tangent at the point can be studied by a transformation of variable $z = 1/\zeta$.

V. Conclusions

The convexity of generalized frequency response curve and inner frequency response set of $\mathcal{D}$-stable polynomial has been proved through a curvature based method. It is felt that this approach may be more easily generalized to investigate the rational function case.

Acknowledgment

The author would like to thank the reviewer for valuable suggestions for improving the quality of the note and pointing out some possible technical pitfalls.

References

[1] J. C. Hamann and B.R. Barmish, "Convexity of frequency response arcs associated with a stable polynomial," *IEEE Trans. Automat. Contr.*, vol. 38, no. 6, pp. 904–915, 1993.
[2] H. W. Guggenheimer, *Differential Geometry*. New York: Dover, 1977.
[3] C.-C. Hsiung, *A First Course in Differential Geometry*. New York: Wiley, 1981.
[4] S. S. Chern, Ed., Studies in Global Geometry and Analysis. Englewood Cliffs, New Jersey: Prentice-Hall, 1967.
[5] M. Spivak, *A Comprehensive Introduction to Differential Geometry*. Brendeis University: Spivak, 1970.

Generalized Chandrasekhar Recursions from the Generalized Schur Algorithm

Ali H. Sayed, Thomas Kailath, and Hanoch Lev-Ari

***Abstract*—We present a new approach to the Chandrasekhar recursions and some generalizations thereof. The derivation uses the generalized Schur recursions, which are $O(N^2)$ recursions for the triangular factorization of $N \times N$ matrices having a certain Toeplitz-like displacement structure. It is shown 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, which are $O(Nn^2)$ recursions for estimating the n-dimensional state of a time-invariant (or constant-parameter) system from N measured outputs. It is further noted that the generalized Schur algorithm factors more general structured matrices, and this fact is readily used to extend the Chandrasekhar recursions to a class of time-variant state-space models, special cases of which often arise in adaptive filtering.**

Manuscript received March 2, 1992; revised February, 1993 and December, 1993. This work was supported in part by the Air Force Office of Scientific Research, Air Force Systems Command under Contract AFOSR91-0060 and by a fellowship (for A. Sayed) from Fundação de Amparo à Pesquisa do Estado de São Paulo and by Escola Politécnica da Universidade de São Paulo, Brazil.

A. H. Sayed is with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106 USA.

T. Kailath is with the Information Systems Laboratory, Stanford University, Stanford CA 94305 USA.

H. Lev-Ari is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115 USA.

IEEE Log Number 9403962.

I. Introduction

Certain fast algorithms for linear least-squares estimation in constant-parameter discrete-time systems were first presented nearly two decades ago for both stationary and nonstationary processes [1]–[4]. Unlike the well-known discrete-Riccati-equation-based Kalman filtering algorithm [5], the new algorithms are the discrete time counterparts of certain so-called generalized Chandrasekhar recursions [6], which extended certain equations introduced by Chandrasekhar to solve finite-time Wiener-Hopf integral equations [7]. Soon after their introduction, the new algorithms were recognized to be closely related to the well-known Levinson algorithm [8] 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 [9] devised to account for the fact that the appropriate coefficient matrix for constant-parameter state-space systems is not Toeplitz but is close-to-Toeplitz in a certain sense [10]. It was shown in [9] that the additional structure provided by the assumption of a constant state-space model allowed the generalized Levinson algorithm to be simplified to the Chandrasekhar recursions of the algorithm in [2]; for the stationary case, a similar connection to the usual Levinson algorithm for Toeplitz matrices was observed by Lindquist [3].

This provided a nice hierarchy: briefly, solving a general set of $N \times N$ linear equations needs $O(N^3)$ elementary computations. The Toeplitz and close-to-Toeplitz assumption allows this to be reduced to $O(N^2)$ by using the generalized Levinson algorithms, while further assuming an n-dimensional underlying constant-parameter (or time-invariant) state-space model allowed, via the Chandrasekhar recursions, a reduction to $O(Nn^2)$ computations. The Riccati equation handles time-variant state-space systems, but takes $O(Nn^3)$ elementary computations whether or not the system is time variant. While these connections were satisfying, it was already clear at that time that the Levinson-type algorithms were not the most natural progenitor of the discrete Chandrasekhar equations. The reason is that the Levinson algorithms are essentially fast algorithms for obtaining the triangular factors of the inverse of Toeplitz (and close-to-Toeplitz) matrices, while the Chandrasekhar recursions essentially give the factors of the Toeplitz (and close-to-Toeplitz) matrices themselves. This distinction is somewhat obscured in the Toeplitz case, because the fast algorithms for the two matrix problems are very closely related and can be easily derived from each other. This fact is true even for certain (so-called admissible [11]) families of close-to-Toeplitz matrices, but the relationship breaks down in general. It was only in the late 1970s that it was first realized [12] that the appropriate fast algorithm for direct factorization was the so-called Schur algorithm, going back to a paper of Schur in 1917 [13], rather than the Levinson algorithm first presented in 1947 [8].

With these insights, it was natural to examine the relationship of the Chandrasekhar recursions and the (generalized) Schur algorithm. This was done by one of the authors in 1982 [14] by using the generating function language approach developed in [15]. More recently, Georgiou *et al.* [16] also wrote about the relation between the Schur algorithm, the Chandrasekhar recursions, and matrix spectral factorization problems. The results and techniques in [16], however, apply only to Toeplitz matrices and stationary stochastic processes that arise from stable constant-parameter state-space models with certain special initial conditions; these methods do not permit extension to any other case, e.g., unstable systems or arbitrary initial conditions.

