Signal Flow Graph Approach to Inversion of (H, m)–quasiseparable Vandermonde Matrices and New Filter Structures

T.Bella*, V.Olshevsky† and P.Zhlobich†

Abstract. We use the language of signal flow graph representation of digital filter structures to solve three purely mathematical problems, including fast inversion of certain polynomial–Vandermonde matrices, deriving an analogue of the Horner and Clenshaw rules for polynomial evaluation in a (H, m)–quasiseparable basis, and computation of eigenvectors of (H, m)–quasiseparable classes of matrices. While algebraic derivations are possible, using elementary operations (specifically, flow reversal) on signal flow graphs provides a unified derivation, and reveals connections with systems theory, etc.

1. Introduction

1.1. Signal flow graphs for proving matrix theorems

Although application–oriented, *signal flow graphs* representing *discrete transmission lines* have been employed to answer purely mathematical questions, such as providing interpretations of the classical algorithms of Schur and Levinson, deriving fast algorithms, etc., see for instance [6, 8, 7, 14, 15]. In particular, questions involving structured matrices that are associated with systems of polynomials satisfying recurrence relations lend themselves well to a signal flow graph approach. For instance, it is well–known that matrices with Toeplitz structure are related to Szegö polynomials (polynomials orthogonal on the unit circle). This relation was exploited in [6] as shown in the next example.

[0]* Department of Mathematics, University of Rhode Island, Kingston RI 02881-0816, USA. Email: tombella@math.uri.edu
† Department of Mathematics, University of Connecticut, Storrs CT 06269-3009, USA. Emails: olshevsky@uconn.edu, zhlobich@math.uconn.edu

Example 1.1 (Proof of the Gohberg–Semencul formula). In [6], the signal flow graph language is used to give a proof of the well–known Gohberg–Semencul formula. In fact, the proof is simply a single signal flow graph, shown here in Figure 1.

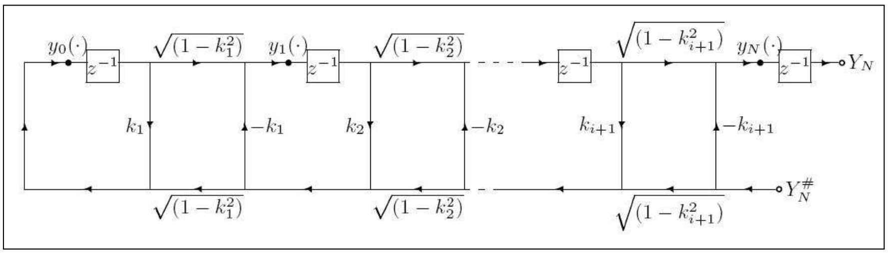


FIGURE 1. Proof of the Gohberg–Semencul formula

The "proof" shown in Figure 1 as presented in [6] may seem quite mysterious at this point, however the intent in presenting it at the beginning is to make the point that the language of signal flow graphs provides a language for proving mathematical results for structured matrices using the recurrence relations of the corresponding polynomial systems.

The results of this paper are presented via signal flow graphs, however we do not assume **any** familiarity with signal flow graphs, and the reader can consider them as a convenient way of visualizing recurrence relations.

1.2. Quasiseparable and semiseparable polynomials

In this paper, the language of signal flow graphs is used to address three closely related problems, posed below in Sections 1.3 through 1.5. While the use of signal flow graphs is applicable to general systems, their use is most effective when the system of polynomials in question satisfy sparse recurrence relations. Herein, we focus on the class of (H, m)–quasiseparable polynomials, systems of polynomials related as characteristic polynomials of principal submatrices of Hessenberg, order m quasiseparable matrices, and their subclass of (H, m)–semiseparable polynomials. Formal definitions of these classes and details of the relations between polynomial systems and structured matrices are given in Section 3.

A motivation for considering (H, m)–quasiseparable polynomials in this context is as follows. It will be demonstrated in detail below that real–orthogonal polynomials and Szegö polynomials (that is, polynomials orthogonal not on a real interval, but on the unit circle) are special cases of $(H, 1)$–quasiseparable polynomials, as are monomials. This relationship is illustrated in Figure 2. Thus all of the results given here generalize those known for these important classes, and additionally provide a unifying derivation of these previous results.

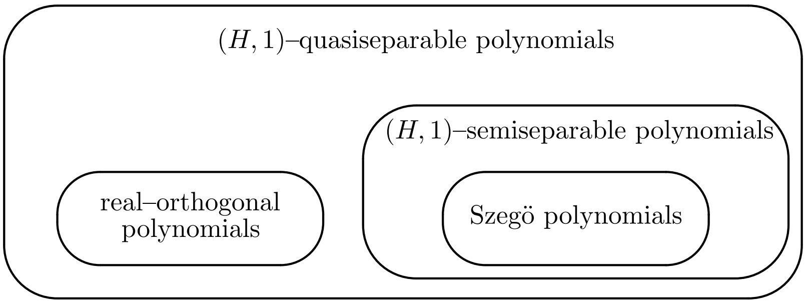


FIGURE 2. Relations between polynomial systems studied in this paper.

1.3. Polynomial evaluation rules extending Horner and Clenshaw type rules for (H, m)–quasiseparable polynomials

The first problem we consider is that of efficient polynomial evaluation. As a motivation, consider a polynomial given in terms of the monomial basis,

$$H(x) = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} + a_n x^n.$$

It is well–known that this can be rewritten as

$$\begin{aligned} H(x) &= a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} + a_n x^n \\ &= \underbrace{a_0 + x(\underbrace{a_1 + x(\cdots x(\underbrace{a_{n-1} + x(\underbrace{a_n}_{\tilde{p}_0(x)}}_{\tilde{p}_1(x)}))}_{\tilde{p}_{n-1}(x)})}_{\tilde{p}_n(x)=H(x)} \end{aligned}$$

which amounts to expressing the polynomial not in terms of the monomials, but in terms of the Horner polynomials; i.e., those satisfying the recurrence relations

$$\tilde{p}_0(x) = a_n, \quad \tilde{p}_k(x) = x\tilde{p}_{k-1}(x) + a_{n-k}. \tag{1.1}$$

Since, as illustrated, $\tilde{p}_n(x) = H(x)$, the polynomial $H(x)$ may be evaluated at a point x by computing successive Horner polynomials, avoiding direct computation of large powers of x, etc.

We consider the problem of similar evaluation of a polynomial given in terms of an arbitrary system of polynomials; that is, of the form

$$H(x) = b_0 r_0(x) + b_1 r_1(x) + \cdots + b_{n-1} r_{n-1}(x) + b_n r_n(x)$$

for some polynomial system $\{r_k\}$. Of particular interest will be the case where the polynomial system in question is a system of (H, m)–quasiseparable polynomials, and in which case the evaluation algorithm will be efficient.

In the case of real–orthogonal polynomials, such an evaluation rule is known, and is due to Clenshaw [10]. In addition, an efficient evaluation algorithm for polynomials given in terms of Szegö polynomials was presented by Ammar, Gragg, and Reichel in [1].[1] These previous results as well as those derived in this paper are listed in Table 1.

TABLE 1. Polynomial evaluation algorithms.

Polynomial System R	efficient evaluation algorithm
monomials	Horner Rule [?]
Real orthogonal polynomials	Clenshaw Rule [10]
Szegö polynomials	Ammar–Gragg–Reichel Rule [1]
(H,m)-quasiseparable	this paper

Using the language of signal flow graphs, the polynomial evaluation rule that we derive is very general, and it generalizes and explains the previous results of Table 1.

1.4. (H,m)–quasiseparable eigenvector problem

The second problem considered in this paper is that of computing eigenvectors of (H,m)–quasiseparable matrices and (H,m)–semiseparable matrices, given their eigenvalues. Applications of this problem can be seen to be numerous knowing that companion matrices, irreducible tridiagonal matrices, and almost unitary Hessenberg matrices are all special cases of (H,m)–quasiseparable matrices and some of their subclasses.

