Eigenvector Computation for Almost Unitary Hessenberg Matrices and Inversion of Szegő-Vandermonde Matrices via Discrete Transmission Lines *

Vadim Olshevsky
Department of Mathematics and Computer Science
Georgia State University
University Plaza, Atlanta, GA 30303
e-mail: volshevsky@cs.gsu.edu
web: http://www.cs.gsu.edu/˜matvro

Abstract

In this paper we use a *discrete transmission line model* (known to geophysicists as a *layered earth model*) to derive several computationally efficient solutions for the following three problems. **(i)** As is well-known, a Hessenberg matrix capturing recurrence relations for Szegő polynomials differs from unitary only by its last column. Hence, the first problem is how to rapidly evaluate the eigenvectors of this almost unitary Hessenberg matrix. **(ii)** The second problem is to design a fast $O(n^2)$ algorithm for inversion of Szegő-Vandermonde matrices (generalizing the well-known Traub algorithm for inversion of the usual Vandermonde matrices). **(iii)** Finally, the third problem is to extend the well-known Horner rule to evaluate a polynomial represented in the basis of Szegő polynomials. As we shall see, all three problems are closely related, and their solutions can be computed by the same family of fast algorithms.

Although all the results can be derived algebraically, here we reveal a connection to system theory to deduce these algorithms via elementary operations on signal flow graphs for digital filter structures, including the celebrated Markel-Gray filter, widely used in speech processing, and certain other filter structures. This choice not only clarifies the derivation and suggests a variety of possible computational schemes, but it also makes an interesting connection to many other results related to Szegő polynomials which have already been interpreted via signal flow graphs for (generalized) lattice filter structures, including the formulas of the Gohberg-Semencul type for inversion of Toeplitz-like matrices, Schur-type and Levinson-type algorithms, etc. For example, this connection allows us to show that moment matrices corresponding to the Horner-Szegő polynomials, though not Toeplitz, are quasi-Toeplitz, i.e., they have a certain shift-invariance property.

Key words: Horner polynomials, Szegő polynomials, Vandermonde matrices, Szegő-Vandermonde matrices, fast algorithms, signal flow graph, discrete transmission line, lattice filter structure, Markel-Gray filter, companion matrices, confederate matrices, unitary Hessenberg matrices.

AMS subject classification: 65F05, 65L20, 15A09, 15A23

*This work was supported in part by NSF contract. The views and conclusions contained in these documents are those of the author(s) and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the National Science Foundation or the U.S. Government.

1 Introduction

1.1 A physical model

1.1.1 The Szegő polynomials

Sometimes a physical model, even at first glance simple, can be useful for purely mathematical studies. For example, a model of a vibrating string with n discrete masses served as a starting point for many interesting mathematical investigations, see, e.g., [K52] or Appendix in [A65]. Similarly, a simple physical device known to electrical engineers as a *"discrete transmission line"* (i.e., one with the piecewise constant impedance profile), and to geophysicists as a *"layered earth model"*, can be useful to study the algebraic properties of *Szegő polynomials* $\Phi^{\#} = \{\phi_k^{\#}(x)\}$, i.e., polynomials orthonormal with respect to a suitable inner product on the unit circle,

$$< p(x), q(x) >= \frac{1}{2\pi} \int_{-\pi}^{\pi} p(e^{i\theta}) \cdot [q(e^{i\theta})]^* w^2(\theta) d\theta. \tag{1.1}$$

We use the sharp sign $^{\#}$ to follow the usual signal processing designations, where $\{[\phi_k^{\#}(\frac{1}{z^*})]^*\}$ are called *backward predictor polynomials*, see, e.g., [MG76].

$$f(x) = g(x) \tag{1.2}$$

Applying the Gram-Schmidt procedure to the power basis $\{1, x, x^2, \ldots, x^{n-1}\}$ one is able to parameterize the first $(n+1)$ Szegő polynomials $\Phi^{\#} = \{\phi_k^{\#}(x)\}_{k=0}^{n}$ by only $(n+1)$ parameters $\{\mu_0, \rho_1, \ldots, \rho_n\}$ via the well-known two-term recurrence relations [GS58], [G48],

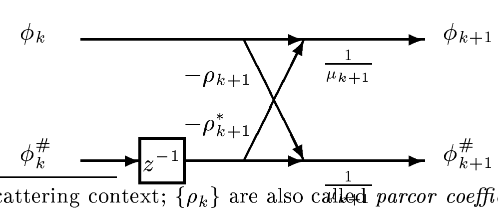
$$\begin{bmatrix} \phi_0(x) \\ \phi_0^{\#}(x) \end{bmatrix} = \frac{1}{\mu_0} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad \begin{bmatrix} \phi_{k+1}(x) \\ \phi_{k+1}^{\#}(x) \end{bmatrix} = \frac{1}{\mu_{k+1}} \begin{bmatrix} 1 & -\rho_{k+1}^* \\ -\rho_{k+1} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & x \end{bmatrix} \begin{bmatrix} \phi_k(x) \\ \phi_k^{\#}(x) \end{bmatrix}. \tag{1.3}$$

The auxiliary polynomials $\{\phi_k(x)\}$ involved in (1.3) obviously have a reversal form: $\phi_k(x) = z^k[\phi^{\#}(\frac{1}{x^*})]^*$. The numbers $\{\rho_0, \rho_1, \ldots, \rho_n\}$ are called the *reflection coefficients*[1], and $\mu_k = \sqrt{1 - |\rho_k|^2}$ are called the *complementary parameters* ($\mu_k := 1$ if $\rho_k = 1$).

1.1.2 Discrete transmission lines

In mathematical and engineering literature one can find several approaches to study problems involving Szegő polynomials. These approaches use different languages, e.g., the interpolation language, array formulations (usually involving structured matrices such as Toeplitz matrices, see, e.g., Sec. 5), reproducing kernel spaces approach, and some others. In this paper we shall take advantage of another method (standard in system theory) associated with *discrete transmission lines*. The point is that the algebraic recurrences in (1.3) can be conveniently represented as a *signal flow graph* in Fig. 1, familiar to an electrical engineer. This representation can be physically *realized* as an electronic device (a transmission line), and such *lattice realizations* (sometimes called *ladder realizations*) have many favorable properties, including inherent stability, low noise accumulation in the state vector loop, possibility to suppress quantization limit cycles under simple arithmetic conventions, etc. They have became increasingly popular in signal modeling, spectrum estimation, speech processing, adaptive filtering, and other applications.

Figure 1. Lattice filter structure realizing (1.3).



[1] This name is used in the inverse scattering context; $\{\rho_k\}$ are also called *parcor coefficients*, or often *Schur parameters*

However, in this paper we do not assume that the reader has any previous knowledge of signal flow diagrams. Thus, the graph in Fig. 1 can be seen as just a convenient graphical representation of algebraic recurrence relations in (1.3), where the delay operation $\boxed{z^{-1}}$ denotes a multiplication by $x = z^{-1}$, scaling is designated by ρ^*_{k+1} or $\frac{1}{\mu_{k+1}}$, and an addition is drawn as . The motivation to use in this paper signal flow diagrams (such as the one in Fig. 1) stems from the fact that this language has already been found to be very useful in *purely algebraic studies* of the properties of various structured matrices, and especially in the design of *fast algorithms.* We refer to [KBM86] (see also [BK86], [BK87a], [BK87b]) for a nice explanation of how the physical properties of a transmission line (i.e., causality, symmetry and energy conservation) can be used to provide a nice interpretation for the classical Schur and Levinson algorithms for factorization of Toeplitz matrices, as well as for the Gohberg-Semencul formula [GS72], [GF74] for inversion of Toeplitz matrices. We also refer to [D82], [LA83], [LAK84] for a discrete-transmission-lines approach[2] to study the more general classes of matrices with Toeplitz-like structure.

1.2 Problem formulation.

In this paper we proceed the work started by our colleagues, and use the above transmission-line-model to provide a surprisingly simple derivation for the solutions of the following three algebraic problems.

1.2.1 Almost unitary Hessenberg eigenvector problem

Let

$$H(x) = b_0 Q_o(x) + b_1 Q_1(x) + \ldots + b_n Q_n(x),$$

where the polynomials $Q = \{Q_k(x)\}$ with $\deg Q_k(x) = k$ satisfy general recurrence relations

$$x \cdot Q_{k-1}(x) = a_{k,k} \cdot Q_k(x) + a_{k-1,k} \cdot Q_{k-1}(x) + ... + a_{0,k} \cdot Q_0(x). \tag{1.4}$$

The Hessenberg matrix

$$C_Q(H) = \begin{bmatrix} a_{01} & a_{02} & \cdots & & \cdots & a_{0,n} - \frac{b_0}{b_n} \\ a_{11} & a_{12} & \cdots & & \cdots & a_{1,n} - \frac{b_1}{b_n} \\ 0 & a_{22} & \cdots & & \cdots & a_{2,n} - \frac{b_2}{b_n} \\ \vdots & \vdots & \ddots & & \ddots & \vdots \\ 0 & \cdots & 0 & a_{n-1,n-1} & & a_{n-1,n} - \frac{b_{n-1}}{b_n} \end{bmatrix} \tag{1.5}$$

capturing the recurrence relations (1.4) has been called the *confederate matrix* in [MB79], where one can find many its useful properties, e.g.,

$$\det(\lambda I - C_Q(H)) = \frac{a_{nn} \cdots a_{00}}{b_n} H(\lambda). \tag{1.6}$$

Of course, for the simplest monomial basis $Q = \{1, x, x^2, \ldots, x^n\}$ the confederate matrix $C_Q(H)$ reduces to the usual companion matrix, so hence (1.6). Further, for the important case of real orthogonal polynomials (satisfying three-term recurrence relations) the matrix $C_Q(H)$ differs from tridiagonal only by its last column, in this special case it is called a *comrade matrix* [B75].

Now, it is well-known and can be easily seen that for the case when $\{Q_k(x)\}$ are Szegő polynomials, the

[2] We listed only a few papers where purely algebraic results were deduced by using the very compact and clarifying arguments based on signal flow diagrams. No possible omissions are intentional; signal flow graphs are widely used to compactly represent and interpret various recursions, see, e.g., [K79], [GL83] among others.

Hessenberg matrix $C_Q(H)$ is constructed from reflection coefficients:

$$C_{\Phi^{\#}}(H) = \begin{bmatrix} -\rho_1\rho_0^* & -\rho_2\mu_1\rho_0^* & -\rho_3\mu_2\mu_1\rho_0^* & \cdots & -\rho_{n-1}\mu_{n-2}\dots\mu_1\rho_0^* & -\rho_n\mu_{n-1}\dots\mu_1\rho_0^* - \frac{b_0}{b_n}\mu_n \\ \mu_1 & -\rho_2\rho_1^* & -\rho_3\mu_2\rho_1^* & \cdots & -\rho_{n-1}\mu_{n-2}\dots\mu_2\rho_1^* & -\rho_n\mu_{n-1}\dots\mu_2\rho_1^* - \frac{b_1}{b_n}\mu_n \\ 0 & \mu_2 & -\rho_3\rho_1^* & \cdots & -\rho_{n-1}\mu_{n-2}\dots\mu_3\rho_2^* & -\rho_n\mu_{n-1}\dots\mu_3\rho_2^* - \frac{b_2}{b_n}\mu_n \\ \vdots & \ddots & \mu_3 & & \vdots & \vdots \\ \vdots & & \ddots & \ddots & -\rho_{n-1}\rho_{n-2}^* & -\rho_n\mu_{n-1}\rho_{n-2}^* - \frac{b_{n-2}}{b_n}\mu_n \\ 0 & \cdots & \cdots & 0 & \mu_{n-1} & -\rho_n\rho_{n-1}^* - \frac{b_{n-1}}{b_n}\mu_n \end{bmatrix} \tag{1.7}$$

and it differs from a unitary matrix only by its last column. Such almost unitary Hessenberg matrices are of interest in several applied areas.

For example, in signal processing literature (1.7) is known under the name the *discrete-time Schwartz form* [M66] because it appears in the context of discrete-time Lyapunov stability test. It has been noted in [M77] (see also [M74]) and elaborated in [ML80], [L80] (and independently in [TKH83]) that such almost unitaryHessenberg matrices describe the state-space structure for the feed-back lattice filters. In [KP83] a recursive nested realization algorithm was specified for the structure in (1.7), see also [K85] for the further extensions. We also refer to a recent monograph [R95] for a nice description of connections of $C_{\Phi^{\#}}(H)$ to the classical Schur and Levinson algorithms, and to lattice filter structures. See also [F96].

In the operator theory literature the structure in (1.7) is associated with the *Naimark dilation*, see, e.g., [C84], Sec. 2.4 in [BC92], and Sec. 6.7 in [FF89].