In this paper, we present a new approach that is valid for all constant-parameter models. The derivation also includes some new Chandrasekhar recursions [18], [19] that have been obtained for a special class of time-variant models that exhibit a certain structure in their time variation.

II. Array Form of the Schur Algorithm

We start by reviewing some results from a famous paper of Schur [13] on the fast triangular factorization of (covariance) matrices with a particular structure [see (1)]. Among several important results in [13] is the following: given an $N \times N$ positive-definite Hermitian matrix R of the special form

$$R = \mathbf{L}(\mathbf{u}_0)\mathbf{L}^*(\mathbf{u}_0) - \mathbf{L}(\mathbf{v}_0)\mathbf{L}^*(\mathbf{v}_0) \tag{1}$$

where the symbol * stands for Hermitian conjugation (complex conjugation for scalars), $\mathbf{u}_0$ and $\mathbf{v}_0$ are two column vectors, say

$$\mathbf{u}_0^{\mathrm{T}} = [u_{00} \quad u_{10} \quad \cdots \quad u_{N-1,0}],$$

$$\mathbf{v}_0^{\mathrm{T}} = [v_{00} \quad v_{10} \quad \cdots \quad v_{N-1,0}]$$

and $\mathbf{L}(\mathbf{a})$ denotes a lower triangular Toeplitz matrix with first column equal to $\mathbf{a}$, there is an efficient recursive algorithm for finding the Cholesky factorization of R, viz., $R = \overline{L}\overline{L}^*$, where $\overline{L}$ is lower triangular. The computational complexity of the algorithm is $O(N^2)$ elementary operations (additions and multiplications), and it can be derived in a variety of ways (see, e.g., [11], [13], [15], [19], [20]). The recursive procedure has a simple array form: start with $G_0 = G = [\mathbf{u}_0 \mathbf{v}_0]$ and choose a $(1 \oplus -1)$-unitary (i.e., hyperbolic) rotation Θ_0 that rotates the top row of G_0 to proper form, viz., $[u_{00} v_{00}]\ \Theta_0 = [\overline{u}_{00} 0]$, where $|\overline{u}_{00}|^2 = |u_{00}|^2 - |v_{00}|^2$. An expression for Θ_0 can be given in terms of the so-called reflection coefficient γ_0

$$\Theta_0 = \frac{1}{\sqrt{1-|\gamma_0|^2}} \begin{bmatrix} 1 & -\gamma_0 \\ -\gamma_0^* & 1 \end{bmatrix}, \qquad \gamma_0 = \frac{v_{00}}{u_{00}}.$$

It follows from the positive-definiteness of R that its $(0, 0)$ entry is positive and hence, $|u_{00}|^2 - |v_{00}|^2 > 0$, or equivalently, $|\gamma_0| < 1$. Multiplying G_0 by Θ_0 leads to a postarray $\overline{G}_0$ of the form

$$G_0\boldsymbol{\Theta}_0 = \begin{bmatrix} u_{00} & v_{00} \\ u_{10} & v_{10} \\ u_{20} & v_{20} \\ \vdots & \vdots \end{bmatrix} \qquad \boldsymbol{\Theta}_0 = \begin{bmatrix} \overline{u}_{00} & 0 \\ \overline{u}_{10} & \overline{v}_{10} \\ \overline{u}_{20} & \overline{v}_{20} \\ \vdots & \vdots \end{bmatrix} = \overline{G}_0.$$

We now proceed to shift down the first column of $\overline{G}_0$ by one element leading to G_1

$$\overline{G}_0 = \begin{bmatrix} \overline{u}_{00} & 0 \\ \overline{u}_{10} & \overline{v}_{10} \\ \overline{u}_{20} & \overline{v}_{20} \\ \vdots & \vdots \end{bmatrix} \xrightarrow{\text{shift}} \begin{bmatrix} 0 & 0 \\ u_{11} & v_{11} \\ u_{21} & v_{21} \\ \vdots & \vdots \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ G_1 \end{bmatrix}.$$

The recursive procedure now continues as follows: compute γ_1, multiply the prearray G_1 by Θ_1 to reduce its top row to proper form, shift down the first column of the postarray $\overline{G}_1$, and so on. Schematically, we have the following picture

$$G_i = \begin{bmatrix} x & x \\ x & x \\ x & x \\ \vdots & \vdots \end{bmatrix} \xrightarrow{\Theta_i(\gamma_i)} \begin{bmatrix} x & 0 \\ x & x \\ x & x \\ \vdots & \vdots \end{bmatrix} \xrightarrow{\text{shift}} \begin{bmatrix} 0 & 0 \\ x & x \\ x & x \\ \vdots & \vdots \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ G_{i+1} \end{bmatrix}. \tag{2}$$

We can interpret (2) in words: 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}.