For instance, it is well–known that the columns of the inverse of the Vandermonde matrix

$$V(x) = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ 1 & x_3 & x_3^2 & & \vdots \\ \vdots & \vdots & & \ddots & x_{n-1}^{n-1} \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{bmatrix}$$

store the eigenvectors of the companion matrix

$$C = \begin{bmatrix} 0 & 0 & \cdots & 0 & -c_0 \\ 1 & 0 & \cdots & 0 & -c_1 \\ 0 & 1 & \ddots & \vdots & -c_2 \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & 1 & -c_{n-1} \end{bmatrix},$$

[1]Although the algorithm of [1] does indeed evaluate a polynomial given in a Szegö basis, it is not exactly an analogue of the Horner and Clenshaw rules in some sense. The signal flow graph interpretation of this paper can be used to explain the difference, see Section 5.4.1.

as can be seen by the easily verified identity

$$V(x)C = D(x)V(x), \quad D(x) = diag(x_1, x_2, \dots, x_n).$$

Using the signal flow graph approach described in this paper, it is described how to use signal flow graph operations to compute the eigenvectors of a given (H, m)–quasiseparable matrix using its eigenvalues. These results include as special cases the descriptions of eigenvectors of companion matrices, tridiagonal matrices, unitary Hessenberg matrices, arrowhead matrices, and Hessenberg banded matrices, among many others.

1.5. Inversion of (H, m)–quasiseparable–Vandermonde matrices

Finally, the third problem considered in this paper is that of efficiently inverting the polynomial–Vandermonde matrix

$$V_R(x) = \begin{bmatrix} r_0(x_1) & r_1(x_1) & \cdots & r_{n-1}(x_1) \\ r_0(x_2) & r_1(x_2) & \cdots & r_{n-1}(x_2) \\ \vdots & \vdots & & \vdots \\ r_0(x_n) & r_1(x_n) & \cdots & r_{n-1}(x_n) \end{bmatrix}, \tag{1.2}$$

where the polynomial system $\{r_k(x)\}$ is a system of (H, m)–quasiseparable polynomials. We refer to such matrices as (H, m)–quasiseparable–Vandermonde matrices. Special cases of (H, m)–quasiseparable–Vandermonde matrices include classical Vandermonde matrices involving the monomial basis (as the monomials are $(H, 0)$–quasiseparable polynomials), three–term Vandermonde matrices involving real orthogonal polynomials (as real orthogonal polynomials are $(H, 1)$–quasiseparable polynomials), and Szegö–Vandermonde matrices involving Szegö polynomials (as Szegö polynomials are $(H, 1)$–quasiseparable polynomials).

The well–known fast $\mathcal{O}(n^2)$ inversion algorithm for classical Vandermonde matrices $V_P(x) = [x_i^{j-1}]$ was initially proposed by Traub in [20] (see for instance, [13] for many relevant references and some generalizations), and has since been extended to many important cases beyond the classical Vandermonde case. In Table 2, several references to previous algorithms in this area are given.

Using the language of signal flow graphs, we rederive the results of the latest and most general work of Table 2, [4]. Thus, this use of signal flow graphs results in an algorithm generalizing the previous work.

1.6. Overview of the paper

The three problems described above are connected and solved via the use of operations on signal flow graphs, specifically, flow reversal of a signal flow graph. That is, all three problems are solved by forming an appropriate signal flow graph, reversing the flow, and reading off the solution in a particular way. In the course of the paper, new filter structures corresponding to both (H, m)–quasiseparable matrices and their subclass, (H, m)–semiseparable matrices, are given and classified in terms of recurrence relations as well.

TABLE 2. Fast $\mathcal{O}(n^2)$ inversion algorithms.

Matrix $V_R(x)$	Polynomial System R	Fast inversion algorithm
Classical Vandermonde	monomials	Traub [20]
Chebyshev–V.	Chebyshev	Gohberg-Olshevsky [12]
Three–Term V.	Real orthogonal	Calvetti-Reichel [9]
Szegö–Vandermonde	Szegö	Olshevsky [19]
(H,m)-semiseparable–Vandermonde	(H,m)-semiseparable	BEGOTZ [4] (new derivation in this paper)
$(H,1)$-quasiseparable–Vandermonde	$(H,1)$-quasiseparable	BEGOT [3] (new derivation in this paper)
(H,m)-quasiseparable–Vandermonde	(H,m)-quasiseparable	BEGOTZ [4] (new derivation in this paper)

2. Signal flow graph overview & motivating example

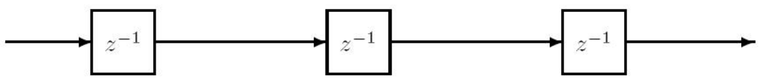
Common in electrical engineering, control theory, etc., signal flow graphs represent realizations of systems as electronic devices. Briefly, the objective is to build a device to implement, or realize, a polynomial, using devices that implement the algebraic operations used in recurrence relations. These building blocks are shown next in Table 3. (Note that in this paper, we often follow the standard notation in signal flow graphs of expressing polynomials in terms of $x = z^{-1}$.)

2.1. Realizing a polynomial in the monomial basis: Observer–type realization

We begin with an example of constructing a realization of a polynomial (in this example of degree three) expressed in the monomial basis, i.e., a polynomial of the form

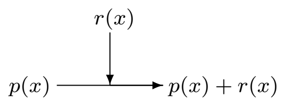
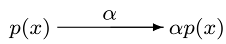
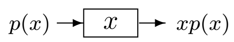
$$H(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3.$$

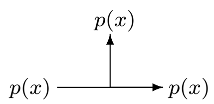
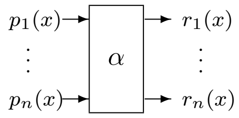
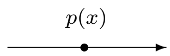
One first uses so–called "delay elements" to implement multiplication by $x = z^{-1}$, and draws the delay line, as in

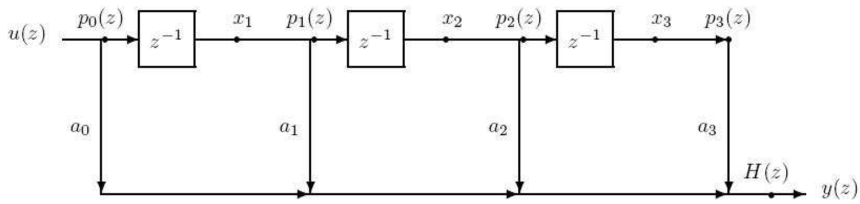


It is easy to see that the inputs of each delay element are simply the monomials 1, x, and x^2, and the outputs are x, x^2, and x^3, all of the building blocks needed to form the polynomial $H(x)$. Then $H(x)$ is formed as a linear combination of these by attaching taps, as in

TABLE 3. Building blocks of signal flow graphs

Adder	Gain	Delay
		
Implements polynomial addition.	Implements scalar multiplication.	Implements multiplication by x.

Splitter	Linear transformation	Label
		
Allows a given signal to be used in multiple places.	Combination of other components to implement matrix–vector products; $r_{1:n} = \alpha \times p_{1:n}$.	Identifies the current signal (just for clarity, does not require an actual device).



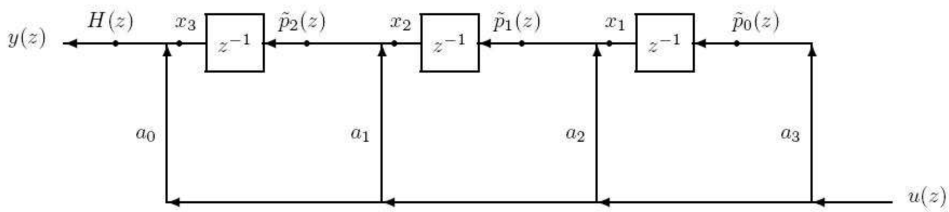
Such a realization is canonical, and is called the *observer–type realization*, as by modifying the gains on the taps, one can observe the values of the states.

