, and Implementations II, 1991, pp. 35–46. vol. 1566.

[149] A. A. NUDELMAN, *On a new problem of moment type*, Soviet Math. Doklady, 18 (1977), pp. 507–510.

[150] ———, *On a generalization of classical interpolation problems*, Dokl. Akad. Nauk SSR, 256 (1981), pp. 790–793. Soviet Math. Dokl., vol. 23, pp. 125–128, 1981.

[151] R. ONN, A. O. STEINHARDT, AND A. W. BOJANCZYK, *The Hyperbolic singular value decomposition and applications*, IEEE Transactions on Signal Processing, 39 (1991), pp. 1575–1588.

[152] D. PAL, *Fast Algorithms for Structured Matrices with Arbitrary Rank Profile*, PhD thesis, Stanford University, Stanford, CA, May 1990.

[153] 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.

[154] ———, *Displacement structure approach to singular root distribution problems: The unit circle*, IEEE Transactions on Automatic Control, 39 (1994), pp. 238–245.

[155] ———, *Fast triangular factorization and inversion of Hankel and related matrices with arbitrary rank profile*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 451–478.

[156] V. PAN, *On computations with dense structured matrices*, Math. of Computation, 55 (1990), pp. 179–190.

[157] ———, *Parametrization of Newton's iteration for computations with structured matrices*, Computers and Math. with Applications, 24 (1992), pp. 61–75.