A. Quasi-Toeplitz Matrices

It is straightforward to verify that a matrix R has the special form (1) if, and only if, it satisfies the following so-called displacement equation [10]

$$R - ZRZ^* = GJG^*, \qquad G = [\mathbf{u}_0 \quad \mathbf{v}_0], \qquad J = (1 \oplus -1). \tag{3}$$

We say that G is a generator of R and that $(1, -1)$ is its displacement inertia. Moreover, R is termed a quasi-Toeplitz matrix, since all such matrices can be shown to be congruent to a Toeplitz matrix [15]. We now go back to the array form (2) of the Schur algorithm and verify its relevance to the Cholesky factorization of quasi-Toeplitz matrices $R \equiv [r_{mj}]_{m,j=0}^{N-1}$ as in (3). Let R_1 denote the Schur complement of r_{00} in R. The claim is that R_1 is also quasi-Toeplitz with a generator equal to the matrix G_1 obtained after the first step of (2), viz., $R_1 = ZR_1Z^* = G_1JG_1^*$. To check this, we compute $R_1 - ZR_1Z^*$ and show that it can indeed be factored as $G_1JG_1^*$. So let l_0 and g_0 denote the first column of R and the first row of G respectively. It follows from (3) and from the definition of Θ_0 that $l_0 = G\Theta_0[\overline{u}_{00}0]^*$ and $d_0 = |\overline{u}_{00}|^2$. But R_1 is defined by

$$R - l_0 d_0^{-1} l_0^* = \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & R_1 \end{bmatrix} \equiv \tilde{R}_1.$$

It is then straightforward to verify that $R_1 - ZR_1Z^* = G_1JG_1^*$, which shows that G_1 is indeed a generator matrix of $R_1 = [r_{mj}^{(1)}]_{m,j=0}^{N-2}$. This process can now be repeated. If we define R_2 to be the Schur complement of $r_{00}^{(1)}$ in R_1 then G_2 is a generator of R_2 and so on. In summary, we have given an alternative verification of the following result established earlier in [11], [15], [20].

Lemma 1 (Schur Complements): The matrices G_i associated with the array form (2) of the Schur algorithm are generator matrices of the successive Schur complements of the Hermitian quasi-Toeplitz matrix R given by (3), viz., $R_i = \mathbf{L}(\mathbf{u}_i)\mathbf{L}^*(\mathbf{u}_i) - \mathbf{L}(\mathbf{v}_i)\mathbf{L}^*(\mathbf{v}_i)$, where R_i denotes the Schur complement with respect to the leading $i \times i$ submatrix of R, and $\{\mathbf{u_i}, \mathbf{v_i}\}$ are the array vectors obtained after i steps of the algorithm. Moreover, the columns of the Cholesky factor $\overline{L}$ (denoted by $\{\overline{l}_i\}$) are the first columns of the successive proper generators $\overline{G}_i$, viz., $\overline{l}_i = G_i\Theta_i[1\,0]^*$. ■

B. Generalized Schur Algorithm

The striking fact is that these results can be nicely extended to matrices with quite general displacement structure (see [19], [26]). A first generalization consists in going beyond quasi-Toeplitz matrices to those with displacement inertia (p, q). For example, a special structure that is relevant to the discussion in later sections is the following

$$\mathcal{R} - \mathcal{Z}_p\mathcal{R}\mathcal{Z}_p^* = \mathcal{G}\mathcal{J}\mathcal{G}^* \tag{4}$$

where $\mathcal{R}$ is a positive-definite matrix with $p \times p$ block entries, $\mathcal{Z}_p$ is the lower triangular shift matrix with ones on the p^{th} subdiagonal and zeros elsewhere, $J = (I_p \oplus -I_q)$ is a signature matrix, and $\mathcal{G}$ is a generator matrix with $p + q$ columns. The Cholesky factorization of such $\mathcal{R}$ can be computed recursively by using the following block array picture (see, e.g., [15], [19]), which is a nice generalization of (2): start with $\mathcal{G}_0 = \mathcal{G}$ and repeat for $i \geq 0$: determine a J-unitary matrix Θ_i that reduces the top p rows of $\mathcal{G}_i$ (denoted by g_i) to the form $g_i\Theta_i = [r_i\mathbf{0}]$, where r_i is a $p \times p$ matrix. That is, a $p \times q$ zero-block is introduced in $g_i\Theta_i$; shift down the first p columns of $\mathcal{G}_i\Theta_i$ by p steps and keep the last q columns unaltered.

III. Exploiting State-Space Structure

We now show that if $\mathcal{R}$ can be obtained as the covariance matrix of the output process of a state-space model, then the array algorithm gets simplified and collapses to the so-called Chandrasekhar recursions in state-space estimation theory. This is due to the fact that the state-space assumption imposes further structure on the generator matrix itself and hence, allows for more simplifications.