2.2. Realizing a polynomial in the Horner basis: Controller–type realization

While this realization is canonical, it is not unique. As stated in the introduction, one can represent the polynomial $H(x)$ as

$$\begin{aligned} H(x) &= a_0 + a_1 x + a_2 x^2 + a_3 x^3 \\ &= a_0 + x(a_1 + x(a_2 + x(a_3))) \end{aligned}$$

leading to the well–known Horner rule for polynomial evaluation. Specifically, the recurrence relations (1.1) for the Horner polynomials allow one to evaluate the polynomial $H(x)$, and so the following realization using Horner polynomials also realizes the same polynomial.



This realization is also canonical, and is called the *controller–type realization*. This name is because by modifying the gains on the taps, it is possible to directly control the inputs to the delay elements.

We conclude this section with the observation that going from the observer–type realization to the controller–type realization involves the passage from using a basis of monomials to a basis of Horner polynomials.

2.3. Key concept: Flow reversal and Horner polynomials

The key observation that we wish to make using this example is that, comparing the observer–type and controller–type realizations, we see that **one is obtained from the other by reversing the direction of the flow**. In particular, the flow reversal of the signal flow graph corresponds to changing from the basis of monomials to that of Horner polynomials.

In this section, this was illustrated for the monomial–classical Horner case. The next procedure, proved in [18], states that this observation is true in general. That is, by constructing a signal flow graph in a specific way, one can determine recurrence relations for *generalized Horner polynomials* for any given system of polynomials.

Procedure 2.1 (Obtaining generalized Horner polynomials). *Given a system of polynomials* $R = \{r_0(x), r_1(x), \ldots, r_n(x)\}$ *satisfying* $\deg r_k(x) = k$*, the system of generalized Horner polynomials* $\tilde{R}$ *corresponding to* R *can be found by the following procedure.*

1. *Draw a minimal*[2] *signal flow graph for the linear time–invariant system with the overall transfer function* $H(x)$*, and such that* $r_k(x)$ *are the partial transfer functions from the input of the signal flow graph to the input of the* k*–th delay element for* $k = 1, 2, \ldots, n-1$.
2. *Reverse the direction of the flow of the signal flow graph to go from the observer–type realization to the controller–type realization.*
3. *Identify the* generalized Horner polynomials $\tilde{R} = \{\tilde{r}_k(x)\}$ *as the partial transfer functions from the input of the signal flow graph to the inputs of the delay elements.*
4. *Read from the reversed signal flow graph a recursion for* $\tilde{R} = \{\tilde{r}_k(x)\}$.

We emphasize at this point that this process is valid for arbitrary systems of polynomials. In the next section, details of some special classes of polynomials

[2] A signal flow graph is called *minimal* in engineering literature if it contains the minimal number n of delay elements. Such minimal realizations where, in this case, $n = \deg H(X)$, always exist.

and their corresponding new filter structures for which this process can be used to yield fast algorithms will be introduced. The goal is then to use these new structures to derive new Horner–like rules, and to then invert the corresponding polynomial–Vandermonde matrices, as described in the introduction.

3. New quasiseparable filter structures

3.1. Interplay between structured matrices and systems of polynomials

In the previous section, details of how to use signal flow graphs to obtain recurrence relations for generalized Horner polynomials associated with an arbitrary system of polynomials were given. In this section, we introduce several new filter structures for which the recurrence relations that result from this procedure are sparse. In order to define these new structures, we will use the interplay between structured matrices and systems of polynomials. At the heart of many fast algorithms involving polynomials are a relation to a class of structured matrices, and so such a relation introduced next should seem natural.

Definition 3.1. *A system of polynomials R is related to a strongly upper Hessenberg (i.e. upper Hessenberg with nonzero subdiagonal elements: $a_{i,j} = 0$ for $i > j + 1$, and $a_{i+1,i} \neq 0$ for $i = 1, \dots, n-1$) matrix A (and vice versa) provided*

$$r_k(x) = \frac{1}{a_{2,1}a_{3,2}\cdots a_{k,k-1}} \det\left(xI - A\right)_{(k\times k)}, \quad k = 1, \dots, n. \tag{3.1}$$

That is, we associate with a Hessenberg matrix the system of polynomials formed from characteristic polynomials of its principal submatrices. It can readily be seen that given a Hessenberg matrix, a related system of polynomials may be constructed. The opposite direction can be seen using the concept of a so–called *confederate matrix* of [17], recalled briefly next.

Proposition 3.2. *Let R be a system of polynomials satisfying the n–term recurrence relations*[3]

$$x \cdot r_{k-1}(x) = a_{k+1,k} \cdot r_k(x) - a_{k,k} \cdot r_{k-1}(x) - \cdots - a_{1,k} \cdot r_0(x), \tag{3.2}$$

for $k = 1, \dots, n$, with $a_{k+1,k} \neq 0$. Then the matrix[4]

$$C_R = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,n} \\ 0 & a_{3,2} & a_{3,3} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_{n-1,n} \\ 0 & \cdots & 0 & a_{n,n-1} & a_{n,n} \end{bmatrix}, \tag{3.3}$$

is related to R as in Definition 3.1.

[3] It is easy to see that any polynomial system $\{r_k(x)\}$ satisfying $\deg r_k(x) = k$ satisfies (3.2) for some coefficients.

[4] Notice that this matrix does not restrict the constant polynomial $r_0(x)$ at all, and hence it may be chosen freely. What is important is that there exists such a matrix.

In the next two sections, special structured matrices related to the new filter structures are introduced.

3.2. (H, m)**–quasiseparable matrices and filter structures**

Definition 3.3 **((H, m)–quasiseparable matrices).** *A matrix A is called (H, m)–-quasiseparable if* **(i)** *it is strongly upper Hessenberg (i.e. upper Hessenberg with nonzero subdiagonal elements: $a_{i,j} = 0$ for $i > j + 1$, and $a_{i+1,i} \neq 0$ for $i = 1, \dots, n-1$), and*
(ii) $\max(rank A_{12}) = m$ *where the maximum is taken over all symmetric partitions of the form*

$$A = \left[\begin{array}{c|c} * & A_{12} \\ \hline * & * \end{array}\right];$$

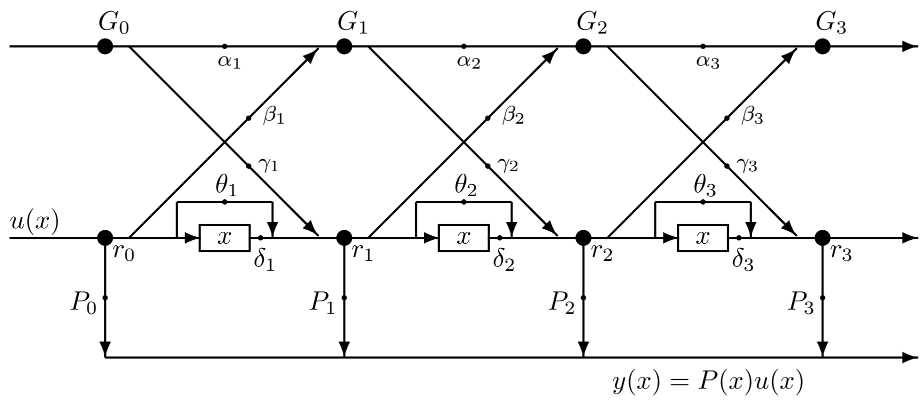
for instance, the low–rank blocks of a 5×5 (H, m)–quasiseparable matrix would be those shaded below:

$$\left[\begin{array}{c|cccc} \star & \star & \star & \star & \star \\ \hline \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star \\ 0 & 0 & \star & \star & \star \\ 0 & 0 & 0 & \star & \star \end{array}\right] \quad \left[\begin{array}{cc|ccc} \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star \\ \hline 0 & \star & \star & \star & \star \\ 0 & 0 & \star & \star & \star \\ 0 & 0 & 0 & \star & \star \end{array}\right]$$

$$\left[\begin{array}{ccc|cc} \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star \\ \hline 0 & 0 & \star & \star & \star \\ 0 & 0 & 0 & \star & \star \end{array}\right] \quad \left[\begin{array}{cccc|c} \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star \\ 0 & \star & \star & \star & \star \\ 0 & 0 & \star & \star & \star \\ \hline 0 & 0 & 0 & \star & \star \end{array}\right]$$

The following theorem gives the filter structure that results from the systems of polynomials related to matrices with this quasiseparable structure when $m = 1$; that is, what we suggest to call $(H, 1)$–quasiseparable filter structure.