The computational issues related to such almost unitary Hessenberg matrices were discussed in numerical linear algebra literature, see, e.g., [G82] for a connection to Gaussian quadrature on the unit circle, [G86] for the unitary Hessenberg QR-algorithm (and [G97] for its numerically accurate implementation), [AGR86], [GR87], [AGR91], [BGE91] for direct and inverse unitary eigenvalue problems, [AGR87], [AGR92] for some applications. In particular in [ACR96] one can find an algorithm to compute the eigenvalues of the general almost unitary Hessenberg matrices.

The first problem considered in this paper is how to efficiently compute the eigenvectors of $C_{\Phi^{\#}}(H)$, assuming that its eigenvalues are already known.

1.2.2 Inversion of Szegő-Vandermonde matrices

The second problem addresses here is to efficiently invert the Szegő-Vandermonde matrix,

$$V_{\Phi}^{\#}(x) = \begin{bmatrix} \phi_0^{\#}(x_1) & \phi_1^{\#}(x_1) & \cdots & \phi_{n-1}^{\#}(x_1) \\ \phi_0^{\#}(x_2) & \phi_1^{\#}(x_2) & \cdots & \phi_{n-1}^{\#}(x_2) \\ \vdots & \vdots & & \vdots \\ \phi_0^{\#}(x_n) & \phi_1^{\#}(x_n) & \cdots & \phi_{n-1}^{\#}(x_n) \end{bmatrix}. \tag{1.8}$$

There is a fast $O(n^2)$ Parker-Forney-Traub algorithm to invert usual Vandermonde matrices $V_P(x) = \left[\ x_i^{j-1}\ \right]$, see, e.g., [GO97] for many relevant references and some generalizations. This algorithm was extended in [HHR89], [CR93] to invert what we call three-term Vandermonde matrices $V_Q(x) = \left[\ Q_{j-1}(x_i)\ \right]$, i.e those involving polynomials $\{Q_k(x)\}$ satisfying three-term recurrence relations (see [GO94a] for the formulas and algorithms for Chebyshev-Vandermonde matrices).

Similarly, Szegő-Vandermonde matrices appear in the context of Gaussian quadrature on the unit circle [G82], in the context of Remez algorithm [R57], [PM72] in the case when the underlying basis consists of Szegő polynomials, elsewhere. So, the second problem is to extend the Parker-Forney-Traub algorithm to invert $V_{\Phi}^{\#}(x)$ in $O(n^2)$ operations.

1.2.3 Evaluation of a polynomial represented in the basis of Szegő polynomials.

The third problem is to extend the widely known Horner rule, i.e., to efficiently evaluate at a given point the polynomial

$$H(x) = b_n\phi_n^{\#}(x) + b_{n-1}\phi_{n-1}^{\#}(x) + \ldots + b_1\phi_1^{\#}(x) + b_0\phi_0^{\#}(x) \tag{1.9}$$

represented in the basis of Szegő polynomials. Szegő polynomials appear in a number of engineering applications, so the problem of evaluating (1.9) has been considered in [AGR93]. As we shall see in Sec. 4.3, there are limitations in the range of application of their two-term algorithm. To overcome this disadvantage, we suggest a modification as well as certain three-term alternative.

1.3 Contents and main results

We show that all three problems are closely related, and they can be solved via essentially the same family of fast algorithms. Note that the above mentioned two-term evaluation algorithm of [AGR93] solves the third problem only. Here we suggest a modified two-term and a new three-term algorithms that solve all three above problems. These algorithms are based on various recursions and realizations for what we suggest to call the Horner-Szegő polynomials.

Although all the proofs can be deduced algebraically, here we have chosen a different approach, and obtain the results via simple manipulations on the signal flow graphs. This choice not only clarifies the derivation and suggests a variety of possibilities to organize computational schemes, but it makes an interesting connection to many other results that have already been interpreted in the framework of transmission lines, e.g., to the results mentioned in Sec. 1.1.2.

The paper is structured as follows. In Sec. 2 a signal-flow-graph-interpretation of the classical Horner rule is used to provide a new derivation of the Parker-Traub-Forney algorithm for inversion of usual Vandermonde matrices. In this section we restrict the discussion to the simplest case of monomial basis to observe that transition from monomials to Horner polynomials corresponds to the transition from the observer to the controller canonical realizations. It is well-known that such a transition can be seen as just a reversal of the direction of the flow in the corresponding signal flow graph.

In Sec. 3 we generalize the above procedure and prove that the reversal of signal flow graphs works not only for monomials, but for arbitrary polynomials, thus leading to a clear and simple derivation for a generalization of a Horner rule, and to an inversion formula for polynomial Vandermonde matrices. We also indicate that such algorithms compute the eigenvectors for the corresponding confederate Hessenberg matrices. Then we specify these results for the basis of polynomials orthogonal on a real interval.

In Sec. 4 we turn to the most important case of Szegő polynomials, and list several corresponding recurrences, interpreting them in terms of different signal flow graphs. Then we just reverse these graphs, and read from them various recursions for the so-called Horner-Szegő polynomials. These recursions invert the Szegő-Vandermonde matrices and compute the eigenvectors of almost unitary Hessenberg matrices.

Finally, it is well-known that the moment matrix for Szegő polynomials have Toeplitz structure, displayed, e.g., in (5.5). In Sec. 5 we relate the Horner-Szegő polynomials to a certain inner product, for which the moment matrix is not Toeplitz, though it has a quasi-Toeplitz structure, in other words, it has displacement rank ≤ 2.

2 Inversion of Vandermonde matrices and signal flow graphs

In Sec. 3 and 4 we shall discuss the problems involving general polynomials and Szegő polynomials, resp. However, before doing so, we consider in this Section the simplest case of the power basis, to clarify the arguments relating algebraic properties to signal flow diagrams, to set the notations, and to recall several known facts.

2.1 Horner polynomials and inversion of Vandermonde matrices.

Horner recursion,

$$\tilde{p}_0(x) = a_n, \qquad \tilde{p}_k(x) = x\tilde{p}_{k-1}(x) + a_{n-k}, \tag{2.10}$$

is the standard method to evaluate the polynomial $H(x) = \tilde{p}_n(x)$,

$$H(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0 \tag{2.11}$$

at a given point. We denote Horner polynomials by $\tilde{P} = \{\tilde{p}_k(x)\}$, to meet a uniform designation to be introduced in Sec. 3.

Along with the evaluation, there are several other applications (for example, Horner used $\{\tilde{p}_k(x)\}$ to solve a single nonlinear equation); in this paper we shall mainly discuss a connection to the problem of inversion of Vandermonde matrices,

$$V_P(x) = \begin{bmatrix} 1 & x_1 & \cdots & x_1^{n-1} \\ 1 & x_2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \cdots & x_n^{n-1} \end{bmatrix}, \tag{2.12}$$

whose nodes $\{x_k\}$ are the zeros of $H(x)$. The connection is that the entries of $V_P(x)^{-1}$ are essentially the Horner polynomials:

$$V_P(x)^{-1} = \begin{bmatrix} \tilde{p}_{n-1}(x_1) & \tilde{p}_{n-1}(x_2) & \cdots & \tilde{p}_{n-1}(x_n) \\ \vdots & \vdots & & \vdots \\ \tilde{p}_1(x_1) & \tilde{p}_1(x_2) & \cdots & \tilde{p}_1(x_n) \\ \tilde{p}_0(x_1) & \tilde{p}_0(x_2) & \cdots & \tilde{p}_0(x_n) \end{bmatrix} \cdot \operatorname{diag}(c_1, \ldots c_n), \tag{2.13}$$

with

$$c_i = H'(x_i) = \frac{1}{\Pi_{\substack{k=1 \\ k\neq i}}^{n}(x_k - x_i)}. \tag{2.14}$$

The formula (2.13) leads to the widely known fast $O(n^2)$ Parker-Forney-Traub algorithm to compute $V_P(x)^{-1}$, see, e.g., [GO97] for many relevant references and connections.

2.2 Horner polynomials and eigenvectors of companion matrices

Let the nodes $\{x_1, \ldots, x_n\}$ be the zeros of $H(x) = b_0 + b_1 x + \ldots + b_n x^n$. It is widely known and can be easily checked that

$$V_P(x) C_P(H) = D(x) V_P(x), \qquad \text{where} \qquad D(x) = \operatorname{diag}\,(x_1, \ldots, x_n), \tag{2.15}$$

i.e, the columns of the inverse Vandermonde matrix store the eigenvectors of the companion matrix

$$C_P(H) = \begin{bmatrix} 0 & 0 & \cdots & 0 & -\frac{b_0}{b_n} \\ 1 & 0 & \cdots & 0 & -\frac{b_1}{b_n} \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & 1 & -\frac{b_{n-1}}{b_n} \end{bmatrix}. \tag{2.16}$$

Thus the eigenvectors of $C_P(H)$ can be computed in $O(n^2)$ operations via the Parker-Traub-Forney algorithm, see, e.g., Sec. 2.1.

2.3 Horner polynomials and signal flow graphs

To provide a signal-flow-graph-interpretation for the Horner polynomials we recall a standard way of realizing a linear time-invariant system

$$y(z) = H(z)u(z)$$

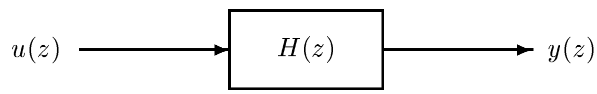
with a scalar transfer function of the form:

$$H(z) = d + c(zI - A)^{-1} b, \tag{2.17}$$

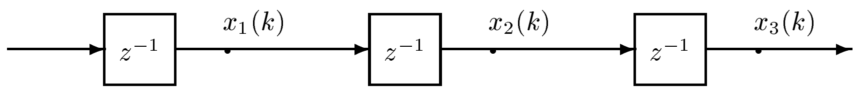
where A is an $n \times n$ matrix, c is a $1 \times n$ row, b is an $n \times 1$ column, and d is a scalar, see, e.g., [K80] or [OS89] for more details. Applying to the system $y(z) = H(z)u(z)$ the inverse z-transform, we obtain from (2.17) that the corresponding time-indexed input $u(k)$ and output $y(k)$ are related by

$$\begin{bmatrix} \mathbf{x}(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} A & b \\ c & d \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ u(k) \end{bmatrix} \tag{2.18}$$

where the coordinates of an auxiliary $n \times 1$ vector $\mathbf{x}(k)$ are called the *states*. To obtain a signal flow diagram



for (2.18) one first draws a delay-line



and takes the outputs of delay elements $\boxed{z^{-1}}$ as states $\{x_1(k), x_2(k), x_3(k)\}$. As we shall see in a moment, a signal flow graph for (2.18) is then obtained by interconnecting the states as suggested by the structure of the matrix $\begin{bmatrix} A & b \\ c & d \end{bmatrix}$ in (2.18), in particular,

- the row c provides the coefficients to read the output $y(k)$ from the states $\mathbf{x}(k)$, and
- the column b provides the coefficients to connect the input $u(k)$ to the states $\mathbf{x}(k+1)$.

We next recall two canonical ways of realizing a *whitening* filter with a polynomial transfer function

$$y(z) = H(z) \cdot u(z) = (a_n z^{-n} + a_{n-1} z^{-n+1} + \ldots + a_1 z^{-1} + a_0) \cdot u(z). \tag{2.19}$$

2.3.1 Observer-type realization

A direct calculation shows that, say, for $n = 3$, $H(z) = a_3 z^{-3} + a_2 z^{-2} + a_1 z^{-1} + a_0$, we have

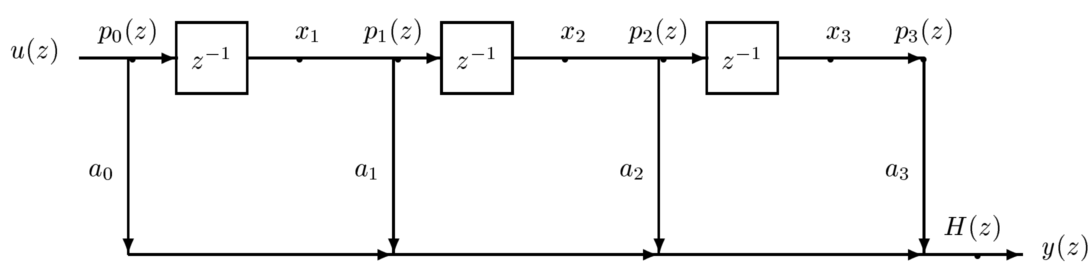
$$H(z) = a_0 + \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} (zI - \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix})^{-1} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{2.20}$$

so that the equivalent state-space description (2.18),

$$\begin{aligned} \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u(k) \\ y(k) &= \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + a_0 \, u(k) \end{aligned} \tag{2.21}$$

obtained from (2.20) via the inverse z-transform can be realized as shown in Fig. 2.

<ins>Figure 2.</ins> Observer-type realization.