We first give a brief review of the state-space estimation problem (see, e.g., [21], [22]). Consider a $p \times 1$ process $\{\mathbf{y}_i\}$ with an n-dimensional state-space model

$$\mathbf{x}_{i+1} = F_i\mathbf{x}_i + G_i\mathbf{u}_i, \qquad \mathbf{y}_i = H_i\mathbf{x}_i + \mathbf{v}_i, \quad \text{for } i \geq 0 \tag{5}$$

where $\{F_i, G_i, H_i\}$ are known matrices with dimensions $n \times n$, $n \times m$, and $p \times n$ respectively. We assume that $\mathbf{x}_0$, $\mathbf{u}_i$, and $\mathbf{v}_i$ are stochastic variables that satisfy $E\mathbf{x}_0 = \bar{\mathbf{x}}_0$, $E(\mathbf{x}_0 - \bar{\mathbf{x}}_0)(\mathbf{x}_0 - \bar{\mathbf{x}}_0)^* = \Pi_0$, $E\mathbf{u}_i(\mathbf{x}_0 - \bar{\mathbf{x}}_0)^* = E\mathbf{v}_i(\mathbf{x}_0 - \bar{\mathbf{x}}_0)^* = \mathbf{0}$, $E\mathbf{v}_i = E\mathbf{u}_i = \mathbf{0}$, and

$$E\begin{bmatrix}\mathbf{u}_i \\ \mathbf{v}_i\end{bmatrix}[\mathbf{u}_j^* \quad \mathbf{v}_j^*] = \begin{bmatrix}Q_i & C_i \\ C_i^* & R_i\end{bmatrix}\delta_{ij}$$

with R_i positive-definite. The symbol δ_{ij} is the Kronecker delta function equal to unity when $i = j$ and zero elsewhere, and the letter E denotes expected value. Let $\hat{\mathbf{x}}_{i|i-1}$ and $\hat{\mathbf{y}}_{i|i-1}$ denote the linear least-squares estimates of $\mathbf{x}_i$ and $\mathbf{y}_i$ given $\{\mathbf{y}_0, \cdots, \mathbf{y}_{i-1}\}$ respectively. The Kalman filter [5], [21], [22] computes these quantities via the recursions

$$\hat{\mathbf{y}}_{i|i-1} = H_i\hat{\mathbf{x}}_{i|i-1}, \qquad \hat{\mathbf{x}}_{i+1|i} = F_i\hat{\mathbf{x}}_{i|i-1} + K_iR_{\epsilon,i}^{-1}\epsilon_i \tag{6}$$

where $\epsilon_i = \mathbf{y}_i - H_i\hat{\mathbf{x}}_{i|i-1}$, $R_{\epsilon,i} = \text{cov}(\epsilon_i)$, and $K_i = \text{cov}(\mathbf{x}_{i+1}\epsilon_i)$. Kalman showed that K_i and $R_{\epsilon,i}$ can be computed by the expressions

$$K_i = F_iP_{i|i-1}H_i^* + G_iC_i \quad \text{and} \quad R_{\epsilon,i} = H_iP_{i|i-1}H_i^* + R_i \tag{7}$$

where $P_{i|i-1} = E(\mathbf{x}_i - \hat{\mathbf{x}}_{i|i-1})(\mathbf{x}_i - \hat{\mathbf{x}}_{i|i-1})^*$ is the error covariance in the one-step prediction of $\mathbf{x}_i$, and satisfies the Riccati difference recursion: $P_{0|-1} = \Pi_0$

$$P_{i+1|i} = F_iP_{i|i-1}F_i^* - \overline{K}_{p,i}\overline{K}_{p,i}^* + G_iQ_iG_i^*, \quad \overline{K}_{p,i} = K_iR_{\epsilon,i}^{-*/2}. \tag{8}$$

We can check that the number of operations (i.e,. multiplications and additions) needed in going from index i to index $(i+1)$ in the Riccati recursion is $O(n^3)$, and this is true whether or not the state-space model has constant parameters. As mentioned in the introduction, however, one expects a computationally more efficient procedure in the case of time-invariant (also called constant-parameter) systems $\{F, G, H, Q, R, C\}$. Indeed, it was shown [2] that in the constant-parameter case the complexity can be reduced to $O(n^2\alpha)$ per iteration, where the so-called displacement rank α is given by $\alpha = \text{rank}\,(P_{1|0} - P_{0|-1}) = \text{rank}\,(F\Pi_0F^* + GQG^* - \overline{K}_{p,0}\overline{K}_{p,0}^* - \Pi_0)$. This is achieved by using the so-called Chandrasekhar recursions to compute $\{K_i, R_{\epsilon,i}\}$ for use in (6). There are many forms for the Chandrasekhar recursions [1], [2], [23], but we have shall rederive here perhaps the simplest and numerically most-favored (so-called square-root) version [24].

A. Displacement of $\mathcal{R}$ with State-Space Structure

Let $\mathcal{R}$ denote the covariance matrix of the process $\{\mathbf{y}_i\}$ of the state-space model (5)

$$\mathcal{R} = E[(\mathbf{y}_i - \bar{\mathbf{y}}_i)(\mathbf{y}_j - \bar{\mathbf{y}}_j)^*]_{i,j=0}^{\infty} \equiv [\text{cov}\,(\mathbf{y}_i, \mathbf{y}_j)]_{i,j=0}^{\infty}, \qquad \bar{\mathbf{y}}_j = E\mathbf{y}_j.$$

The matrix $\mathcal{R}$ is clearly a Hermitian positive-definite block-matrix with $p \times p$ block-entries. We now show that for time-invariant state-space models, the covariance matrix $\mathcal{R}$ exhibits displacement structure, in the sense that $\mathcal{R} - \mathcal{Z}_p\mathcal{R}\mathcal{Z}_p^*$ has low rank. We then show that the generalized Schur recursion (Section II-B) reduces to the extended Chandrasekhar recursions when the extra structure provided by the assumed state-space model is properly incorporated into the Schur recursion.