[158] ———, *Concurrent iterative algorithm for Toeplitz-like linear systems*, IEEE Trans. Parallel and Distributed Systems, 24 (1993), pp. 592–600.
[159] Y. Potapov, *The multiplicative structure of J-contractive matrix functions*, Amer. Math. Soc. Transl., 15 (1960), pp. 131–244.
[160] J. G. Proakis, C. M. Rader, F. Ling, and C. L. Nikias, *Advanced Digital Signal Processing*, Macmillan Publishing Co., New York, NY, 1992.
[161] C. M. Rader and A. O. Steinhardt, *Hyperbolic Householder transformations*, IEEE Transactions on Acoustics, Speech and Signal Processing, 34 (1986), pp. 1589–1602.
[162] S. K. Rao and T. Kailath, *Orthogonal digital filters for VLSI implementation*, IEEE Transactions on Circuits and Systems, 31 (1984), pp. 933–945.
[163] ———, *VLSI arrays for digital signal processing: A model-identification approach to digital filter realizations*, IEEE Transactions on Circuits and Systems, 32 (1985), pp. 1105–1117.
[164] R. M. Redheffer, *On the relation of transmission-line theory to scattering and transfer*, J. Math. Phys., 41 (1962), pp. 1–41.
[165] L. Reichel and G. Opfer, *Chebyshev-Vandermonde systems*, Math. of Computation, 57 (1991), pp. 703–721.
[166] J. Rissanen, *Algorithms for triangular decomposition of block Hankel and Toeplitz matrices with application to factoring positive matrix polynomials*, Mathematics of Computation, 27 (1973), pp. 147–154.
[167] E. A. Robinson and S. Treitel, *Geophysical Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
[168] ———, *Maximum entropy and the relationship of the partial autocorrelation to the reflection coefficients of a layered system*, IEEE Trans. Acoust. Speech and Signal Processing, 28 (1980), p. 22.
[169] M. Rosenblum and J. Rovnyak, *Hardy Classes and Operator Theory*, Oxford Univ. Press, 1985.
[170] A. H. Sayed and T. Kailath, *Structured matrices and fast RLS adaptive filtering*, in 2nd IFAC Workshop on Algorithms and Architectures for Real-Time Control, P. J. Fleming and W. H. Kwon, eds., Seoul, Korea, September 1992, Pergamon Press, pp. 211–216.
[171] A. H. Sayed, *Displacement Structure in Signal Processing and Mathematics*, PhD thesis, Stanford University, Stanford, CA, August 1992.
[172] ———, *Time-variant structured matrices: An application to instrumental variable methods*, Proc. SPIE Conference on Advanced Signal Processing, 2296 (1994), pp. 516–527.
[173] A. H. Sayed, T. Constantinescu, and T. Kailath, *Square-root algorithms for structured matrices, interpolation, and completion problems*, IMA volumes in Mathematics and Its Applications, Springer-Verlag, 69 (1994).
[174] ———, *Time-variant displacement structure and interpolation problems*, IEEE Transactions on Automatic Control, 39 (1994), pp. 960–976.
[175] A. H. Sayed and T. Kailath, *Fast algorithms for generalized displacement structures and lossless systems*, Linear Algebra and Its Applications, to appear.
[176] ———, *Recursive solutions to rational interpolation problems*, in Proc. IEEE International Symposium on Circuits and Systems, San Diego, CA, May 1992, pp. 2376–2379.
[177] ———, *A state-space approach to adaptive filtering*, in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Minneapolis, MN, April 1993, pp. 559–562.
[178] ———, *Extended Chandrasekhar recursions*, IEEE Transactions on Automatic Control, 39 (1994), pp. 619–623.
[179] ———, *A state-space approach to adaptive RLS filtering*, IEEE Signal Processing Magazine, 11 (1994), pp. 18–60.
[180] ———, *A look-ahead block Schur algorithm for Toeplitz-like matrices*, SIAM J. Matrix Anal. Appl., (1995), to appear.
[181] A. H. Sayed, T. Kailath, and H. Lev-Ari, *The Chandrasekhar recursions from the schur algorithm*, IEEE Transactions on Automatic Control, 39 (1994), pp. 2265–2269.
[182] A. H. Sayed, T. Kailath, H. Lev-Ari, and T. Constantinescu, *Recursive solutions of rational interpolation problems via fast matrix factorization*, Integral Equations and Operator Theory, 20 (1994), pp. 84–118.
[183] A. H. Sayed, H. Lev-Ari, and T. Kailath, *Fast triangular factorization of the sum of quasi-Toeplitz and quasi-Hankel matrices*, Linear Algebra and Its Applications, 191 (1993), pp. 77–106.
[184] ———, *Time-variant displacement structure and triangular arrays*, IEEE Transactions on Signal Processing, 42 (1994), pp. 1052–1062.
[185] 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).
[186] G. S. Sidhu, *A Shift Invariance Approach to Fast Estimation Algorithms*, PhD thesis, Stanford University, Stanford, CA, 1975.
[187] D. T. M. Slock, *Fast Algorithms for Fixed-Order Recursive Least-Squares Parameter Estimation*, PhD thesis, Stanford University, Stanford, CA, 1989.
[188] D. T. M. Slock and K. Maouche, *The fast subsampled updating recursive least-squares (FSURLS) algorithm for adaptive filtering based on displacement structure and fft*, Signal Processing, 40 (1994).
[189] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, NY, 1973.
[190] G. Strang, *A proposal for Toeplitz matrix calculations*, Stud. Appl. Math., 74 (1986), pp. 171–176.
[191] D. R. Sweet, *Fast Toeplitz orthogonalization*, Numerische Mathematik, 43 (1984), pp. 1–21.
[192] G. Szegő, *Orthogonal Polynomials*, vol. 23, American Mathematical Society Colloquium Publishing, Providence, RI, 1939.
[193] W. Trench, *An algorithm for the inversion of finite Toeplitz matrices*, J. Soc. Indust. Appl. Math., 12 (1964), pp. 515–522.
[194] Z. Vavrin, *A unified approach to Loewner and Hankel matrices*, Linear Algebra and Its Applications, 143 (1991), pp. 171–222.
[195] G. Verghese, B. Friedlander, and T. Kailath, *Scattering theory and linear least-squares estimation, Part III: The estimates*, IEEE Transactions on Automatic Control, 25 (1980), pp. 779–794.
[196] R. A. Wiggins and E. A. Robinson, *Recursive solution to the multichannel filtering problem*, J. Geophys. Res., 70 (1965), pp. 1885–1891.
[197] A. E. Yagle, *New analogs of split algorithms for arbitrary Toeplitz-plus-Hankel matrices*, IEEE Transactions on Signal Processing, 39 (1991), pp. 2457–2463.
[198] C. J. Zarowski, *A Schur algorithm and linearly connected processor array for Toeplitz-plus-Hankel matrices*, IEEE Transactions on Signal Processing, 40 (1992), pp. 2065–2078.