Theorem 3.4. *A system of polynomials $\{r_k(x)\}$ is related to an $(H, 1)$–quasiseparable matrix if and only if they admit the realization*



Algebraic proofs of the results of this section can be found in [2], [5], but here we give a proof using the language of the signal flow graphs.

Proof. Suppose $\{r_k(x)\}$ admit the shown realization. Then by reading from the signal flow graph, it can be readily seen that each $r_k(x)$ satisfies the n–term recurrence relations

$$\begin{aligned} r_k(x) &= (\delta_k x + \theta_k) r_{k-1}(x) + \gamma_k \beta_{k-1} r_{k-2}(x) + \gamma_k \alpha_{k-1} \beta_{k-2} r_{k-3}(x) \\ &\quad + \gamma_k \alpha_{k-1} \alpha_{k-2} \beta_{k-3} r_{k-4}(x) + \cdots + \gamma_k \alpha_{k-1} \cdots \alpha_2 \beta_1 r_0(x). \end{aligned}$$

Using Proposition 3.2 and these n–term recurrence relations, we have that the matrix

$$\begin{bmatrix} -\frac{\theta_1}{\delta_1} & -\frac{1}{\delta_2}\gamma_2\beta_1 & -\frac{1}{\delta_3}\gamma_3\alpha_2\beta_1 & \cdots & -\frac{1}{\delta_n}\gamma_n\alpha_{n-1}\alpha_{n-2}\cdots\alpha_3\alpha_2\beta_1 \\ \frac{1}{\delta_1} & -\frac{\theta_2}{\delta_2} & -\frac{1}{\delta_3}\gamma_3\beta_2 & \cdots & -\frac{1}{\delta_n}\gamma_n\alpha_{n-1}\alpha_{n-2}\cdots\alpha_3\beta_2 \\ 0 & \frac{1}{\delta_2} & -\frac{\theta_3}{\delta_3} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & -\frac{1}{\delta_n}\gamma_n\beta_{n-1} \\ 0 & \cdots & 0 & \frac{1}{\delta_{n-1}} & -\frac{\theta_n}{\delta_n} \end{bmatrix} \tag{3.4}$$

is related to the polynomial system $\{r_k(x)\}$. It can be observed that the off–diagonal blocks as in Definition 3.3 are all of rank one, and so the polynomial system $\{r_k(x)\}$ is indeed related to an $(H, 1)$–quasiseparable matrix. The opposite direction is proven using the observation that *any* $(H, 1)$–quasiseparable matrix can be written in the form (3.4) (such is called the *generator representation*, and details can be found in [11], [2]). This completes the proof. □

An analogous proof later in this section for $(H, 1)$–semiseparable polynomials and their realizations would follow the exact same pattern, and thus is omitted. An immediate consequence of Theorem 3.4 are recurrence relations that can be read off of the signal flow graph.

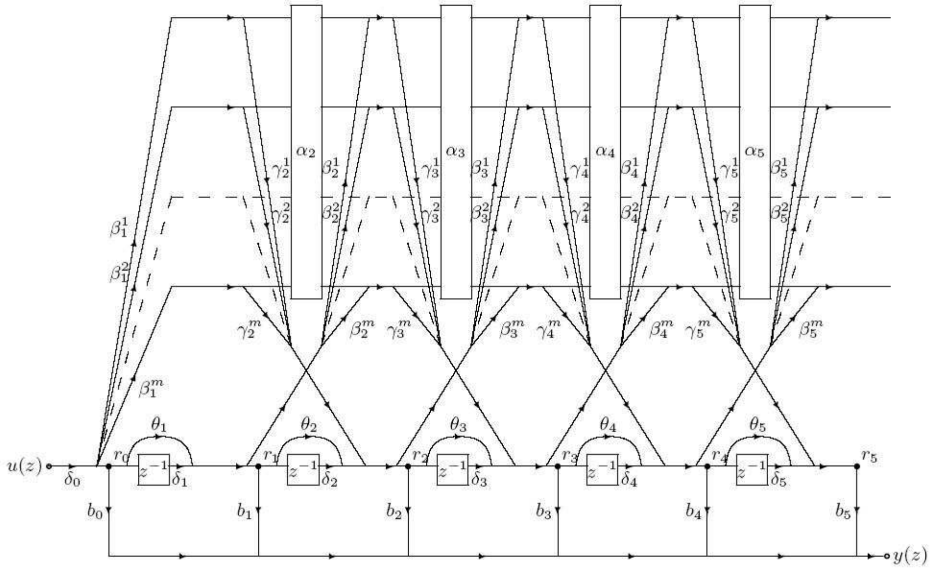
Corollary 3.5. *The polynomials $\{r_k(x)\}$ are related to an $(H, 1)$–quasiseparable matrix if and only if they satisfy the two–term recurrence relations*

$$\begin{bmatrix} F_k(x) \\ r_k(x) \end{bmatrix} = \begin{bmatrix} \alpha_k & \beta_k \\ \gamma_k & \delta_k x + \theta_k \end{bmatrix} \begin{bmatrix} F_{k-1}(x) \\ r_{k-1}(x) \end{bmatrix} \tag{3.5}$$

for some system of auxiliary polynomials $\{F_k(x)\}$ and some scalars α_k, β_k, γ_k, δ_k, and θ_k.

The next theorem extends Theorem 3.4 to give the realization of polynomials related to an (H, m)–quasiseparable matrix. The essential difference in going to the order m case is that m additional (non–delay) lines are required in the realization, whereas in the order 1 case only one is required.

Theorem 3.6. *A system of polynomials $\{r_k(x)\}$ is related to an (H, m)–quasiseparable matrix if and only if they admit the realization*



Although the signal flow graph of the realization in this theorem is considerably more complicated than that of Theorem 3.4 for $(H, 1)$–quasiseparable matrices, the proof follows in the same manner, however involving vectors and matrices instead of scalars. Affording simpler generalizations is a feature of working with signal flow graphs. For an algebraic proof, see [5].

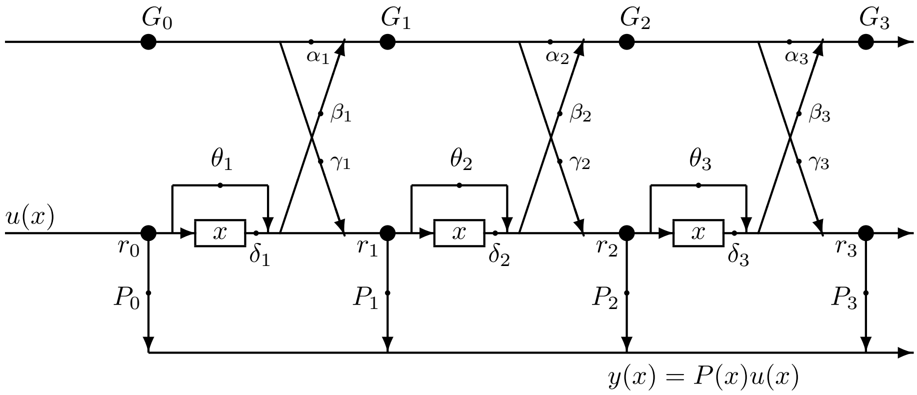
3.3. (H, m)–semiseparable matrices and filter structures

Definition 3.7 (**(H, m)–semiseparable matrices**). *A matrix A is called (H, m)–-semiseparable if* **(i)** *it is strongly upper Hessenberg (i.e. upper Hessenberg with nonzero subdiagonal elements: $a_{i,j} = 0$ for $i > j + 1$, and $a_{i+1,i} \neq 0$ for $i = 1, \ldots, n-1$), and* **(ii)** *it is of the form*

$$A = B + \operatorname{triu}(A_U, 1)$$

with rank$(A_U) = m$ and a lower bidiagonal matrix B, where following the MATLAB command ***triu***, $\operatorname{triu}(A_U, 1)$ *denotes the strictly upper triangular portion of the matrix A_U.*

Theorem 3.8. *A system of polynomials $\{r_k(x)\}$ is related to an $(H, 1)$–semiseparable matrix if and only if they admit the realization*



with $G_0(x) = 1$.