Let $\bar{\mathbf{x}}_i = E\mathbf{x}_i$, and define $\Pi_i = E(\mathbf{x}_i - \bar{\mathbf{x}}_i)(\mathbf{x}_i - \bar{\mathbf{x}}_i)^*$ to be the state covariance matrix. It follows from the state equation (5) that Π_i obeys the recursion $\Pi_{i+1} = F_i\Pi_iF_i^* + G_iQG_i^*$. Moreover, the following identities are easy to verify. Let $\Delta = \Pi_1 - \Pi_0$. Then $\Pi_{i+1} - \Pi_i = F^i\,\Delta F^{*i}$, $\text{cov}\,(\mathbf{y}_i, \mathbf{y}_i) - \text{cov}\,(\mathbf{y}_{i-1}, \mathbf{y}_{i-1}) = HF^{(i-1)}\,\Delta F^{*(i-1)}H^*$, and $\text{cov}\,(\mathbf{y}_i, \mathbf{y}_{i+1}) - \text{cov}\,(\mathbf{y}_{i-1}, \mathbf{y}_i) = HF^{(i-1)}\,\Delta F^{*i}H^*$.

From these identities, we can readily conclude that (recall that $K_0 = F\Pi_0H^* + GC$)

$$\nabla\mathcal{R} = \mathcal{R} - \mathcal{Z}_p\mathcal{R}\mathcal{Z}_p^* = \begin{bmatrix} R_{\epsilon,0} & K_0^*H^* & K_0^*F^*H^* & K_0^*F^{*2}H^* & \cdots \\ HK_0 & H\,\Delta H^* & H\,\Delta F^*H^* & H\,\Delta F^{*2}H^* & \\ HFK_0 & HF\,\Delta H^* & HF\,\Delta F^*H^* & HF\,\Delta F^{*2}H^* & \\ HF^2K_0 & HF^2\,\Delta H^* & HF^2\,\Delta F^*H^* & HF^2\,\Delta F^{*2}H^* & \\ \vdots & & & & \ddots \end{bmatrix}.$$

There is clearly a significant redundancy in the elements of $\mathcal{R} - \mathcal{Z}_p\mathcal{R}\mathcal{Z}_p^*$, since the third 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 $\nabla\mathcal{R}$, which is

$$\begin{aligned}\nabla\mathcal{R} - \begin{bmatrix} R_{\epsilon,0} \\ HK_0 \\ HFK_0 \\ \vdots \end{bmatrix} R_{\epsilon,0}^{-1}[R_{\epsilon,0} \quad K_0^*H^* \quad K_0^*F^*H^* \quad \cdots] \\ = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots \\ 0 & H\,\delta P_1\,H^* & H\,\delta P_1\,F^*H^* & H\,\delta P_1F^{*2}H^* & \\ 0 & HF\,\delta P_1\,H^* & HF\,\delta P_1\,F^*H^* & HF\,\delta P_1F^{*2}H^* & \\ 0 & HF^2\,\delta P_1\,H^* & HF^2\,\delta P_1\,F^*H^* & HF^2\,\delta P_1F^{*2}H^* & \\ \vdots & & & & \ddots \end{bmatrix} \\ = \begin{bmatrix} 0 \\ H \\ HF \\ \vdots \end{bmatrix} \delta P_1\,[0 \quad H^* \quad F^*H^* \quad \cdots]\end{aligned}$$

where we used the following relation

$$\begin{aligned}H\,\Delta H^* - H\overline{K}_{p,0}\overline{K}_{p,0}^*H^* &= H(\Pi_1 - \Pi_0 - \overline{K}_{p,0}\overline{K}_{p,0}^*)H^* \\ &= H(P_{1|0} - P_{0|-1})H^*\end{aligned}$$

and defined $\delta P_1 = P_{1|0} - P_{0|-1}$. It now follows easily that the Schur complement of the leading 2×2 block in $\nabla\mathcal{R}$ is zero. In other words, the displacement $\nabla\mathcal{R}$ has block rank 2

$$\nabla\mathcal{R} = \begin{bmatrix} R_{\epsilon,0} \\ HK_0 \\ HFK_0 \\ \vdots \end{bmatrix} R_{\epsilon,0}^{-1} \begin{bmatrix} R_{\epsilon,0} \\ HK_0 \\ HFK_0 \\ \vdots \end{bmatrix}^* + \begin{bmatrix} 0 \\ H \\ HF \\ \vdots \end{bmatrix} \delta P_1 \begin{bmatrix} 0 \\ H \\ HF \\ \vdots \end{bmatrix}^*.$$

To find a generator for $\mathcal{R}$, we factor δP_1 as $\delta P_1 \equiv P_{1|0} - P_{0|-1} = L_0SL_0^*$, where L_0 is $N \times \alpha$ and S is the $\alpha \times \alpha$ signature matrix of $(P_{1|0} - P_{0|-1})$, viz., S is a diagonal matrix with as many ± 1's on the diagonal as δP_1 has positive and negative eigenvalues. Then we

can write $\nabla\mathcal{R} = \mathcal{R} - \mathcal{Z}_p\mathcal{R}\mathcal{Z}_p^* = \mathcal{G}J\mathcal{G}^*$, where

$$J = \begin{bmatrix} I_p & 0 \\ 0 & S \end{bmatrix} \quad \text{and} \quad \mathcal{G} = \begin{bmatrix} R_{\epsilon,0}^{1/2} & 0 \\ H\overline{K}_{p,0} & HL_0 \\ HF\overline{K}_{p,0} & HFL_0 \\ \vdots & \vdots \end{bmatrix}.$$