As above, in Fig. 2 the outputs of delay elements are taken as state variables $\{x_k\}$. The numbers $\{a_k\}$ are simply scaling coefficients, so that each signal passing through the corresponding branch is multiplied by a_k. Finally, $\{p_k(z)\}$ denote *partial transfer functions*, i.e. the ones from the input of the line to the *input of the k-th delay element.* Of course, an examination of the signal flow graph in Fig. 2 reveals immediately that the graph takes a linear combination of $p_k(z) = z^{-k}$ to indeed realize $H(z) = a_3 z^{-3} + a_2 z^{-2} + a_1 z^{-1} + a_0$. This realization is canonical, and it is called *observer-type*, see, e.g., [K80], [F96] because the output is read from the state variables through the taps. We next recall another canonical realization to be used in what follows.

2.3.2 Controller-type realization

For any realization of a scalar transfer function $H(z) = d + c(zI - A)^{-1}b$, one more realization can be immediately obtained by taking transposes, and reversing the indexing of the state variables using the reverse identity matrix $\tilde{I}$:

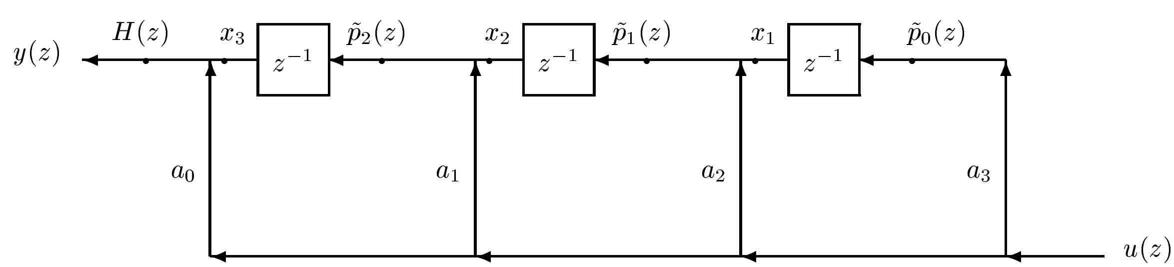
$$H(z) = d + b^T \tilde{I}(zI - \tilde{I}A^T\tilde{I})^{-1}\tilde{I}c^T, \tag{2.22}$$

Clearly, transposition does not change a scalar function $H(z)$, but the inner state-space structure is, of course, different, say for $n = 3$ we have

$$\begin{aligned}
\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} a_3 \\ a_2 \\ a_1 \end{bmatrix} u(k) \\
y(k) &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + a_0 \, u(k).
\end{aligned} \tag{2.23}$$

The signal flow graph for (2.23) is given in Fig. 3, and one sees that it is obtained from the one in Fig. 2 by simply reversing the direction of the flow for each branch. A new realization obtained in this way is called *transposed* [OS89] or *dual* [K80]. Recall that it is an elementary fact in system theory, i.e., the Mason's rules [MZ60], that such a reversal of the flow in a signal flow graph (passing to the transposed (dual) system) does not change the overall transfer function $H(z)$.

Figure 3. Controller-type realization.



The realization in Fig. 3 is also canonical, and it is called *controller-type*, see, e.g., [K80], [F96] because the input, $u(z)$, is directly connected to each of the state variables.

2.3.3 The standard Horner polynomials

Now let us examine the partial transfer function for the two above realizations. For any realization, $H(z) = d + c(zI - A)^{-1}b$, the vector of *partial transfer functions* from the input of the line to input of the k-th delay element is given by

$$\begin{bmatrix} H_1(z) \\ \vdots \\ H_n(z) \end{bmatrix} = z(zI - A)^{-1}b, \tag{2.24}$$

cf. with the arguments at the beginning of Sec. 2.2. For example, for the observer-type realization in Fig. 2 we have

$$\begin{bmatrix} p_0(z) \\ p_1(z) \\ p_2(z) \end{bmatrix} = \begin{bmatrix} 1 \\ z^{-1} \\ z^{-2} \end{bmatrix}, \tag{2.25}$$

whereas for the controller-type realization in Fig. 3

$$\begin{bmatrix} \tilde{p}_0(z) \\ \tilde{p}_1(z) \\ \tilde{p}_2(z) \end{bmatrix} = \begin{bmatrix} a_3 \\ a_3 z^{-1} + a_2 \\ a_3 z^{-2} + a_2 z^{-1} + a_1 \end{bmatrix}. \tag{2.26}$$

Thus one easily identifies the power basis $P = \{1, z^{-1}, z^{-2}\}$ as the partial transfer functions to the *inputs of the delay elements* for the observer-type realization; and the Horner polynomials $\tilde{P} = \{\tilde{p}_0(z), \tilde{p}_1(z), \tilde{p}_2(z)\}$ are the partial transfer functions to the *inputs of the delay elements* in the controller-type realization.

These observations are, of course, trivial, but the point is that this very simple procedure works for any polynomial basis, as shown in the next section.

3 General Horner-like polynomials

3.1 Obtaining the Horner-like polynomials

In this section we shall justify the following procedure.

Procedure 3.1 Obtaining the Horner-like polynomials.

- *Given a recursion for the polynomials* $R = \{r_0(x), r_1(x), \ldots, r_n(x)\}$, *where* $\deg r_k(x) = k$, *and a polynomial*
$$H(z) = b_n r_n(z) + b_{n-1} r_{n-2}(z) + \ldots + b_1 r_1(z) + b_0 r_0(z), \tag{3.1}$$
- *Draw a signal flow graph for the linear time-invariant system with the overall transfer function* $H(z)$, *and such that* $r_k(z^{-1})$ *are the partial transfer functions to the input of the* k*-th delay element for* $k = 1, 2, \ldots, n-1$.
- *Pass to the transposed (dual) system by reversing the direction of the flow.*
- *Identify the* Horner-like polynomials $\tilde{R} = \{\tilde{r}_k(z^{-1})\}$ *as the partial transfer functions from the input of the line to the inputs of the delay elements.*
- *Read from the obtained signal flow graph a recursion for* $\tilde{R} = \{\tilde{r}_k(z^{-1})\}$.

As was noted, the reversal of the signal flow graph does not change the overall transfer function $H(z)$, so we shall certainly obtain a new recursion for evaluation of $H(x) = \tilde{r}_n(z^{-1})$, generalizing the usual Horner rule. The question, however, is: how these Horner-like polynomials $\tilde{R} = \{\tilde{r}_k(x)\}$ can be used to invert a polynomial Vandermonde matrix in

$$V_R(x) = \begin{bmatrix} r_0(x_1) & r_1(x_1) & \cdots & r_{n-1}(x_1) \\ r_0(x_2) & r_1(x_2) & \cdots & r_{n-1}(x_2) \\ \vdots & \vdots & & \vdots \\ r_0(x_n) & r_1(x_n) & \cdots & r_{n-1}(x_n) \end{bmatrix}, \tag{3.2}$$

(as we know, the usual Horner polynomials are essentially the entries of the inverse of the usual Vandermonde matrix, see, e.g., (2.13)). The answer to this question is given next.

3.2 Inversion of polynomial Vandermonde matrices

Proposition 3.2 *Let $V_R(x)$ be a polynomial Vandermonde matrix in (3.2) involving the first n polynomials of $R = \{r_0(x), r_1(x), \dots, r_n(x)\}$ and whose nodes $\{x_k\}$ are the zeros of $H(x)$. Then the inverse of $V_R(x)$ is given by*

$$V_R(x)^{-1} = \begin{bmatrix} \tilde{r}_{n-1}(x_1) & \tilde{r}_{n-1}(x_2) & \cdots & \tilde{r}_{n-1}(x_n) \\ \vdots & \vdots & & \vdots \\ \tilde{r}(x_1) & \tilde{r}_1(x_2) & \cdots & \tilde{r}r_1(x_n) \\ \tilde{r}_0(x_1) & \tilde{r}_0(x_2) & \cdots & \tilde{r}_0(x_n) \end{bmatrix} \cdot \operatorname{diag}(c_1, \dots c_n), \tag{3.3}$$

with $c_i = H'(x_i) = \frac{1}{\prod_{\substack{k=1 \\ k\neq i}}^{n}(x_k - x_i)}$, and where the Horner-like polynomials $\tilde{R} = \{\tilde{r}_0(x), \tilde{r}_1(x), \dots, \tilde{r}_n(x)\}$ are obtained via the procedure 3.1.

Proof. In this proof we use the notations introduced in Sec. 2.2.1 and 2.2.2, and consider the observer-type and the controller-type realizations,

$$H_P(z) = d + c(zI - A)^{-1}b, \qquad H_{\tilde{P}}(z) = d + (b^T\tilde{I})(zI - \tilde{I}A^T\tilde{I})^{-1}(\tilde{I}c^T). \tag{3.4}$$

Since these realizations are transposed (dual) to each other, they have the same transfer function. So we use the dual notation $H_P(z) = H_{\tilde{P}}(z)$ for this transfer function to reflect the fact that the partial transfer functions for these realizations are the power basis, P, and the usual Horner polynomials $\tilde{P}$, resp. For this simplest case we know from (2.13) that

$$\begin{bmatrix} 1 & x_k & \dots & x_k^{n-1} \end{bmatrix} \tilde{I} \begin{bmatrix} \tilde{p}_0(x_j) \\ \tilde{p}_1(x_j) \\ \vdots \\ \tilde{p}_{n-1}(x_j) \end{bmatrix} = \delta_{kj}c_k. \tag{3.5}$$

Now suppose that we have another realization, $H_R(z)$ for the same transfer function, for which the partial transfer functions,

$$\begin{bmatrix} r_0(z) \\ r_1(z) \\ r_2(z) \\ \vdots \\ r_{n-1}(z) \end{bmatrix} = F \cdot \begin{bmatrix} 1 \\ z^{-1} \\ z^{-2} \\ \vdots \\ z^{-n+1} \end{bmatrix} \tag{3.6}$$

satisfy $\deg r_k(z) = k$, so that F is a lower triangular matrix. Clearly,

$$\begin{aligned} H_R(z) &= d + cF(zI - F^{-1}AF)^{-1}F^{-1}b, \\ H_{\tilde{R}}(z) &= d + (b^T\tilde{I})(\tilde{I}F^{-T}\tilde{I})(zI - (\tilde{I}F^T\tilde{I})(\tilde{I}A^T\tilde{I})(\tilde{I}F^{-T}\tilde{I}))^{-1}(\tilde{I}F^T\tilde{I})(\tilde{I}c^T) \end{aligned} \tag{3.7}$$

Now, using the formula (2.24) to obtain the partial transfer functions for the two realizations in (3.4) and (3.6), ones easily sees that the Horner-like polynomials are obtained from the usual Horner polynomials via

$$\begin{bmatrix} \tilde{r}_0(z) \\ \tilde{r}_1(z) \\ \tilde{r}_2(z) \\ \vdots \\ \tilde{r}_{n-1}(z) \end{bmatrix} = \tilde{I}F^{-T}\tilde{I} \begin{bmatrix} \tilde{p}_0(z) \\ \tilde{p}_1(z) \\ \tilde{p}_2(z) \\ \vdots \\ \tilde{p}_{n-1}(z) \end{bmatrix} \tag{3.8}$$

This relation, (3.5), and (3.6) imply

$$\begin{bmatrix} r_0(x_k) & r_1(x_k) & \dots & r_{n-1}(x_k) \end{bmatrix} \tilde{I} \begin{bmatrix} \tilde{r}_0(x_j) \\ \tilde{r}_1(x_j) \\ \vdots \\ \tilde{r}_{n-1}(x_j) \end{bmatrix} = \delta_{kj}c_k, \tag{3.9}$$

which proves the desired (3.3).

3.3 A Hessenberg eigenvector problem

As we recalled in Sec. 2.2, the columns of the inverse of the usual Vandermonde matrix $V_P(x)^{-1}$ store the eigenvectors of the corresponding companion matrix $C_P(H)$, see, e.g., (2.15). Clearly, a companion matrix (2.16) is a special case (corresponding to the power basis $P = \{1, x, x^2, \dots, x^n\}$) of the more general confederate matrix $C_Q(H)$ defined in (1.5). It can be easily verified that for $C_Q(H)$ a generalization of (2.15) holds:

$$V_Q(x)C_Q(H) = D(x)V_Q(x),$$

implying that the columns of the inverse of polynomial Vandermonde matrix $V_Q(x)^{-1}$ store the eigenvectors of the confederate matrix $C_Q(H)$. Therefore, the results of Sec. 3.2 mean that the procedure 3.1 for obtaining the Horner-like polynomials solves the eigenvector problem for general confederate matrices.