The proof is given by reading the n–term recurrence relations off of the signal flow graph of the given realization as

$$\begin{aligned}
r_k(x) &= (\delta_k x + \theta_k + \gamma_k\beta_{k-1})r_{k-1}(x) + \gamma_k(\alpha_{k-1} - \beta_{k-1}\gamma_{k-1})\beta_{k-2}r_{k-2}(x) \\
&\quad + \gamma_k(\alpha_{k-1} - \beta_{k-1}\gamma_{k-1})(\alpha_{k-2} - \beta_{k-2}\gamma_{k-2})\beta_{k-3}r_{k-3}(x) + \cdots + \\
&\quad + \gamma_k(\alpha_{k-1} - \beta_{k-1}\gamma_{k-1})(\alpha_{k-2} - \beta_{k-2}\gamma_{k-2})\cdots(\alpha_2 - \beta_2\gamma_2)\beta_1 r_1(x) + \\
&\quad + \gamma_k(\alpha_{k-1} - \beta_{k-1}\gamma_{k-1})(\alpha_{k-2} - \beta_{k-2}\gamma_{k-2})\cdots(\alpha_1 - \beta_1\gamma_1)r_0(x)
\end{aligned}$$

and then relating them via Proposition 3.2 to a matrix shown to have $(H, 1)$–semiseparable structure, following the blueprint of Theorem 3.4.

Just as for the quasiseparable filter structure, recurrence relations for the $(H, 1)$–semiseparable polynomials can be read off of the signal flow graph of this realization.

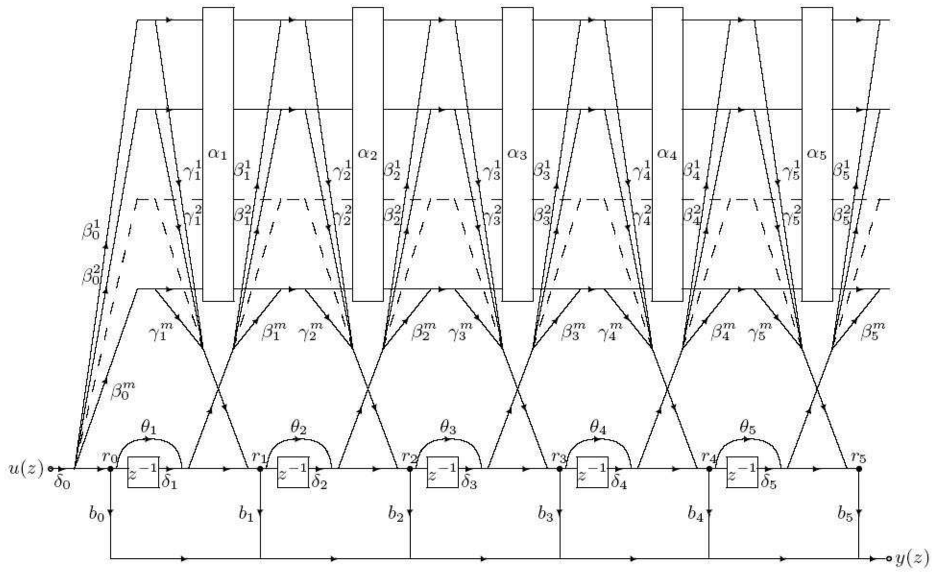
Corollary 3.9. *The polynomials $\{r_k(x)\}$ are related to an $(H, 1)$–semiseparable matrix if and only if they satisfy the two–term recurrence relations*

$$\begin{bmatrix} G_k(x) \\ r_k(x) \end{bmatrix} = \begin{bmatrix} \alpha_k & \beta_k \\ \gamma_k & 1 \end{bmatrix} \begin{bmatrix} G_{k-1}(x) \\ (\delta_k x + \theta_k)r_{k-1}(x) \end{bmatrix} \tag{3.6}$$

for some system of auxiliary polynomials $\{G_k(x)\}$ and some scalars α_k, β_k, γ_k, δ_k, and θ_k.

As for Theorem 3.4, we next extend the realization of Theorem 3.8 to the order m case.

Theorem 3.10. *A system of polynomials $\{r_k(x)\}$ is related to an (H, m)–semiseparable matrix if and only if they admit the realization*



4. Special cases of the new filter structures

In this brief section, we enumerate some well–known special cases of the polynomials given in the previous section. As subclasses of these polynomials, they are then also special cases of polynomials that may be realized by using the new filter structures presented, and hence are examples of classes for which the problems solved in this paper may be applied to.

4.1. Monomials

In Section 2, the first motivating example of monomials and Horner polynomials was considered. Monomials are in fact special cases of $(H, 1)$–quasiseparable polynomials, as well as of $(H, 1)$–semiseparable polynomials, and hence both filter structures of the previous section can be used to realize the monomial system.

4.2. Real–orthogonal polynomials

A second well–known class of polynomials to which these new filter structures may be applied are polynomial systems orthogonal with respect to some inner product on the real line. Such polynomials are well–known to satisfy three–term recurrence relations of the form

$$r_k(x) = (\alpha_k x - \delta_k) r_{k-1}(x) - \gamma_k r_{k-2}(x), \tag{4.1}$$

from which one can easily draw a signal flow graph of the form shown in Figure 3.

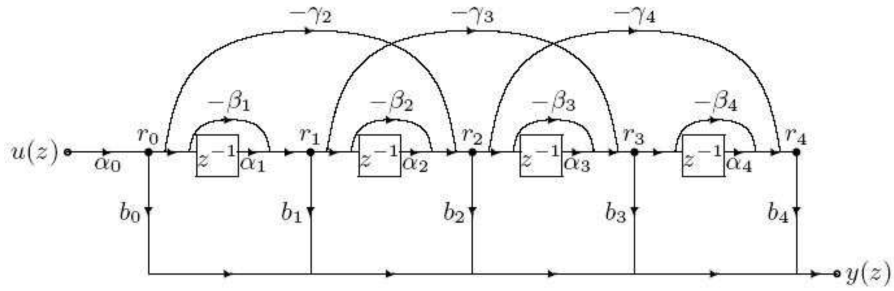


FIGURE 3. Signal flow graph realizing real–orthogonal polynomials.

Real–orthogonal polynomials are subclasses of $(H, 1)$–quasiseparable polynomials, and so the new filter structures may also be used to realized real–orthogonal polynomials.

4.3. Szegö polynomials: Markel–Gray filter structure

Another example of a common class of polynomials for which these filter structures are applicable is that of the Szegö polynomials $\{\phi_k^{\#}\}$, or those orthogonal with respect to an inner product on the unit circle. Such polynomial systems are known to satisfy the two–term recurrence relations

$$\begin{bmatrix} \phi_k(x) \\ \phi_k^{\#}(x) \end{bmatrix} = \frac{1}{\mu_k} \begin{bmatrix} 1 & -\rho_k \\ -\rho_k^* & 1 \end{bmatrix} \begin{bmatrix} \phi_{k-1}(x) \\ x\phi_{k-1}^{\#}(x) \end{bmatrix}. \tag{4.2}$$

involving a system of auxiliary polynomials $\{\phi_k\}$. Such recurrence relations lead to the Markel–Gray filter structure shown in Figure 4.