This establishes the fact that $\mathcal{R}$ is indeed a structured matrix as defined in Section II-B, and hence we can compute its Cholesky factor via the array form of the generalized Schur algorithm. We may note the special caes of stable F (i.e., the eigenvalues of F have less than unit magnitude) and $\Pi_0 = \overline{\Pi}$, the unique solution of $\overline{\Pi} = F\overline{\Pi}F^* + GQG^*$, which can be seen to yield a block Toeplitz covariance matrix $\mathcal{R}$, studied in [16] by using a different argument, which does not extend to nonstationary processes (non-Toeplitz covariance matrices) as considered here.

B. Generalized Chandrasekhar from Generalized Schur

Notice, however, 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 $\mathcal{R}$. We now verify that because of this additional structure in the generator matrix, the generalized Schur algorithm collapses to the Chandrasekhar recursions.

The first step in the Schur algorithm involves multiplying by Θ_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

$$\mathcal{G}_1 = \begin{bmatrix} R_{\epsilon,0}^{1/2} & HL_0 \\ H\overline{K}_{p,0} & HFL_0 \\ HF\overline{K}_{p,0} & HF^2L_0 \\ \vdots & \vdots \end{bmatrix}.$$

Let Θ_1 be a J-unitary matrix such that $[R_{\epsilon,0}^{1/2} HL_0]\Theta_1 = [X\,0]$. Applying Θ_1 to the first two (block) rows of $\mathcal{G}_1$ (denoted by A) we obtain a (block-triangular) postarray of the form

$$A\boldsymbol{\Theta}_1 = \begin{bmatrix} R_{\epsilon,0}^{1/2} & HL_0 \\ H\overline{K}_{p,0} & HFL_0 \end{bmatrix}\boldsymbol{\Theta}_1 = \begin{bmatrix} X & 0 \\ Y & Z \end{bmatrix}$$

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 $AJA^* = A\Theta_1 J\Theta_1^* A^*$ leading to $XX^* = R_{\epsilon,0} + HL_0SL_0^*H^* = R_{\epsilon,1}$. So we can choose $X = R_{\epsilon,1}^{1/2}$. Moreover, $YX^* = K_0 + FL_0SL_0^*H^* = K_1$ and hence, we can identify $Y = K_1R_{\epsilon,1}^{-1/2} = \overline{K}_{p,1}$. Finally, $YY^* + ZSZ^* = K_0R_{\epsilon,0}^{-1}K_0^* + FL_0SL_0^*F^* = P_{2|1} - P_{1|0} \equiv L_1SL_1^*$, which shows that we can identify Z as L_1. We thus conclude that

$$\begin{bmatrix} R_{\epsilon,0}^{1/2} & HL_0 \\ \overline{K}_{p,0} & FL_0 \end{bmatrix}\boldsymbol{\Theta}_1 = \begin{bmatrix} R_{\epsilon,1}^{1/2} & 0 \\ \overline{K}_{p,1} & L_1 \end{bmatrix}.$$

Therefore, $\mathcal{G}_1\boldsymbol{\Theta}_1$ is equal to (we now invoke the special structure of the rows of $\mathcal{G}_1$)

$$\mathcal{G}_1\boldsymbol{\Theta}_1 = \begin{bmatrix} R_{\epsilon,1}^{1/2} & 0 \\ H\overline{K}_{p,1} & HL_1 \\ HF\overline{K}_{p,1} & HFL_1 \\ \vdots & \vdots \end{bmatrix}.$$

Next we shift down the first p columns to get

$$\mathcal{G}_2 = \begin{bmatrix} R_{\epsilon,1}^{1/2} & HL_1 \\ H\overline{K}_{p,1} & HFL_1 \\ HF\overline{K}_{p,1} & HF^2L_1 \\ \vdots & \vdots \end{bmatrix}$$

choose a J-unitary matrix Θ_2, shift down, from Θ_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 square-root Chandrasekhar recursions

$$\begin{bmatrix} R_{\epsilon,i}^{1/2} & HL_i \\ \overline{K}_{p,i} & FL_i \end{bmatrix}\boldsymbol{\Theta}_{i+1} = \begin{bmatrix} R_{\epsilon,i+1}^{1/2} & 0 \\ \overline{K}_{p,i+1} & L_{i+1} \end{bmatrix} \tag{9}$$

where $\boldsymbol{\Theta}_{i+1}$ is any $J = (I \oplus S)$-unitary matrix that introduces the block zero entry on the right-hand side, and $P_{i+2|i+1} - P_{i+1|i} = L_{i+1}SL_{i+1}^*$.