3.4 An example: n-term recurrence relations

The conclusion of the above discussion is that the procedure of reversing a signal flow graph, described at the beginning of this section, in fact suggests a surprisingly simple way of derivation for efficient algorithms for inversion of polynomial Vandermonde matrices, for evaluation of polynomials, as well as for computing eigenvectors of Hessenberg matrices. A specification of this method to Szegő polynomials will be addressed in Sec. 4. However, before doing so, we next show how this works for the general case when the polynomials R are given by the most general n-term recurrence relations

$$r_0(x) = \alpha_0,$$

$$r_k(x) = \alpha_k x r_{k-1}(x) - a_{k-1,k} r_{k-1}(x) - a_{k-2,k} r_{k-2}(x) - \dots - a_{1,k} r_1(x) - a_{0,k} r_0(x). \tag{3.10}$$

Proposition 3.3 *Let the basis $R = \{r_0(x), \dots, r_n(x)\}$ be defined by (3.10). Then the Horner-like polynomials $\tilde{R} = \{\tilde{r}_0(x), \dots, \tilde{r}_n(x)\}$ satisfy*

$$\tilde{r}_0(x) = \tilde{\alpha}_0 b_n,$$

$$\tilde{r}_k(x) = \tilde{\alpha}_k x \tilde{r}_{k-1}(x) - \tilde{a}_{k-1,k} \tilde{r}_{k-1}(x) - \tilde{a}_{k-2,k} \tilde{r}_{k-2}(x) - \dots - \tilde{a}_{1,k} \tilde{r}_1(x) - \tilde{a}_{0,k} \tilde{r}_0(x) + b_{n-k}, \tag{3.11}$$

where

$$\tilde{\alpha}_k = \alpha_{n-k}, \qquad (k = 0, 1, ..., n), \tag{3.12}$$

and

$$\tilde{a}_{k,j} = \frac{\alpha_{n-j}}{\alpha_{n-k}} a_{n-j,n-k} \qquad (k = 0, 1, ...n-1; j = 1, 2, ..., n). \tag{3.13}$$

Proof. Follows immediately from the arguments in this section and comparing the signal flow graphs for R and $\tilde{R}$, drawn in Fig. 4 and 5, resp., for the case $n = 3$.

Figure 4. n-term recurrence relations

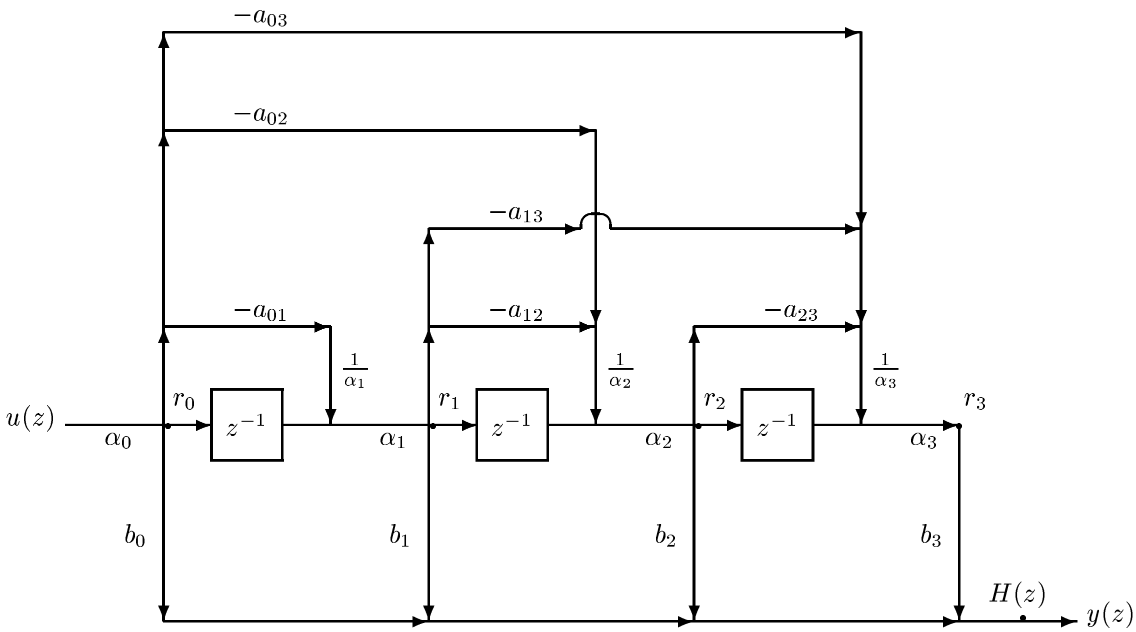
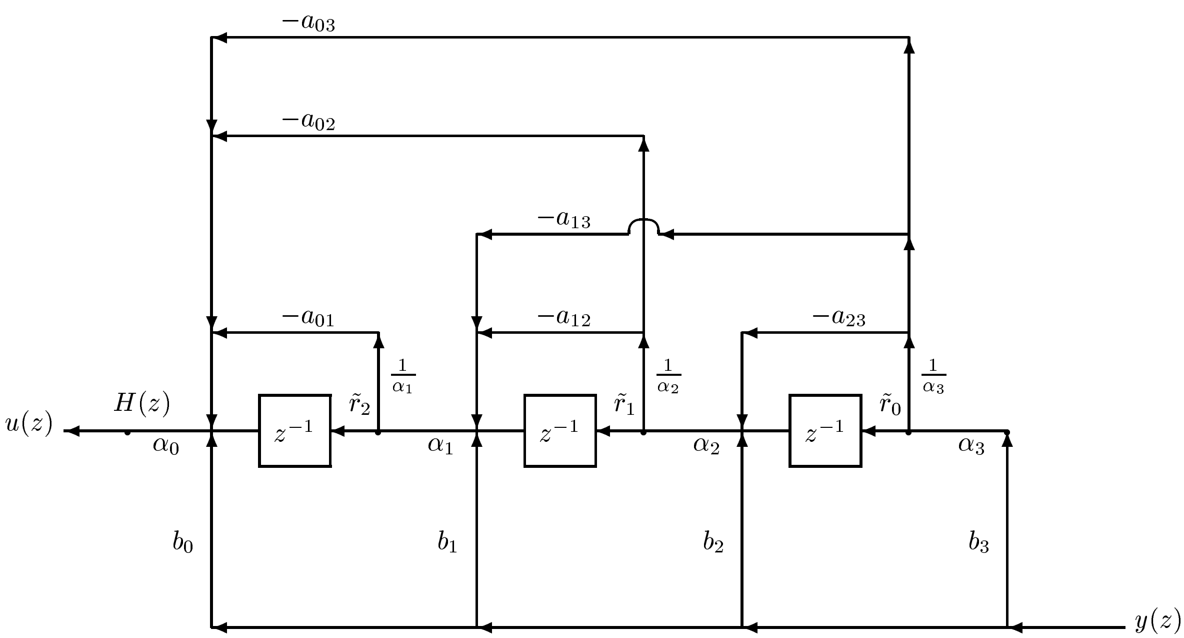


Figure 5. Dual to the filter of Fig. 4.



3.5 A special case: the Clenshaw rule and inversion of three-term Vandermonde matrices

Of course, the case when polynomials R satisfy the three-term recurrence relations,

$$r_k(x) = (\alpha_k \cdot x - \beta_k) \cdot r_{k-1}(x) - \gamma_k \cdot r_{k-1}(x), \tag{3.14}$$

is one of the most important cases, since the class (3.14) contains polynomials orthogonal on a real interval. The Clenshaw rule,

$$\tilde{r}_k(x) = \alpha_{n-k} x \tilde{r}_{k-1}(x) - \frac{\alpha_{n-k}}{\alpha_{n-k+1}} \beta_{n-k+1} \tilde{r}_{k-1}(x) - \frac{\alpha_{n-k}}{\alpha_{n-k+2}} \gamma_{n-k+2} \tilde{r}_{k-2}(x) + b_{n-k} \tag{3.15}$$

is a well-known extension of the Horner rule to evaluate $H(z) = b_n r_n(x) + b_{n-1} r_{n-1}(x) + \ldots + b_0 r_0(x)$. One sees that (3.15) is just a special case of (3.11). According to Proposition 3.2, this Clenshaw recursion not only evaluates $H(x)$, but it also inverts the corresponding three-term Vandermonde matrix, thus being closely related to the algorithms for this purpose specified in [HHR89], [GO94a], [CR93].

4 Horner-Szegő polynomials, inversion of Szegő-Vandermonde matrices and eigenvector computation for almost unitary Hessenberg matrices

4.1 Various realizations for the Szegő polynomials

Here we consider a representation in the basis of the Szegő polynomials

$$H(z^{-1}) = b_n \phi_n^{\#}(z^{-1}) + b_{n-1} \phi_{n-1}^{\#}(z^{-1}) + \ldots + b_0 \phi_0^{\#}(z^{-1}). \tag{4.1}$$

Szegő polynomials $\Phi^{\#} = \{\phi_0^{\#}(x), \ldots, \phi_n^{\#}(x)\}$ are completely described by μ_0 and n reflection coefficients $\{\rho_1, \ldots, \rho_n\}$, see, e.g., (1.3). In the next lemma we use several recurrence relations for $\Phi^{\#}$ to draw the signal flow graphs realizing $H(z^{-1})$.

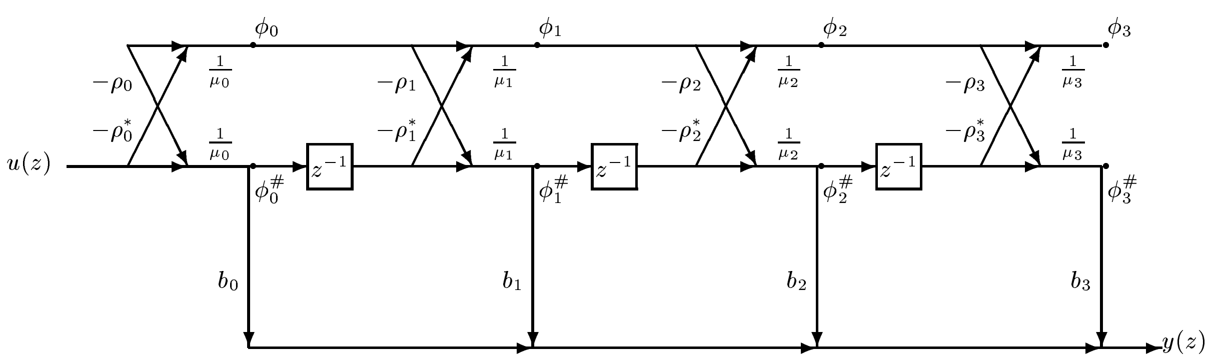
Lemma 4.1 *The polynomial in (4.1) can be realized using of the recurrences listed next.*

1. Two-term recurrence relations. *[GS58], [G48]*

$$\begin{bmatrix} \phi_0(x) \\ \phi_0^{\#}(x) \end{bmatrix} = \frac{1}{\mu_0} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad \begin{bmatrix} \phi_{k+1}(x) \\ \phi_{k+1}^{\#}(x) \end{bmatrix} = \frac{1}{\mu_{k+1}} \begin{bmatrix} 1 & -\rho_{k+1}^* \\ -\rho_{k+1} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & x \end{bmatrix} \begin{bmatrix} \phi_k(x) \\ \phi_k^{\#}(x) \end{bmatrix}, \tag{4.2}$$

Using (4.2), the polynomial in (4.1) can be realized as shown in Fig. 6

Figure 6. The Markel-Gray [MG76] whitening filter (with $\rho_0 = -1$).



2. Three-term recurrence relations. *[G48]*

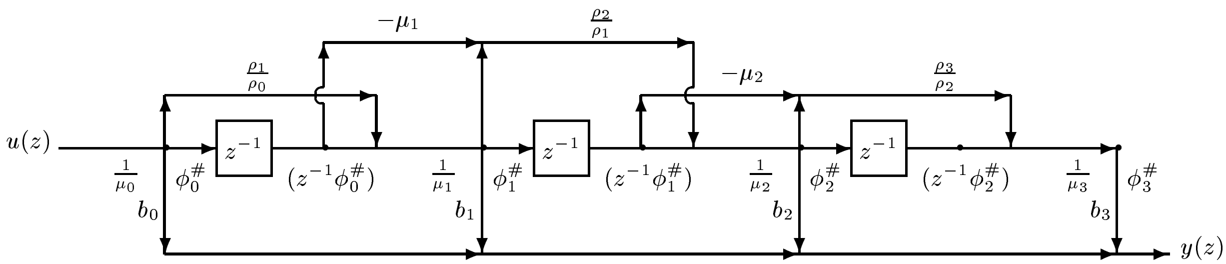
$$\phi_0^{\#}(x) = \frac{1}{\mu_0}, \qquad \phi_1^{\#}(x) = \frac{1}{\mu_1}(x \cdot \phi_0^{\#}(x) + \rho_1 \rho_0^* \cdot \phi_0^{\#}(x)),$$

where $\rho_0 = -1$, *and*

$$\phi_k^{\#}(x) = [\frac{1}{\mu_k} \cdot x + \frac{\rho_k}{\rho_{k-1}} \frac{1}{\mu_k}] \cdot \phi_{k-1}^{\#}(x) - \frac{\rho_k}{\rho_{k-1}} \frac{\mu_{k-1}}{\mu_k} \cdot x \cdot \phi_{k-2}^{\#}(x). \tag{4.3}$$

Using (4.3), the polynomial in (4.1) can be realized as shown in Fig. 7.