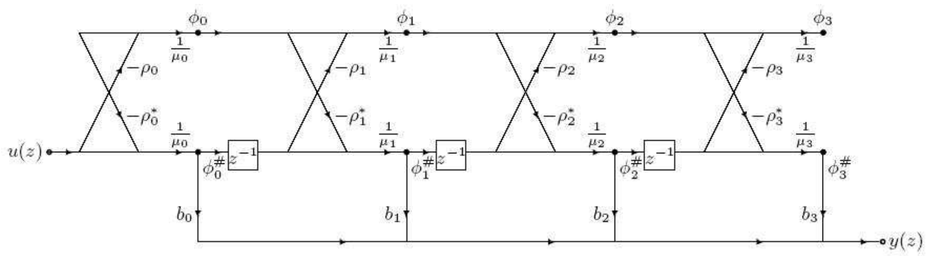


FIGURE 4. Signal flow graph showing the Markel–Gray filter structure, realizing Szegö polynomials.

The $(H, 1)$–semiseparable filter structure is a direct generalization of the Markel–Gray filter structure, and hence semiseparable filters can be used to realize Szegö polynomials.

Furthermore, Szegö polynomials are not only $(H, 1)$–semiseparable, but $(H, 1)$–quasiseparable as well, and hence one can also use the quasiseparable filter structures to realize Szegö polynomials. The semiseparable and quasiseparable filter structures are considerably different, notably in the locations of the delay elements with respect to the cross–connections, and next in Figure 5, the reduction of the quasiseparable filter structure to the Szegö case is given.

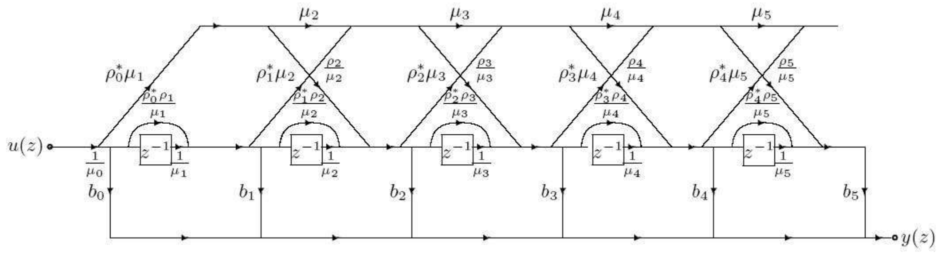


FIGURE 5. Signal flow graph realizing Szegö polynomials using an $(H, 1)$–quasiseparable filter structure.

Notice that new two–term recurrence relations for Szegö polynomials can be read directly from the signal flow graph of Figure 5. Such recurrence relations were derived algebraically in [3], and are found to be

$$\begin{bmatrix} \phi_0(x) \\ \phi_0^{\#}(x) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \phi_k(x) \\ \phi_k^{\#}(x) \end{bmatrix} = \begin{bmatrix} \mu_k & \rho_{k-1}^*\mu_k \\ \frac{\rho_k}{\mu_k} & \frac{x+\rho_{k-1}^*\rho_k}{\mu_k} \end{bmatrix} \begin{bmatrix} \phi_{k-1}(x) \\ \phi_{k-1}^{\#}(x) \end{bmatrix}. \tag{4.3}$$

5. Horner–type polynomial evaluation rules for (H, m)–quasiseparable polynomials

As was described in Section 2, one can use the Horner polynomials to evaluate a polynomial given in the monomial basis. The crux of the trick is that if a polynomial $H(x)$ is given in the basis of the first n monomials, then, while the values of the first $n-1$ Horner polynomials may differ from the first $n-1$ monomial bases, the n–th will coincide; that is, the last Horner polynomial $\tilde{p}_n(x) = H(x)$, so $H(x)$ may be evaluated using the recurrence relations for $\{\tilde{p}_k(x)\}$ of (1.1). In terms of systems theory, it is known that flow reversal does not change the overall transfer function, which is essentially the same statement.

In this section, we use Procedure 2.1 and the new filter structures to demonstrate generalizations of this algorithm. We then provide the special cases of the two previously considered cases of real–orthogonal polynomials and Szegö polynomials.

5.1. New evaluation rule: Polynomials in a quasiseparable basis

Assume that, given a polynomial in a basis of (H, m)–quasiseparable polynomials, the value of that polynomial at a given point is to be determined in a similar manner as the Horner rule for the monomials basis. The method is contained in the following theorem.

Theorem 5.1. *Let*

$$H(x) = b_0 r_0(x) + b_1 r_1(x) + \cdots + b_n r_n(x)$$

be a polynomial expressed in a basis of (H, m)–quasiseparable polynomials. Then $H(x)$ can be evaluated using the recurrence relations

$$\begin{bmatrix} \tilde{F}_k(x) \\ \tilde{r}_k(x) \end{bmatrix} = \begin{bmatrix} 0 \\ b_n \end{bmatrix}$$

$$\begin{bmatrix} \tilde{F}_k(x) \\ \tilde{r}_k(x) \end{bmatrix} = \begin{bmatrix} \alpha_{n-k+1}^T & \frac{1}{\delta_{n-k+1}} \gamma_{n-k+1}^T \\ \delta_{n-k} \beta_{n-k+1}^T & \delta_{n-k} x + \frac{\theta_{n-k+1}}{\delta_{n-k+1}} \end{bmatrix} \begin{bmatrix} \tilde{F}_k(x) \\ \tilde{r}_k(x) \end{bmatrix} + \begin{bmatrix} 0 \\ \delta_{n-k} b_{n-k} \end{bmatrix}$$

and the relation $H(x) = \tilde{r}_n(x)$.

Proof. Following Procedure 2.1, the signal flow graph of Theorem 3.6 is reversed to obtain that of Figure 6.

From the signal flow graph in Figure 6, the stated recurrence relations for the generalized Horner polynomials associated with (H, m)–quasiseparable polynomials are observed. □

5.2. New evaluation rule: Polynomials in a semiseparable basis

Theorem 5.2. *Let*

$$H(x) = b_0 r_0(x) + b_1 r_1(x) + \cdots + b_n r_n(x)$$

be a polynomial expressed in a basis of (H, m)–semiseparable polynomials. Then $H(x)$ can be evaluated using the recurrence relations

$$\begin{bmatrix} \tilde{G}_0(x) \\ \tilde{r}_0(x) \end{bmatrix} = \begin{bmatrix} -b_n \beta_n^T \\ b_n \end{bmatrix}$$

$$\begin{bmatrix} \tilde{G}_k(x) \\ \tilde{r}_k(x) \end{bmatrix} = \begin{bmatrix} \alpha_{n-k}^T & \gamma_{n-k}^T \\ \delta_{n-k} \beta_{n-k}^T & \delta_{n-k} \end{bmatrix} \begin{bmatrix} \tilde{G}_k(x) \\ \left(x + \frac{\theta_{n-k+1}}{\delta_{n-k+1}} \right) \tilde{r}_k(x) + b_{n-k} \end{bmatrix}$$

and the relation $H(x) = \tilde{r}_n(x)$.

Proof. Following Procedure 2.1, the signal flow graph of Theorem 3.10 is reversed to obtain that of Figure 7.

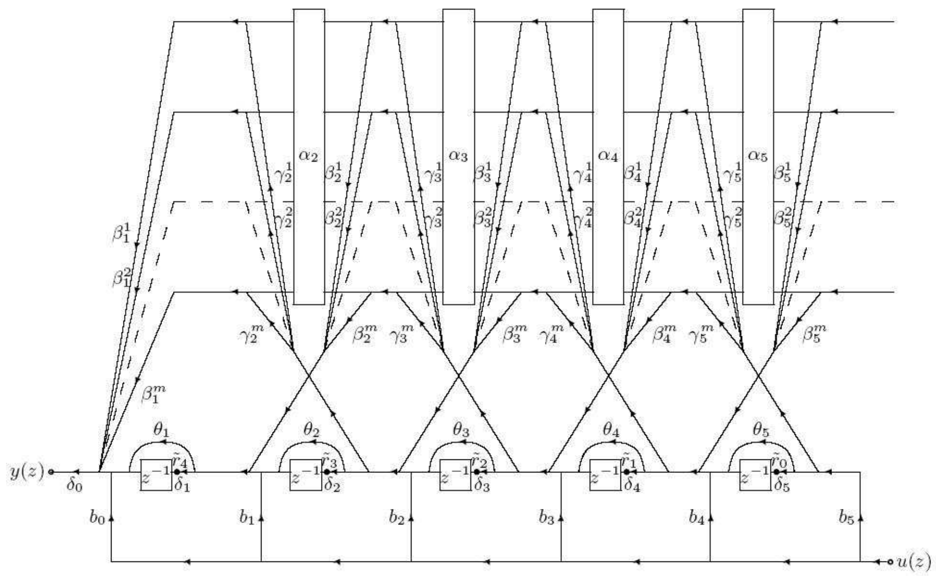


FIGURE 6. Signal flow graph of the reversal of the (H, m)–quasi-separable filter structure.