IV. Structured Time-Variant Models

The computational advantage of the Chandrasedkhar recursions (9) stems from the fact that it propagates a low rank factor L_i instead of $P_{i+1|i}$, where $P_{i+1|i} - P_{i|i-1} = L_iS_iL_i^*$. A direct generalization would be to consider differences of the form $P_{i+1|i} - \boldsymbol{\Psi}_iP_{i|i-1}\boldsymbol{\Psi}_i^*$, where $\boldsymbol{\Psi}_i$ are convenient time-variant matrices that also result in a low rank difference, say of rank α. That is, $P_{i+1|i} - \boldsymbol{\Psi}_iP_{i|i-1}\boldsymbol{\Psi}_i^* \equiv L_iS_iL_i^*$, 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 again the state-space model given by (5), and we shall say that it is a structured time-variant model if there exist $n \times n$ matrices $\boldsymbol{\Psi}_i$ such that F_i, G_i, and H_i vary according to the rules

$$H_i = H_{i+1}\boldsymbol{\Psi}_i, \qquad F_{i+1}\boldsymbol{\Psi}_i = \boldsymbol{\Psi}_{i+1}F_i, \qquad G_{i+1} = \boldsymbol{\Psi}_{i+1}G_i. \tag{10}$$

It is clear that constant-parameter systems satisfy (10) with $\boldsymbol{\Psi}_i = I$. Other special cases of (10) often arise in the recursive least-squares problem [17]–[19]. We shall assume that the covariance matrices R_i, Q_i, and C_i are time-invariant whereas F_i, H_i, G_i vary in time according to (10) (the restrictions on $\{R_i, Q_i, C_i\}$ can be relaxed as discussed in [18], [19]).

The point is that the conditions specified in (10) guarantee that the covariance matrix $\mathcal{R}$ of the output process $\{\mathbf{y}_i\}$ still has a time-invariant displacement structure of the form $\mathcal{R} - \mathcal{Z}_p\mathcal{R}\mathcal{Z}_p^* = \mathcal{G}J\mathcal{G}^*$, and hence its Cholesky factorization can still be carried out via the generalized Schur algorithm. Thus, following the same reasoning as in Section III, we can easily verify that for structured time-variant models as above we get

$$\mathcal{R} - \mathcal{Z}_p\mathcal{R}\mathcal{Z}_p^* = \begin{bmatrix} R_{\epsilon,0}^{1/2} & 0 \\ H_1\overline{K}_{p,0} & H_1L_0 \\ H_2F^{[1]}\overline{K}_{p,0} & H_2F^{[1]}L_0 \\ H_3F^{[2]}\overline{K}_{p,0} & H_3F^{[2]}L_0 \\ \vdots & \vdots \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} R_{\epsilon,0}^{1/2} & 0 \\ H_1\overline{K}_{p,0} & H_1L_0 \\ H_2F^{[1]}\overline{K}_{p,0} & H_2F^{[1]}L_0 \\ H_3F^{[2]}\overline{K}_{p,0} & H_3F^{[2]}L_0 \\ \vdots & \vdots \end{bmatrix}^*$$

where we define $F^{[i]} = F_iF_{i-1}\cdots F_1$, $F^{[0]} = I$, and L_0 and S are defined via the (nonunique) factorization $P_{1|0} - \Psi_0P_{0|-1}\Psi_0^* = L_0SL_0^*$. Applying the generalized Schur algorithm to the above generator we readily verify that it collapses to the following extended Chandrasekhar recursions [18], [19]

$$\begin{bmatrix} R_{\epsilon,i}^{1/2} & H_{i+1}L_i \\ \Psi_{i+1}\overline{K}_{p,i} & F_{i+1}L_i \end{bmatrix}\boldsymbol{\Theta}_{i+1} = \begin{bmatrix} R_{\epsilon,i+1}^{1/2} & 0 \\ \overline{K}_{p,i+1} & L_{i+1} \end{bmatrix}. \tag{11}$$

V. Summary

We reviewed a generalized Schur algorithm for fast Cholesky factorization of matrices with displacement structure. The covariance matrices of the generally nonstationary processes generated by constant-parameter state-space systems have displacement structure. But the state-space assumption induces additional structure, which collapses the Schur recursions to the Chandrasekhar recursions derived in the 1970s for fast recursive linear estimation.

References

[1] 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. Infor. Sci. Syst.*, Princeton, NJ, 1973, pp. 344–352.

[2] M. Morf, G. S. Sidhu, and T. Kailath, "Some new algorithms for recursive estimation in constant, linear, discrete-time systems," *IEEE Trans. Automat. Contr.*, vol. AC-19, no. 4, pp. 315–323, Aug. 1974.

[3] A. Lindquist, "A new algorithm for optimal filtering of discrete-time stationary processes," *SIAM J. Contr.*, vol. 12, pp. 736–746, 1974.

[4] A. Lindquist, "Some reduced-order non-Riccati equations for linear least-squares estimation: the stationary, single output case," *Int. J. Contr.*, vol. 24, pp. 821–842, 1976.

[5] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME J. Basic Eng.*, vol. 82, pp. 34–45, Mar. 1960.

[6] T. Kailath, "Some new algorithms for recursive estimation in constant linear systems," *IEEE Trans. Inform. Theory*, vol. IT-19, no. 6, pp. 750–760, Nov. 1973.

[7] S. Chandrasedkhar, *Radiative Transfer.* New York: Dover, 1950.

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

[9] B. Friedlander, T. Kailath, M. Morf, and L. Ljung, "Extended Levinson and Chandrasekhar equations for general discrete-time linear estimation problems," *IEEE Trans. Automat. Contr.*, vol. AC-23, no. 4, pp. 653–659, Aug. 1978.