Figure 7. Three-term realization.



3. n-term recurrence relations. *(cf. with the structure in (1.7))*

$$\phi_k^{\#}(x) = \frac{1}{\mu_k}[x \cdot \phi_{k-1}^{\#}(x) + \rho_k \ \rho_{k-1}^* \cdot \phi_{k-1}^{\#}(x) + \rho_k\mu_{k-1}\rho_{k-2}^* \cdot \phi_{k-2}^{\#}(x) + \rho_k\mu_{k-1}\mu_{k-2}\rho_{k-3}^* \cdot \phi_{k-3}^{\#}(x) + ...$$

$$... + \rho_k\mu_{k-1}\mu_{k-2} \cdot ... \cdot \mu_2\rho_1^* \cdot \phi_1^{\#}(x) + \rho_k\mu_{k-1}\mu_{k-2} \cdot ... \cdot \mu_1\rho_0^* \cdot \phi_0^{\#}(x)], \tag{4.4}$$

The realization for (4.4) can be drawn as in Fig. 4.

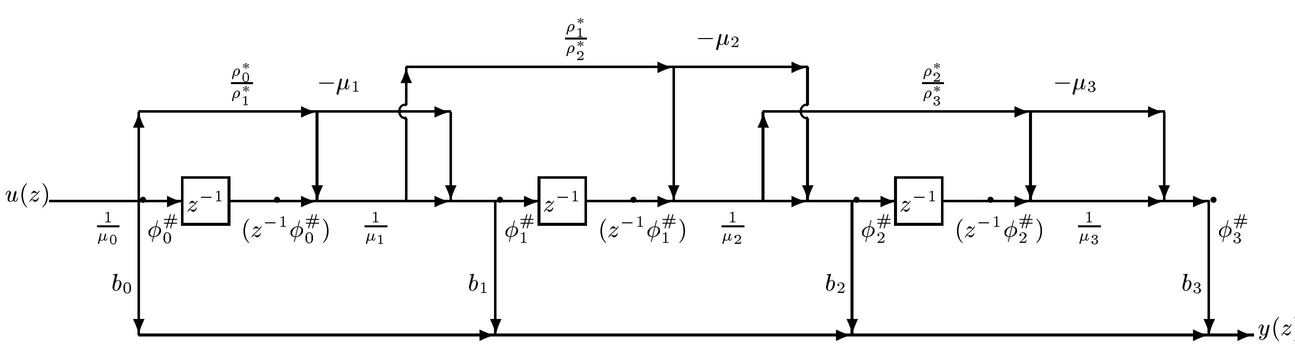
4. Shifted n-term recurrence relations.

$$\phi_k^{\#}(x) = \frac{1}{\mu_k}x\phi_{k-1}^{\#}(x) + \frac{\rho_k\rho_{k-1}^*}{\mu_k\mu_{k-1}}x\phi_{k-2}^{\#}(x) + \frac{\rho_k\rho_{k-2}^*}{\mu_k\mu_{k-1}\mu_{k-2}}x\phi_{k-3}^{\#}(x) + \dots$$

$$+\frac{\rho_k\rho_1^*}{\mu_k\mu_{k-1}\dots\mu_1}x\phi_0^{\#}(x) + \frac{\rho_k\rho_0^*}{\mu_k\mu_{k-1}\dots\mu_1}\phi_0^{\#}(x) \tag{4.5}$$

Using (4.5), the polynomial in (4.1) can be realized as shown in Fig. 8.

Figure 8. Shifted n-term realization.



Here we have to warn that there is no one-to-one-correspondence between the recursion formulas and particular realizations, for example, the n-term recursions (4.5) can also be easily read from the signal flow graph in Fig. 6.

4.2 Dual filters and the recursions for the Horner-Szegő polynomials

The next statement presents analogous recursions and realizations for the Horner-Szegő polynomials.

Proposition 4.2 *Let $H(x)$ be given by (4.1), and Szegő polynomials $\Phi^{\#} = \{\phi_k^{\#}(x)\}$ be given by one of the recursions of lemma 4.1. Then the Horner-Szegő polynomials $\tilde{\Phi}^{\#} = \{\tilde{\phi}_k^{\#}(x)\}$ can be obtained via one of the following recursions, where $\tilde{\rho}_k = \rho^*_{n-k}$ for $k = 0, 1, ..., n$ and $\tilde{\mu}_k = \sqrt{1 - |\tilde{\rho}_k|^2}$, $\tilde{\mu}_n = 1$ (and if $|\tilde{\rho}_0| = 1$, then $\tilde{\mu}_0 := 1$).*

1. Two-term recurrence relations.

$$\begin{bmatrix} \tilde{\phi}_0(x) \\ \tilde{\phi}_0^{\#}(x) \end{bmatrix} = \frac{1}{\tilde{\mu}_0} \begin{bmatrix} -\tilde{\rho}_0 b_n \\ b_n \end{bmatrix}, \qquad \begin{bmatrix} \tilde{\phi}_k(x) \\ \tilde{\phi}_k^{\#}(x) \end{bmatrix} = \frac{1}{\tilde{\mu}_k} \begin{bmatrix} 1 & -\tilde{\rho}_k^* \\ -\tilde{\rho}_k & 1 \end{bmatrix} \begin{bmatrix} \tilde{\phi}_{k-1}(x) \\ x\tilde{\phi}_{k-1}^{\#}(x) + b_{n-k} \end{bmatrix}, \tag{4.6}$$

2. Three-term recurrence relations.

$$\tilde{\phi}_0^{\#}(x) = \frac{b_n}{\tilde{\mu}_0}, \qquad \tilde{\phi}_1^{\#}(x) = \{\frac{1}{\tilde{\mu}_1} \cdot x\tilde{\phi}_0^{\#}(x) - \frac{\tilde{\rho}_1\tilde{\rho}_0^*}{\tilde{\mu}_1}\tilde{\phi}_0^{\#}(x)\} + \frac{b_{n-1}}{\tilde{\mu}_1}.$$

$$\tilde{\phi}_k^{\#}(x) = \{[\frac{1}{\tilde{\mu}_k} \cdot x + \frac{\tilde{\rho}_k}{\tilde{\rho}_{k-1}} \frac{1}{\tilde{\mu}_k}]\tilde{\phi}_{k-1}^{\#}(x) - \frac{\tilde{\rho}_k}{\tilde{\rho}_{k-1}} \frac{\tilde{\mu}_{k-1}}{\tilde{\mu}_k} \cdot x \cdot \tilde{\phi}_{k-2}^{\#}(x)\} - \frac{b_{n-k} - b_{n-k+1}\tilde{\mu}_{k-1}\frac{\tilde{\rho}_k}{\tilde{\rho}_{k-1}}}{\tilde{\mu}_k}. \tag{4.7}$$

3. n-term recurrence relations.

$$\tilde{\phi}_k^{\#}(x) = \frac{1}{\tilde{\mu}_k} x \cdot \tilde{\phi}_{k-1}^{\#}(x) + \tilde{\rho}_k\ \tilde{\rho}_{k-1}^* \cdot \tilde{\phi}_{k-1}^{\#}(x) + \tilde{\rho}_k \mu_{k-1} \tilde{\rho}_{k-2}^* \cdot \tilde{\phi}_{k-2}^{\#} + \tilde{\rho}_k \tilde{\mu}_{k-1}\tilde{\mu}_{k-2}\tilde{\rho}_{k-3}^* \cdot \tilde{\phi}_{k-3}^{\#} + ...$$

$$... + \tilde{\rho}_k\tilde{\mu}_{k-1}\tilde{\mu}_{k-2} \cdot ... \cdot \tilde{\mu}_2\tilde{\rho}_1^* \cdot \tilde{\phi}_1^{\#}(x) + \tilde{\rho}_k\tilde{\mu}_{k-1}\tilde{\mu}_{k-2} \cdot ... \cdot \tilde{\mu}_1\tilde{\rho}_0 \cdot \tilde{\phi}_0^{\#}(x) + b_{n-k}], \tag{4.8}$$

4. Shifted n-term recurrence relations.

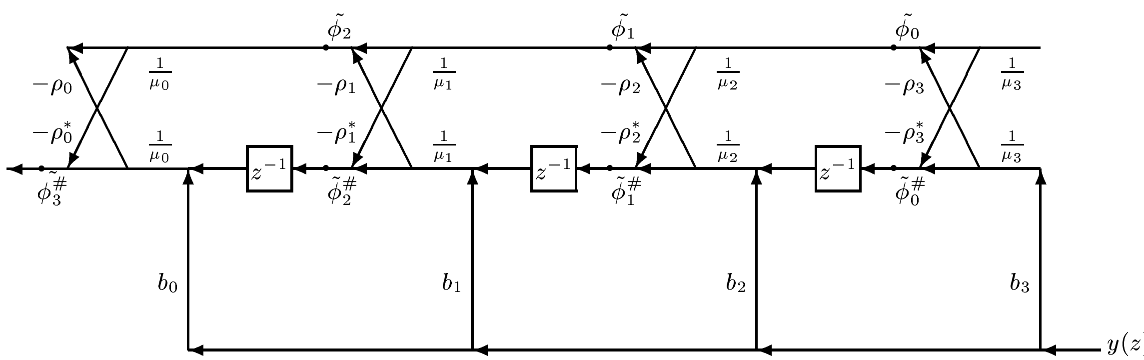
$$\tilde{\phi}_k^{\#}(x) = \frac{1}{\tilde{\mu}_k}(x\tilde{\phi}_{k-1}^{\#}(x) + b_{n-k+1}) + \frac{\tilde{\rho}_k\tilde{\rho}_{k-1}^*}{\tilde{\mu}_k\tilde{\mu}_{k-1}}(x\tilde{\phi}_{k-2}^{\#}(x) + b_{n-k+2}) + \frac{\tilde{\rho}_k\tilde{\rho}_{k-2}^*}{\tilde{\mu}_k\tilde{\mu}_{k-1}\tilde{\mu}_{k-2}}(x\tilde{\phi}_{k-3}^{\#}(x) + b_{n-k+2}) + \dots$$

$$+ \frac{\tilde{\rho}_k\tilde{\rho}_1^*}{\tilde{\mu}_k\tilde{\mu}_{k-1}\dots\tilde{\mu}_1}(x\tilde{\phi}_0^{\#}(x) + b_{n-1}) + \frac{\tilde{\rho}_k\tilde{\rho}_0^*}{\tilde{\mu}_k\tilde{\mu}_{k-1}\dots\tilde{\mu}_1}\tilde{\phi}_0^{\#}(x). \tag{4.9}$$

Proof. The formula for the general n-term recurrence relations was obtained in proposition 3.3, and (4.8) is its specification.

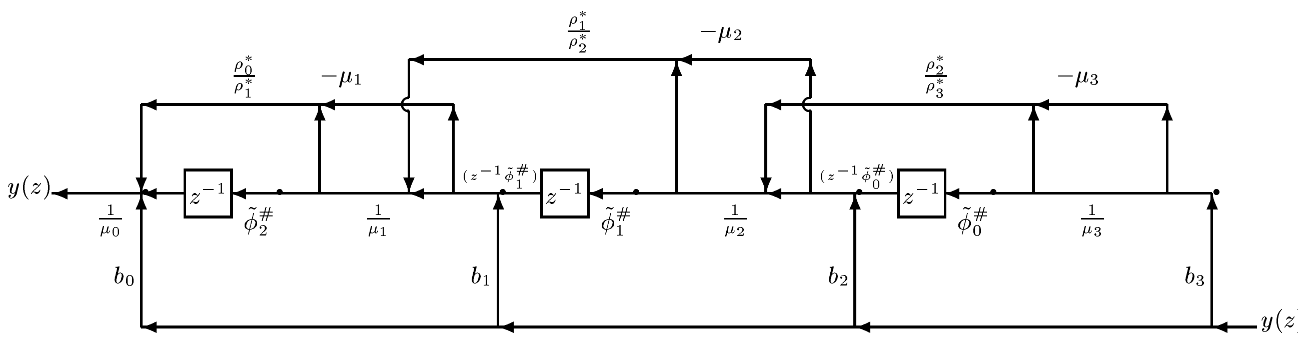
To obtain (4.6) we follow the Procedure 3.1, and reverse the direction of the flow in Fig. 6, obtaining the dual signal flow graph shown in Fig. 9. The formulas (4.6) are easily read from the structure in Fig. 9.