From this signal flow graph, the stated recurrence relations for the generalized Horner polynomials associated with (H, m)–semiseparable polynomials can be read off. □

5.3. Polynomials in a real orthogonal basis: The Clenshaw rule

We next consider some classical cases which are special cases of the given filter structures. Suppose we are given a polynomial $H(x)$ in the basis of real–orthogonal polynomials, i.e. satisfying the three–term recurrence relations (4.1), with the goal of evaluating said polynomial at some value x. Applying Procedure 2.1, we first draw a signal flow graph of the observer–type for real–orthogonal polynomials, and reverse the flow to find the generalized Horner polynomials. The former signal flow graph was presented in Figure 3, and we next present the latter in Figure 8.

From Figure 8, one can read off the recurrence relations satisfied by the generalized Horner polynomials as

$$\begin{aligned}\tilde{r}_k(x) \quad &= \quad \alpha_{n-k} x \tilde{r}_{k-1}(x) - \frac{\alpha_{n-k}}{\alpha_{n-k+1}} \beta_{n-k+1} \tilde{r}_{k-1}(x) \\ &\quad - \frac{\alpha_{n-k}}{\alpha_{n-k+2}} \gamma_{n-k+2} \tilde{r}_{k-2}(x) + b_{n-k}, \end{aligned} \tag{5.1}$$

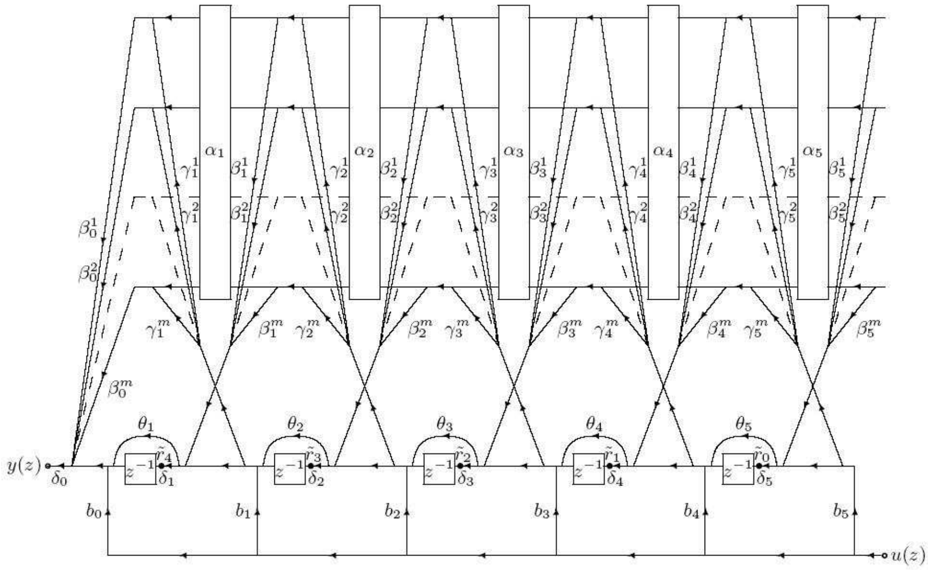


FIGURE 7. Signal flow graph of the reversal of the (H, m)–semiseparable filter structure.

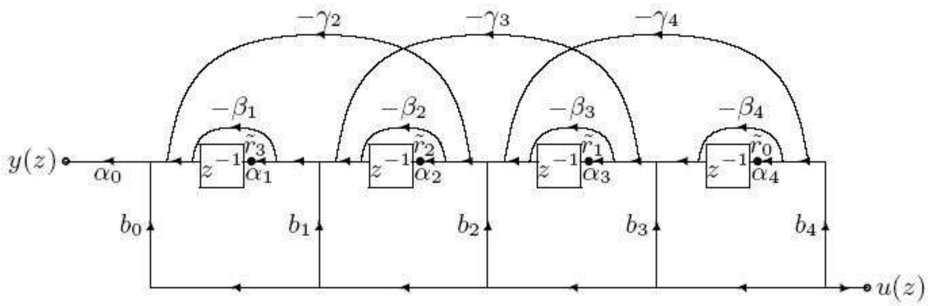


FIGURE 8. Reversal of the signal flow graph realizing real–orthogonal polynomials.

which is the well–known Clenshaw rule, an extension of the Horner rule to the basis of real–orthogonal polynomials.

5.4. Polynomials in a Szegö basis

As above, if one needs to evaluate a polynomial given in a basis of Szegö polynomials using the Horner rule, it can be done by using recurrence relations found by reversing the flow of the Markel–Gray filter structure of Figure 4. The reversed signal flow graph is shown in Figure 9.

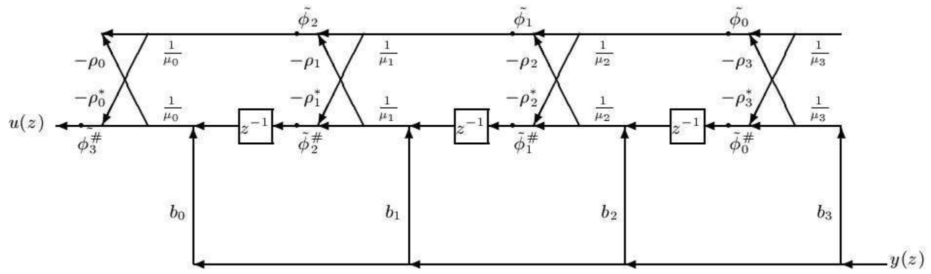


FIGURE 9. Reversal of the signal flow graph showing the Markel–Gray filter structure realizing Szegö polynomials.

From the reversed Markel–Gray filter structure in Figure 9, one can directly read the following recurrence relations for the generalized Horner polynomials. They are read as

$$\begin{bmatrix} \tilde{\phi}_k(x) \\ \tilde{\phi^\#}_k(x) \end{bmatrix} = \frac{1}{\mu_{n-k}} \begin{bmatrix} 1 & -\rho_{n-k} \\ -\rho^*_{n-k} & 1 \end{bmatrix} \begin{bmatrix} \tilde{\phi}_{k-1}(x) \\ x\tilde{\phi^\#}_{k-1}(x) + b_{n-k} \end{bmatrix}. \tag{5.2}$$

These recurrence relations, among others including three–term, n–term, and shifted n–term, were introduced in [18].

5.4.1. The Ammar–Gragg–Reichel algorithm. It was noted by Olshevsky in [18] that these two–term recurrence relations (5.2) for the generalized Horner polynomials related to Szegö polynomials are not the same as the result of an algebraic derivation of the same by Ammar, Gragg, and Reichel in [1]. There, the authors derived the recursion

$$\begin{bmatrix} \tau_n \\ \tilde{\tau}_n \end{bmatrix} = \begin{bmatrix} \frac{b_n}{\mu_n} \\ 0 \end{bmatrix}, \quad \begin{bmatrix} \tau_k \\ \tilde{\tau}_k \end{bmatrix} = \frac{1}{\mu_k} \begin{bmatrix} b_k + x(\tau_{k+1} + \rho^*_{k+1}\tilde{\tau}_{k+1}) \\ \rho_{k+1}\tau_{k+1} + \tilde{\tau}_{k+1} \end{bmatrix}, \tag{5.3}$$

where $H(x) = \tau_0 + \tilde{\tau}_0$. Indeed, if one draws a signal flow graph in Figure 10 depicting these relations, the difference becomes apparent. Procedure 2.1, Step 3 states that the generalized Horner polynomials are to be chosen as the partial transfer functions to the inputs of the delays, but this is not the case in Figure 10. That is, the recursion (5.3) is based on a different choice of polynomials than the generalized Horner polynomials.