[10] T. Kailath, S. Y. Kung, and M. Morf, "Displacement ranks of matrices and linear equations," *J. Math. Anal. Appl.*, vol. 68, pp. 395–407, 1979.

[11] H. Lev-Ari and T. Kailath, "Lattice filter parametrization and modeling of nonstationary processes," *IEEE Trans. Info. Theory*, vol. IT-30, no. 1, pp. 2–16, Jan. 1984.

[12] P. Dewilde, A. Vieira, and T. Kailath, "On a generalized Szegö-Levinson realization algorithm for optimal linear predictions on a network synthesis approach," *IEEE Trans. Circ. Syst.*, vol. CAS-25, pp. 663–675, 1978.

[13] I. Schur, "Über potenzreihen die im Inneren des Einheitskreises beschränkt sind," *Journal für die Reine und Angewandte Mathematik*, vol. 147, pp. 205–232, 1917.

[14] H. Lev-Ari, "The Schur algorithm for processes with a state-space model," unpublished notes, 1982.

[15] H. Lev-Ari and T. Kailath, "Triangular factorization of structured Hermitian matrices," *Operator Theory: Advances and Applications*, I. Gohberg (Ed.). Birkhäuser: Boston, vol. 18, 1986, pp. 301–324.

[16] C. Chang and T. T. Georgiou, "On a Schur-algorithm based approach to spectral factorization: Connection with the Riccati equation," *Linear Algebra and Its Applications*, vol. 171, pp. 233–247, Jul. 1992.

[17] A. H. Sayed and T. Kailath, "A state-space approach to adaptive RLS filtering," *IEEE Signal Processing Magazine*, vol. 11, no. 3, pp. 18–60, Jul. 1994.

[18] ——, "Extended Chandrasekhar recursions," *IEEE Trans. Automat. Contr.*, vol. 39, no. 3, pp. 619–623, Mar. 1994.

[19] A. H. Sayed, Displacement structure in signal processing and mathematics," Ph.D. dissertation, Stanford Univ., Stanford, CA, Aug. 1992.

[20] T. Kailath, A. Bruckstein, and D. Morgan, "Fast matrix factorizations via discrete transmission lines," *Linear Algebra and Its Applications*, pp. 1–25, vol. 75, 1986.

[21] T. Kailath, *Lectures on Wiener and Kalman Filtering*, 2nd ed. New York: Springer-Verlag, 1981.

[22] B. D. O. Anderson and J. B. Moore, *Optimal Filtering.* Englewood Cliffs, NJ: Prentice-Hall, 1979.

[23] M. Morf and T. Kailath, "Square root algorithms for least squares estimation," *IEEE Trans. Automat. Contr.*, vol. AC-20, no. 4, pp. 487–497, Aug. 1975.

[24] T. Kailath, A. C. Vieira, and M. Morf, *Orthogonal Transformation (Square-Root) Implementations of the Generalized Chandrasekhar and Generalized Levinson Algorithms*, (Lecture Notes in Control and Information Science), vol. 32. New York: Springer-Verlag, 1978.

[25] G. Szegö, *Orthogonal Polynomials*, vol. 23. Providence, RI: AMS, 1939.

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

Input–Output Linearization with State Equivalence and Decoupling

In-Joong Ha and Sung-Joon Lee

***Abstract*— In this note, we attempt to characterize the whole class of nonlinear systems that can be linearized to controllable and decouplable linear systems. We present the necessary and sufficient conditions for our problem to be solvable. More importantly, we explicitly characterize the nonlinear system satisfying these conditions by a set of parameters which are invariant under the group action of state feedback and transformation. This set of parameters can be calculated without solving a set of partial differential equations. Using this set of parameters, we can directly determine which of the canonical forms of decouplable and controllable linear systems is feedback equivalent to the nonlinear system. For the design of decoupled system with linear input–output dynamic characteristics, it is more convenient to deal with the canonical form which is the simplest representer of the original system.**

I. Introduction

Suppose that we can transform a nonlinear system of dimension n with the output into a linear system of dimension n with the output via appropriate state feedback and transformation. The resulting closed-loop system then will have the input–output dynamic characteristics of a linear system. Moreover, the internal stability of the closed-loop system can be readily guaranteed. This approach is often called input–output linearization with state equivalence for which the necessary and sufficient conditions have been found by Lee *et al.* [11] for the case of single-input/single-output (SISO) systems and by Cheng *et al.* [1] for the case of multi-input/multi-output (MIMO) systems. The problem we consider in this note is more demanding than the problem of input–output linearization with state equivalence. We require that the MIMO nonlinear system, for which the problem of input–output linearization with state equivalence is solvable, can be decoupled.

There are important reasons why we attempt to solve our problem. First, decoupling is known to be an efficient control method for some MIMO systems. For instance, see [10], [12], and [15]. Second, the necessary and sufficient conditions for our problem to be solvable are

Manuscript received August, 1992; revised March 4, 1993, June 14, 1993, and January 24, 1994.

I.-J. Ha is with the Department of Control and Instrumentation Engineering, Automation and Systems Research Institute, Seoul National University, San 56-1 Shinrim-Dong, Kwanak-Ku, Seoul 151-742, Korea.

S.-J. Lee is with the Department of Electronics Engineering, Seoul National University, San 56-1 Shinrim-Dong, Kwanak-Ku, Seoul 151-742, Korea.

IEEE Log Number 9405635.