Figure 9. Dual to the Markel-Gray filter of Fig 6.



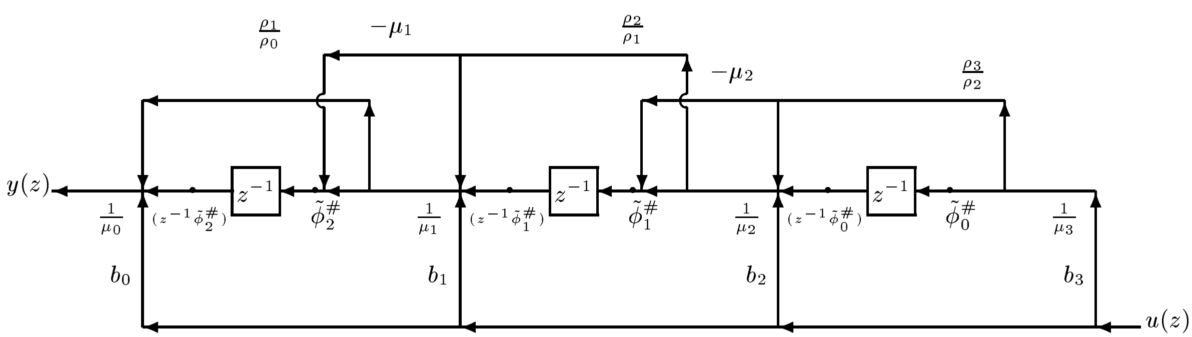
To obtain the two-term recursions for the Horner-Szegő polynomials we used the two-term recursions for the original Szegő polynomials. However, one sees that there is no such symmetry for the other filter structures in Lemma 4.1 and Proposition 4.2. Indeed, for the structure in Fig. 7, we have two outcoming branches and one incoming branch between two consecutive delay elements. Therefore, after reversing of the flow we obtain a different structure with two incoming and one outcoming branch, like the structure in Fig. 8. Therefore, the structures in Fig. 7 and 8 are essentially dual to each other, and to obtain the three-term recursions (4.6) for the Horner-Szegő polynomials we transpose the signal flow graph in Fig. 8, i.e., corresponding to the shifted n-term recursions for the original Szegő polynomials, arriving at the structure shown in Fig. 10.

Figure 10. Dual to the shifted n-term realization of Fig. 8, giving rise to the three-term recursions for the Horner-Szegő polynomials.



Similarly, to obtain the shifted n-term recursions for the Horner-Szegő polynomials, we transpose the signal flow graph in Fig. 7 which corresponds to the three-term recursions for the original Szegő polynomials.

Figure 11.Dual to the three-term realization of Fig. 7, giving rise to the shifted n-term recursions for the Horner-Szegő polynomials.



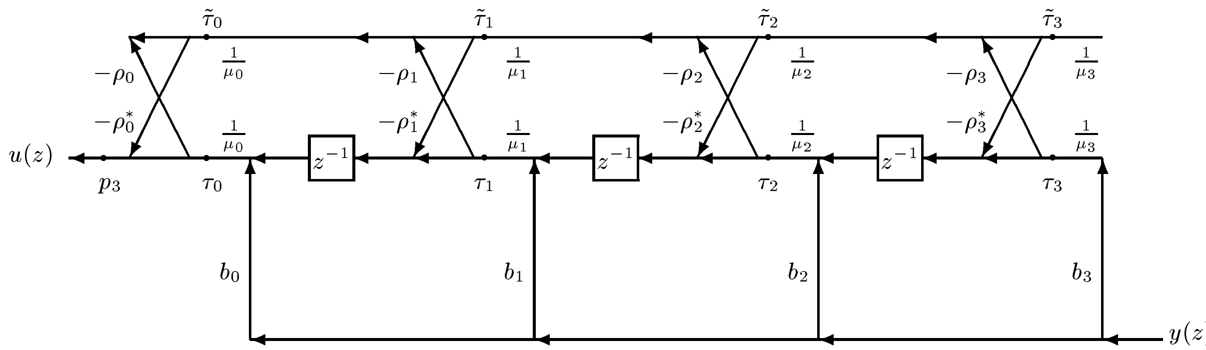
4.3 The Ammar-Gragg-Reichel evaluation algorithm

In [AGR93] a different recursion,

$$\begin{bmatrix} \tau_n \\ \tilde{\tau}_n \end{bmatrix} = \begin{bmatrix} \frac{b_n}{\mu_n} \\ 0 \end{bmatrix}, \qquad \begin{bmatrix} \tau_k \\ \tilde{\tau}_k \end{bmatrix} = \frac{1}{\mu_k} \begin{bmatrix} b_k + x(\tau_{k+1} + \rho_{k+1}^* \tilde{\tau}_{k+1}) \\ \rho_{k+1}\tau_{k+1} + \tilde{\tau}_{k+1} \end{bmatrix}, \qquad H(x) = \tau_0 + \tilde{\tau}_0, \tag{4.10}$$

to evaluate $H(x)$ in (4.1) was deduced algebraically. The results in Sec 3, 4 allow us to interpret the recursion (4.10) via the signal flow graph in Fig. 9. Indeed, as shown in Fig. 12 the AGR algorithm simply uses different intermediate points. This observation indicates that the transmission line model proposes a variety of different possibilities to organize the computational scheme for evaluation of the overall transfer function $H(z)$.

Figure 12. The Ammar-Gragg-Reichel recursion.



At the same time, the AGR algorithm does not seem to be a complete analog of the Horner method, because the polynomials $\tau_k(z^{-1})$ are not chosen to be the partial transfer functions to the inputs of the delay elements, as required in Procedure 3.1. As a result, the recursion (4.10) does not invert the corresponding Szegő-Vandermonde matrix, and therefore it does not computes the eigenvectors for almost unitary Hessenberg matrices. The alternative two-term recursion (4.6) as well as three other recursions in Proposition 4.2 do so, and therefore they seem to be more appropriate counterparts of the Horner method.

4.4 Several useful formulas

Here we formulate several more descriptions for the Horner-Szegő polynomials, which can be useful by occasion.

- The Horner-Szegő polynomials can be defined by

$$\frac{H(x)-H(y)}{x-y} = \phi_0^{\#}(x)\tilde{\phi}_{n-1}^{\#}(y) + \phi_1^{\#}(x)\tilde{\phi}_{n-2}^{\#}(y) + \cdots + \phi_{n-1}^{\#}(x)\tilde{\phi}^{\#}(y),$$

  see, e.g. Sec. 3 in [KO97a].

- An examination of the structure in Fig. 6 allows one to easily identify the state space structure, see, e.g., [M74], [ML80]. Using formula (2.24) we obtain the following formula for Szegő polynomials,

$$\begin{bmatrix} \phi_0^{\#}(z^{-1}) \\ \phi_1^{\#}(z^{-1}) \\ \phi_2^{\#}(z^{-1}) \\ \vdots \\ \phi_n^{\#}(z^{-1}) \end{bmatrix} = z(zI - ZA)^{-1} \cdot \begin{bmatrix} \frac{1}{\mu_0} \\ \frac{\rho_1\rho_0^*}{\mu_1\mu_0} \\ \frac{\rho_2\rho_0^*}{\mu_2\mu_1\mu_0} \\ \vdots \\ \frac{\rho_n\rho_0^*}{\mu_n\cdot\ldots\cdot\mu_0} \end{bmatrix},$$

  where Z is the lower shift matrix (small z is, of course, a variable), and

$$A = \begin{bmatrix} \frac{1}{\mu_1} & 0 & \cdots & \cdots & \cdots & 0 \\ \frac{\rho_1^*\rho_2}{\mu_1\mu_2} & \frac{1}{\mu_2} & 0 & & & \vdots \\ \frac{\rho_1^*\rho_3}{\mu_1\mu_2\mu_3} & \frac{\rho_2^*\rho_3}{\mu_2\mu_3} & \frac{1}{\mu_3} & 0 & & \vdots \\ \frac{\rho_1^*\rho_4}{\mu_1\cdot\ldots\cdot\mu_4} & \frac{\rho_2^*\rho_4}{\mu_2\mu_2\mu_4} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & & \ddots & \frac{1}{\mu_{n-1}} & 0 \\ \frac{\rho_1^*\rho_n}{\mu_1\cdot\ldots\cdot\mu_n} & \frac{\rho_2^*\rho_n}{\mu_2\cdot\ldots\cdot\mu_n} & \cdots & \cdots & \frac{\rho_{n-1}^*\rho_n}{\mu_{n-1}\mu_n} & \frac{1}{\mu_n} \end{bmatrix} \tag{4.11}$$

A similar examination of the state space structure for the dual filter in Fig. 9 leads to the analogous description for the Horner-Szegő polynomials,

$$\begin{bmatrix} \tilde{\phi}_0^{\#}(z^{-1}) \\ \tilde{\phi}_1^{\#}(z^{-1}) \\ \tilde{\phi}_2^{\#}(z^{-1}) \\ \vdots \\ \tilde{\phi}_n^{\#}(z^{-1}) \end{bmatrix} = z(z\tilde{I}A^{-T}\tilde{I} - Z)^{-1} \begin{bmatrix} b_n \\ b_{n-1} \\ \vdots \\ b_1 \\ b_0 \end{bmatrix}.$$

- Below we shall need an explicit expression for the last coefficients of the Horner-Szegő polynomials, the vector of these coefficients is given by

$$\begin{bmatrix} a_{00} \\ a_{11} \\ a_{22} \\ \vdots \\ a_{nn} \end{bmatrix} = A \begin{bmatrix} b_n \\ b_{n-1} \\ \vdots \\ b_0 \end{bmatrix}. \tag{4.12}$$

This identity can be deduced from the signal flow graphs in Fig. 9 or 11 by forming a delay-free path from the input of the line to the input of the k-th delay element, or algebraically from (4.9). Of course, by setting $b_k = 0$ for $k = 0, 1, \ldots, n-1$ and b_n we obtain the well-known fact the vector of the last coefficients of Szegő polynomials (however, corresponding to the reversed vector of reflection coefficients) is given by the first column of A in (4.11).

5 Horner-Szegő polynomials and direct and inverse factorization of quasi-Toeplitz matrices

5.1 General inner products and factorization of moment matrices

For any Hermitian positive definite $n \times n$ matrix M we can define an inner product in the space of all polynomials of degree less that n by

$$< \sum_{k=0}^{n-1} p_k x^k, \sum_{k=0}^{n-1} q_k x^k > := \begin{bmatrix} q_0 & q_1 & \cdots & q_{n-1} \end{bmatrix} M \begin{bmatrix} p_0^* \\ p_1^* \\ \vdots \\ p_{n-1}^* \end{bmatrix} \tag{5.1}$$

The matrix $M = \left[\ < x^i, x^j > \ \right]_{0 \le i,j \le n-1}$ is called a *moment matrix.* For a triangular factorization,

$$\tilde{I}M^{-1}\tilde{I} = \begin{bmatrix} a_{n0}^* & 0 & \cdots & 0 \\ \vdots & a_{n-1,0}^* & \ddots & \vdots \\ a_{n,n-1}^* & & \ddots & 0 \\ a_{n,n}^* & a_{n-1,n-1}^* & \cdots & a_{0,0}^* \end{bmatrix} \begin{bmatrix} a_{n,0} & a_{n,1} & \cdots & a_{n,n} \\ 0 & a_{n-1,1} & \cdots & a_{n-1,n-1} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_{00} \end{bmatrix}, \tag{5.2}$$

it is a trivial exercise to see that the polynomials

$$a_k(x) = a_{k,k} + a_{k,k-1}x + \ldots + a_{k,1}x^{k-2} + a_{k,0}x^{k-1} \tag{5.3}$$

are orthonormal with respect to the moment matrix

$$M = \begin{bmatrix} a_{0,0} & 0 & \cdots & 0 \\ a_{1,1} & a_{1,0} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_{n,n} & a_{n,n-1} & \cdots & a_{n,0} \end{bmatrix}^{-1} \cdot \begin{bmatrix} a_{0,0}^* & a_{1,1}^* & \cdots & a_{n,n}^* \\ 0 & a_{1,0}^* & \cdots & a_{n,n-1}^* \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_{n,0}^* \end{bmatrix}^{-1}. \tag{5.4}$$

These triangular factorizations indicate that in general $\frac{(n+1)n}{2}$ parameters are needed to represent M. As we shall see next, for special polynomials much less parameters can be needed.

5.2 Szegő polynomials, Toeplitz moment matrices and the classical Levinson algorithm

Recall that in the special case of Szegő polynomials the inner product $< \cdot, \cdot >_{\Phi\#}$ is defined by (1.1), so that the corresponding moment matrix T has a *Toeplitz structure* shown in the next formula

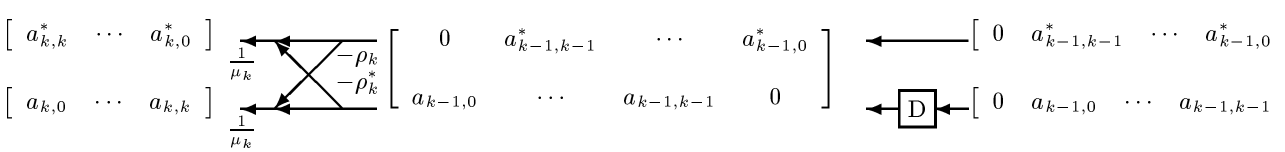
$$T_n = \begin{bmatrix} t_0 & t_1^* & t_2^* & \cdots & t_{n-1}^* \\ t_1 & t_0 & t_1^* & \ddots & \vdots \\ t_2 & t_1 & t_0 & \ddots & t_2^* \\ \vdots & \ddots & \ddots & \ddots & t_1^* \\ t_{n-1} & \cdots & t_2 & t_1 & t_0 \end{bmatrix}, \tag{5.5}$$

i.e., it has constant values along all its diagonals. Summarizing, the moment matrix for Szegő polynomials can be parameterized by only $(n+1)$ parameters $\{t_0, \ldots, t_n\}$ (or by another set of $(n+1)$ parameters $\{\mu_0, \rho_1, \ldots, \rho_n\}$). The coefficients of Szegő polynomials,

$$\phi_k^{\#}(x) = a_{k,0}x^k + a_{k,1}x^{k-1} + \ldots + a_{k-1,k-1},$$

can be computed via the well-known Levinson algorithm which starts with the entries of the Toeplitz matrix T and computes the coefficients $\{a_{k,j}\}$ of $\{\phi_k^{\#}(x)\}$ for $k = 0, 1, \ldots, n$. It is not relevant at the moment how the Levinson algorithm finds the reflection coefficients, more important that its recursion coincides with the recursion (1.3), which is translated to the array manipulations in Fig. 13, where the delay element, $\boxed{\text{D}}$, is understood as a left shift.

Figure 13. Array form of the classical Levinson algorithm.



5.3 Array form of the recursion for the Horner-Szegő polynomials

Analogously for the Horner-Szegő polynomials, denoting

$$\tilde{\phi}_k(x) = v_{k,0}x^k + \ldots + v_{k,k-1}x + v_{k,k},$$

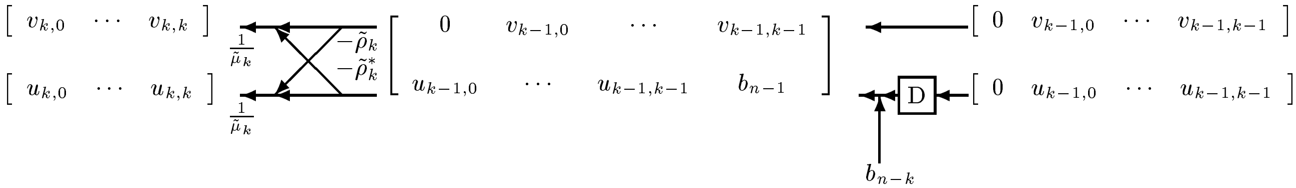
$$\tilde{\phi}_k^{\#}(x) = u_{k,0}x^k + \ldots + u_{k,k-1}x + u_{k,k},$$

we can rewrite (4.6) in the array form,

$$\begin{bmatrix} v_{k,0} & \cdots & v_{k,k} \\ u_{k,0} & \cdots & u_{k,k} \end{bmatrix} = \frac{1}{\tilde{\mu}_k} \begin{bmatrix} 1 & -\tilde{\rho}_k^* \\ -\tilde{\rho}_k & 1 \end{bmatrix} \begin{bmatrix} 0 & v_{k-1,0} & \cdots & v_{k-1,k-1} \\ u_{k-1,0} & \cdots & u_{k-1,k-1} & b_{n-k} \end{bmatrix} \tag{5.6}$$

which is drawn in Fig. 14.

Figure 14. Array form of the recursion for the Horner-Szegő polynomials.



In brief, the difference between Szegő polynomials in Fig. 13 and the Horner-Szegő polynomials in Fig. 14 is in the feed-in branches, providing the last components b_{n-k}. As we shall see below, this means that while the reflection coefficients completely define the (Toeplitz) moment matrix T for the classical Szegő polynomials, the moment matrix for the Horner-Szegő polynomials will be parameterized by $2(n+1)$ parameters, $\{\mu_0, \rho_1, \ldots, \rho_n, b_0, \ldots, b_n\}$, as expected. This, of course, will imply that it will no longer be Toeplitz, but we next show that it will nevertheless have a similar shift-invariance property.

5.4 Matrices with shift-invariant structure

An $n \times n$ matrix R is said to be *Toeplitz-like* [FMKL79], [KKM79], if its *displacement rank*,

$$\alpha = \operatorname{rank}(R - ZRZ^*) \tag{5.7}$$

is small compared to the size n of R. Here Z is the lower shift matrix, so that ZRZ^* is obtained from R by just shifting all the entries along the diagonals. It can be easily checked that any Toeplitz matrix has displacement rank not exceeding 2:

$$T - ZTZ^* = \frac{1}{c_0} \begin{bmatrix} 0 & c_{-1} & \cdots & c_{-n+1} \\ c_0 & c_{-1} & \cdots & c_{-n+1} \end{bmatrix}^* \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & c_{-1} & \cdots & c_{-n+1} \\ c_0 & c_{-1} & \cdots & c_{-n+1} \end{bmatrix}.$$

so the classical Toeplitz matrices are Toeplitz-like. There are many examples of matrices with displacement rank higher than 2, for example, for the product of two Toeplitz matrices it does not exceed 4.

For our purposes in this paper we restrict ourselves with *quasi-Toeplitz* matrices, which have displacement rank 2, and moreover, the *displacement inertia* (1,1, n-2), i.e. $R - ZRZ^*$ has only two nonzero eigenvalues: one positive and one negative.

We next show that moment matrices for the Horner-Szegő polynomials are quasi-Toeplitz. To do so we need to recall a well-known connection of quasi-Toeplitz matrices to the celebrated Schur algorithm, see, e.g., [K86].

5.5 The classical Schur algorithm for fast triangular factorization of quasi-Toeplitz matrices

It was now well-known that the celebrated classical Schur algorithm [S17] in its array form computes the Cholesky factorization for quasi-Toeplitz matrices. The procedure can be briefly described as follows. Given a positive definite quasi-Toeplitz matrix R defined by

$$R - ZRZ^* = \begin{bmatrix} \mathbf{v}_{n+1}^* & \mathbf{u}_{n+1}^* \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{v}_{n+1} \\ \mathbf{u}_{n+1} \end{bmatrix} \tag{5.8}$$

where

$$\begin{bmatrix} \mathbf{v}_k \\ \mathbf{u}_k \end{bmatrix} = \begin{bmatrix} v_{k,0} & v_{k,1} & \cdots & v_{k,k} \\ u_{k,0} & u_{k,1} & \cdots & u_{k,k} \end{bmatrix} \tag{5.9}$$

Define the recursion

$$\begin{bmatrix} 0 & \mathbf{v}_{k-1} \\ & \bar{\mathbf{u}}_{k-1} \end{bmatrix} = \begin{bmatrix} 0 & v_{k-1,0} & \cdots & v_{k-1,k-2} & v_{k-1,k-1} \\ u_{k-1,0} & u_{k-1,1} & \cdots & u_{k-1,k-1} & u_{k-1,k} \end{bmatrix} := \frac{1}{\mu_k} \begin{bmatrix} 1 & \rho_k^* \\ \rho_k & 1 \end{bmatrix} \begin{bmatrix} \mathbf{v}_k \\ \mathbf{u}_k \end{bmatrix}, \tag{5.10}$$

where we choose the reflection coefficient $\rho_k^* = -\frac{v_{k,0}}{u_{k,0}}$ to introduce zero in the first row of the matrix on the left hand side of (5.10). We obtain the next pair of row vectors $\begin{bmatrix} \mathbf{v}_{k-1} \\ \mathbf{u}_{k-1} \end{bmatrix}$ by shifting the second row one position right:

$$\begin{bmatrix} 0 & \mathbf{v}_{k-1} \\ 0 & \mathbf{u}_{k-1} \end{bmatrix} := \begin{bmatrix} 0 & v_{k-1,0} & v_{k-1,1} & \cdots & v_{k-1,k-1} \\ 0 & u_{k-1,0} & u_{k-1,1} & \cdots & u_{k-1,k-1} \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{v}_{k-1} \\ 0 & \bar{\mathbf{u}}_{k-1} Z^T \end{bmatrix}. \tag{5.11}$$

The point is that this procedure computes the Cholesky factorization,

$$R = U^* U, \qquad U = \begin{bmatrix} u_{n-1,0} & u_{n-1,1} & \cdots & u_{n-1,n} \\ 0 & u_{n-2,0} & \cdots & u_{n-2,n-1} \\ \cdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & u_{-1,0} \end{bmatrix} \tag{5.12}$$

for the quasi-Toeplitz matrix R in (5.8).

5.6 Parameterizations of quasi-Toeplitz matrices

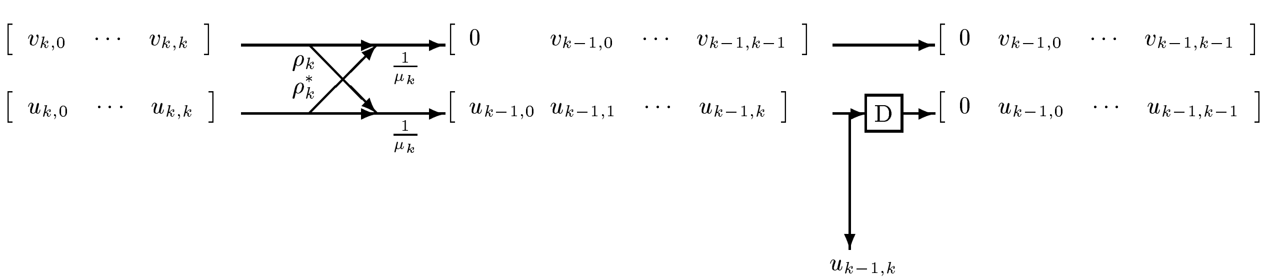
Of course, a representation (5.8) parameterizes the class of quasi-Toeplitz matrices by $2(n+1)$ entries of the vectors $\{\mathbf{u}_n, \mathbf{v}_n\}$. There are several other well-known ways to parameterize quasi-Toeplitz matrices. For example, any quasi-Toeplitz matrix R is congruent to a certain Toeplitz matrix T,

$$R = L(x) T L(x)^*,$$

where the congruence matrix $L(x)$ is a lower triangular Toeplitz matrix, see, e.g., [LA83], [LAK86].

The association of the classical Schur algorithm with discrete transmission lines offers a variety of other possibilities to parameterize the class of quasi-Toeplitz matrices, cf., e.g., [D82] and [LA83]. For example, one can draw a single recursion step (5.10) (5.11) as shown in Fig. 13.

Figure 15. Lattice filter interpretation of the classical Schur algorithm.



If we would like to reconstruct the $2(k+1)$ input entries from $2k$ output entries, we need to know two more parameters $\{\rho_k, u_{k-1,k}\}$ which we associate with this single recursion step. Therefore we can recover two vectors $\{\mathbf{u}_{n+1}, \mathbf{v}_{n+1}\}$ defining the quasi-Toeplitz matrix via (5.8) by running the classical Schur algorithm backward, and the only parameters we need to know are the reflection coefficients $\{\rho_k\}$ and the feed-in parameters $\{u_{k-1,k}\}$.

5.7 Quasi-Toeplitz structure of the moment matrix

After these observations, a closer at the signal flow graphs in Fig. 14 and 15 reveals that the recursion (4.6) for the Horner-Szegő polynomials is in fact a backward Schur algorithm for the triangular factorization of a quasi-Toeplitz matrix given by

$$(\tilde{I} M^{-1} \tilde{I}) - Z(\tilde{I} M^{-1} \tilde{I}) Z^* =$$

$$\begin{bmatrix} 0 & q_{n-1,0} & \cdots & q_{n-1,n-1} \\ p_{n-1,0} & \cdots & p_{n-1,n-1} & b_0 \end{bmatrix}^* \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & q_{n-1,0} & \cdots & q_{n-1,n-1} \\ p_{n-1,0} & \cdots & p_{n-1,n-1} & b_0 \end{bmatrix}$$

Since the (Toeplitz-like, as in (5.7)) displacement structure of a matrix is inherited under passing $M \to \tilde{I}M^{-*}\tilde{I}$, the matrix M is quasi-Toeplitz itself, and it is completely parameterized by the parameters of the transmission line: the reflection coefficients $\{\mu_0, \rho_1, \ldots, \rho_n\}$ and feed-in parameters $\{b_k\}$ which are clearly the entries of the last column of the Cholesky factor,

$$U = \begin{bmatrix} * & * & \cdots & \cdots & * & b_n \\ 0 & * & \cdots & \cdots & * & b_{n-1} \\ \vdots & \ddots & \ddots & & \vdots & \vdots \\ \vdots & & \ddots & \ddots & * & \vdots \\ \vdots & & & \ddots & * & b_1 \\ 0 & \cdots & \cdot & \cdots & 0 & b_0 \end{bmatrix}$$

in $(\tilde{I}M^{-1}\tilde{I}) = U^*U$, see, e.g., (5.12).

5.8 Some concluding remarks

In this section we restricted ourselves with the simplest case of quasi-Toeplitz matrices, but the approach allowed us to study more general classes of Toeplitz-like matrices and matrices with more general displacement structures, which will be addressed in the forthcoming publication.

Finally note that the displacement structure approach allows us to study the more general classes of Szegő-Vandermonde-like matrices, and polynomial Vandermonde-like matrices, see, e.g., [KO95], [KO97a] and also [GO94b], [GKO95].

References

[A65] N.I.Akhiezer, *The classical moment problem and some related problems in analysis*, Hafner Publishing Co., New York, 1965.

[ACR96] G.S.Ammar, D.Calvetti and L.Reichel, *Continuation methods for the computation of zeros of Szegő polynomials*, Linear Algebra and Its Applications, **249**: 125-155 (1996).

[AGR86] G.S.Ammar, W.B.Gragg and L.Reichel, *On the eigenproblem for orthogonal matrices*, Proc. of the 25th IEEE Conference on Decision and Control, Athens, Greece, 1963-1966 (1986).

[AGR87] G.S.Ammar, W.B.Gragg and L.Reichel, *Determination of Pisarenko frequency estimates as eigenvalues of orthonormal matrices*, SPIE vol. **826**, Proc. of the 1987 SPIE conference on Advanced Algorithms and Architectures for Signa Processing II, 143-145 (1987).

[AGR91] G.S.Ammar, W.B.Gragg and L.Reichel, *Constructing a unitary Hessenberg matrix from spectral data*, in Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms (G.Golub and P.Van Doores, Eds.), 385-595, Springer Verlag, Berlin, 1991.

[AGR92] G.S.Ammar, W.B.Gragg and L.Reichel, *Downdating of Szegő polynomials and data-fitting applications*, Linear Algebra and Its Appl., **172**: 315-336 (1992).

[AGR93] G.S.Ammar, W.B.Gragg and L.Reichel, *An analogue for the Szegő polynomials of the Clenshaw algorithm*, J. Computational Appl. Math., **46**: 211-216 (1993).

[B75] S.Barnett, *A companion matrix analogue for orthogonal polynomials*, Linear Algebra Appl., **12**: 197-208 (1975).

[BC92] M.Bakonyi and T.Constantinescu, *Schur's algorithm and several applications*, in Pitman Research Notes in Mathematics Series, vol. 61, Longman Scientific and Technical, Harlow, 1992.

[BGE91] A.Bunse-Gerstner and L.Elsner, *Schur parameter pensils for the solution of unitary eigenproblem*, Linear Algebra and Its Appl., **154-156**: 741-778 (1991).

[BK86] A.Bruckstein and T.Kailath, *Some Matrix Factorization Identities for Discrete Inverse Scattering*, Linear Algebra Appl., **74**: 157-172 (1986).

[BK87a] A.Bruckstein and T.Kailath, *Inverse Scattering for Discrete Transmission Line Models*, SIAM Review, **29**: 359-389 (1987).

[BK87b] A.Bruckstein and T.Kailath, *An Inverse Scattering Framework for Several Problems in Signal Processing*, IEEE ASSP Magazine, January 1987, 6-20.

[C55] C.Clenshaw, *A note on summation of Chebyshev series*, M.T.A.C., **9**(51): 118-120 (1955).

[C84] T.Constantinescu, *On the structure of the Naimark dilation*, J. of Operator Theory, **12**: 159-175 (1984).

[CR93] D.Calvetti and L.Reichel, *Fast inversion of Vandermonde-like matrices involving orthogonal polynomials*, BIT, **33**: 473-484 (1993).

[D82] J.M.Delosme, *Algorithms for finite shift-rank processes*, Ph.D.Thesis, Stanford University, 1982.

[F96] P.Fuhrmann, *A polynomial approach to linear algebra*, Springer Verlag, New York, 1996.

[FF89] C.Foias and A.E.Frazho, *The Commutant Lifting Approach to Interpolation Problems*, Birkhauser Verlag, 1989.

[FMKL79] B.Friedlander, M.Morf, T.Kailath and L.Ljung, *New inversion formulas for matrices classified in terms of their distance from Toeplitz matrices*, Linear ALgebra and Its Applications, **27**: 31-60 (1979).

[G48] L.Y.Geronimus, *Polynomials orthogonal on a circle and their applications*, Amer. Math. Translations, **3**:1-78 (1954) (Russian original 1948).

[G82] W.B.Gragg, *Positive definite Toeplitz matrices, the Arnoldi process for isometric operators, and Gaussian quadrature on the unit circle* (in Russian). In : E.S. Nikolaev (Ed.), *Numerical methods in Linear Algebra*, Moscow University Press, 16-32 (1982).
English translation in : J. Comput. and Appl. Math., **46**: 183-198 (1993).

[G86] W.B.Gragg, *The QR algorithm for unitary Hessenberg matrices*, J.Comput. Appl. Math., **16**: 1-8 (1986).

[G97] W.B.Gragg, *A stabilized UHQR algorithm*, preprint, 1997.

[GF74] I.Gohberg and I.Feldman, *Convolution equations and projection methods for their solutions*, Translations of Mathematical Monographs, **41**, Amer. Math. Soc., 1974.

[GL83] W.B.Gragg and A.Lindquist, *On the partial realization problem*, Linear Algebra and Its Appl., **50**: 277-319 (1983).

[GKO95] I.Gohberg, T.Kailath and V.Olshevsky, *Fast Gaussian elimination with partial pivoting for matrices with displacement structure*, Math. of Computation, **64**: 1557-1576 (1985).

[GO94a] I.Gohberg and V.Olshevsky, *Fast inversion of Chebyshev-Vandermonde matrices*, Numerische Mathematik, **67**, No. 1: 71-92 (1994).

[GO94b] I.Gohberg and V.Olshevsky, *Fast state space algorithms for matrix Nehari and Nehari-Takagi interpolation problems*, Integral Equations and Operator Theory, **20**, No. 1: 44-83 (1994).

[GO97] I.Gohberg and V.Olshevsky, *The fast generalized Parker-Traub algorithm for inversion of Vandermonde and related matrices*, J.of Complexity, **13(2)**: 208-234 (1997).

[GR87] W.B.Gragg and L.Reichel, *A divide and conquer algorithm for unitary eigenproblem*, in Hypercube Multiprocessors (M.T.Heath, ed.), SIAM Publications, Philadelphia, 639-647 (1987).

[GS58] U.Grenader and G.Szegő, *Toeplitz forms and Applications*, University of California Press, 1958.

[GS72] I.Gohberg and A.Semencul, *On the inversion of finite Toeplitz matrices and their continuous analogs (in Russian)*, Mat. Issled., **7(2)**: 201-233 (1972).

[HHR89] G.Heinig, W.Hoppe and K.Rost, *Structured matrices in interpolation and approximation problems*, Wissenschaftl. Zeitschrift der TU Karl-Marx-Stadt 31, **2**: 196-202 (1989).

[K52] M.G.Krein, *On a generalization of investigations of Stiltjes*, Doklady Akad. Nauk USSR, **87**: 881-884 (1952).

[K85] H.Kimura, *Generalized Schwartz Form and Lattice-Ladder Realizations for Digital Filters*, IEEE Transactions on Circuits and Systems, **32**, No 11: 1130-1139 (1985).

[K86] T.Kailath, *A theorem of I.Schur and its impact on modern signal processing*, in Operator Theory: Advances and Applications (I.Schur methods in Operator Theory and Signal Processing), **18**, 9-30, Birkhauser, 1986.

[K79] R.E.Kalman, *On partial realizations, transfer functions, and canonical forms*, Acta Polytech. Scand., MA31 (1979), 9-32.

[K80] T.Kailath, *Linear Systems*, Prentice Hall, Englewood Cliffs, New Jersey, 1980.

[KBM86] T.Kaiath, A.Bruckstein and D.Morgan, *Fast Matrix Factoriation via Discrete Transmission Lines*, Linear Algebra Appl., **75**: 1-25 (1986).

[KKM79] T.Kailath, S.Kung and M.Morf, *Displacement ranks of matrices and linear equations*, J. Math. Anal. and Appl., **68**: 395-407 (1979).

[KO95] T.Kailath and V.Olshevsky, *Displacement structure approach to Chebyshev-Vandermonde and related matrices*, Integral Equations and Operator Theory, **22**: 65-92 (1995).

[KO97a] T.Kailath and V.Olshevsky, *Displacement structure approach to polynomial Vandermonde and related matrices*, Linear Algebra and Its Appl., **261**: 49-90 (1997).

[KO97b] T.Kailath and V.Olshevsky, *Symmetric and Bunch-Kaufman pivoting for partially structured Cauchy-like matrices with applications to Toeplitz-like linear systems, and to boundary rational matrix interpolation problems*, Linear Algebra and Its Appl., **254**: 251-302 (1997).

[KP83] T.Kailath and B.Porat, *State-space generators for orthogonal polynomials*, in "Prediction Theory and Harmonic Analysis", the Pesi Masani volume (V.Mandrekar and J.Salehi, Eds.), North Holland Publishing Company, Amsterdam, 131-163 (1983).

[L80] D.T.Lee, *Canonical Ladder form realizations and fast estimation algorithms*, Ph.D. thesis, Department of Electrical Engineering, Stanford University, August 1980.

[LA83] H.Lev-Ari, *Nonstationary Lattice-Filter Modeling*, Ph.D. thesis, Stanford University, December 1983.

[LAK84] H.Lev-Ari and T.Kailath, *Lattice filter parameterization and modeling of nonstationary processes*, IEEE Trans on Information Theory, **30**: 2-16 (1984).

[LAK86] H.Lev-Ari and T.Kailath, *Triangular factorization of structured Hermitian matrices*, in *Operator Theory : Advances and Applications* (I.Gohberg. ed.), vol. **18**, 301-324, Birkhäuser, Boston, 1986.

[M66] M.Mansour, *Stability criteria of linear systems and the second method of Lyapunov*, Sientia Electrica, vol. XI, 87-96 (1966).

[M74] M.Morf, *Fast algorithms for multivariable systems*, Ph.D. thesis, Department of Electrical Engineering, Stanford University, 1974.

[M77] M.Morf, *Ladder forms in estimation and control*, Proc. 11 Asilomar Annual Conference on CSC, November 1997, 424-429.

[MB79] J.Maroulas and S.Barnett, *Polynomials with respect to a general basis. I. Theory*, J. of Math. Analysis and Appl., **72**: 177-194 (1979).

[MG76] J.D.Markel and A.H.Gray, *Linear Prediction of Speech*, Communications and Cybernetics, Springer-Verlag, Berlin, 1976.

[ML80] M.Morf and D.T.Lee, *State-space structure of ladder canonical forms*, Proc. 18th Conf. on Control and Design, Dec. 1980, 1221-1224.

[MZ60] S.H.Mason and H.A.Zimmerman, *Electronic Circuits, Signals and Systems*, J.Wiley, New York, 1960.

[OS89] A.V.Oppenheim and R.W.Schafer, *Discrete-time Signal Processing*, Prentice Hall, Eglewood Cliffs, NJ, 1989.

[PM72] T.W.Park and J.H.McClellan, *Chebyshev approximation for the design of nonrecursive digital filters with linear phase*, IEEE Trans. Circuit Theory, vol. **CT-19**: 189-194 (1972).

[R57] E.Ya.Remez, *General Computational Methods of Chebyshev Approximations*, Atomic Energy Translation **4491**, Kiev, USSR, 1957.

[R95] P.A.Regalia, *Adaptive IIR filtering in signal processing and control*, Marcel Dekker, New York, 1995.

[S17] I.Schur, *Über Potenzreihen die im Inneren des Einheitskreises beschränkt sind*, Journal für die Reine und Angewandte Mathematik, **147**: 205-232 (1917). English translation in *Operator Theory : Advances and Applications* (I.Gohberg. ed.), vol. **18**: 31-88 (1986), Birkhäuser, Boston, 1986.

[TKH83] M.Takizawa, Hisao Kishi and N.Hamada, *Synthesis of Lattice Digital Filter by the State Space Variable Method*, Trans. IECE Japan, vol. J65-A: 363-370 (1983).