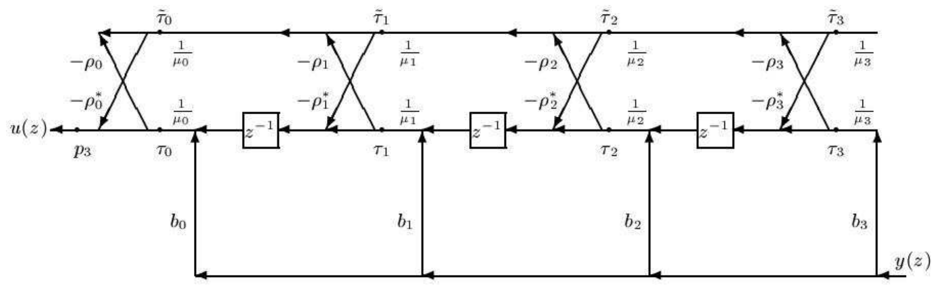


FIGURE 10. Signal flow graph depicting the recursion of the Ammar–Gragg–Reichel algorithm.

5.4.2. A new algorithm based on the quasiseparable filter structure. In Section 4.3, it was noticed that because Szegö polynomials are subclasses of both $(H, 1)$–semiseparable and $(H, 1)$–quasiseparable polynomials, both of the corresponding filter structures can be used to realize Szegö polynomials. It was further seen that using the $(H, 1)$–semiseparable filter structure reduced to the well–known Markel–Gray filter structure of [16], and that using the $(H, 1)$–quasiseparable filter structure yielded new result, including the new recurrence relations 4.3.

Such results also apply to the generalized Horner polynomials associated with Szegö polynomials. By reversing the flow in the $(H, 1)$–semiseparable filter structure (Markel–Gray in this special case), the recurrence relations 5.2 above[5].

Reversing the flow of the $(H, 1)$–quasiseparable filter structure yields a new set of recurrence relations for the generalized Horner polynomials associated with the Szegö polynomials. Specifically, reading off the reversal of the signal flow graph in Figure 5, one arrives at the recurrence relations

$$\begin{bmatrix} \tilde{\phi}_0(x) \\ \tilde{\phi}_0^{\#}(x) \end{bmatrix} = \begin{bmatrix} 0 \\ b_n \end{bmatrix}$$

$$\begin{bmatrix} \tilde{\phi}_k(x) \\ \tilde{\phi}_k^{\#}(x) \end{bmatrix} = \begin{bmatrix} \mu_{n-k+1} & \frac{\rho_{n-k+1}}{\mu_{n-k+1}} \\ \rho_{n-k}^*\mu_{n-k+1} & \frac{x+\rho_{n-k}^*\rho_{n-k+1}}{\mu_{n-k+1}} \end{bmatrix} \begin{bmatrix} \tilde{\phi}_{k-1}(x) \\ \tilde{\phi}_{k-1}^{\#}(x) \end{bmatrix} + \begin{bmatrix} 0 \\ b_{n-k} \end{bmatrix}.$$

6. (H, m)–quasiseparable eigenvector problem

In this section, the second problem of the paper is solved, namely the eigenvector computation of (H, m)–quasiseparable matrices and their subclasses.

It can be easily verified that

$$V_R(x)C_R = D(x)V_R(x), \quad D(x) = diag(x_1, x_2, \ldots, x_n),$$

[5] And, as stated above, by moving the locations of the polynomials in the signal flow graph, one also gets the recurrence relations of Ammar, Gragg, and Reichel in [1].

which implies that the columns of the inverse of polynomial Vandermonde matrix $V_R(x)^{-1}$ store the eigenvectors of the confederate matrix C_R of Proposition 3.2. Thus, in order to compute the eigenvectors of a matrix C_R, one need only to invert the polynomial–Vandermonde matrix $V_R(x)$ formed by polynomials corresponding to the matrix $C_R(H)$, a topic described in detail in Section 7.

Special cases of confederate matrices C_R described in this paper include (H,m)–quasiseparable matrices as well as (H,m)–semiseparable matrices, and hence this procedure allows one to compute eigenvectors of both of these classes of matrices, given their eigenvalues. As special cases of these structures, tridiagonal matrices, unitary Hessenberg matrices, upper–banded matrices, etc. also can have their eigenvectors computed via this method.

7. Inversion of (H,m)–quasiseparable–Vandermonde matrices

In this section we address the problem of inversion of polynomial–Vandermonde matrices of the form

$$V_R(x) = \begin{bmatrix} r_0(x_1) & r_1(x_1) & \cdots & r_{n-1}(x_1) \\ r_0(x_2) & r_1(x_2) & \cdots & r_{n-1}(x_2) \\ \vdots & \vdots & & \vdots \\ r_0(x_n) & r_1(x_n) & \cdots & r_{n-1}(x_n) \end{bmatrix}, \tag{7.1}$$

with specific attention, as elsewhere in the paper, to the special case where the polynomial system $R = \{r_0(x),\ r_1(x),\ \ldots,\ r_{n-1}\}$ are (H,m)–quasiseparable or (H,m)–semiseparable. The following proposition is an extension of one for the classical Vandermonde case by Traub [20], whose proof in terms of signal flow graphs may be found in [18].

Proposition 7.1. *Let $R = \{r_0(x),\ r_1(x),\ \ldots,\ r_{n-1}(x)\}$ be a system of polynomials, and $H(x)$ a monic polynomial with exactly n distinct roots. Then the polynomial–Vandermonde matrix $V_R(x)$ whose nodes $\{x_k\}$ are the zeros of $H(x)$ has inverse*

$$V_R(x)^{-1} = \begin{bmatrix} \tilde{r}_{n-1}(x_1) & \tilde{r}_{n-1}(x_2) & \cdots & \tilde{r}_{n-1}(x_n) \\ \vdots & \vdots & & \vdots \\ \tilde{r}_1(x_1) & \tilde{r}_1(x_2) & \cdots & \tilde{r}_1(x_n) \\ \tilde{r}_0(x_1) & \tilde{r}_0(x_2) & \cdots & \tilde{r}_0(x_n) \end{bmatrix} \cdot D, \tag{7.2}$$

with

$$D = diag\,(H'(x_i)) = diag\left(\frac{1}{\Pi_{\substack{k=1\\k\neq i}}^{n}(x_k - x_i)}\right),$$

involving the generalized Horner polynomials $\tilde{R} = \{\tilde{r}_0(x),\ \tilde{r}_1(x),\ \ldots,\ \tilde{r}_{n-1}(x)\}$ defined in Procedure 2.1.

From this proposition, we see that the main computational burden in computing the inverse of a polynomial–Vandermonde matrix is in evaluating the generalized Horner polynomials as each of the nodes. But Procedure 2.1, illustrated in the previous sections for several examples is exactly a procedure for determining efficient recurrence relations for just these polynomials, and evaluating them at given points.

So the procedure of the above sections is exactly a procedure for inversion of the related polynomial–Vandermonde matrix; that is, reversing the flow of the signal flow graph corresponds to inverting the related polynomial–Vandermonde matrix. We state the following two corollaries of this proposition and also Theorems 5.1 and 5.2, respectively, allowing fast inversion of (H,m)–quasiseparable Vandermonde systems and (H,m)–semiseparable Vandermonde systems, respectively.

Corollary 7.2. *Let R be a system of (H,m)–quasiseparable polynomials given in terms of recurrence relation coefficients, and $H(x)$ a monic polynomial with exactly n distinct roots. Then the (H,m)–quasiseparable Vandermonde matrix $V_R(x)$ whose nodes $\{x_k\}$ are the zeros of $H(x)$ can be inverted as*

$$V_R(x)^{-1} = \begin{bmatrix} \tilde{r}_{n-1}(x_1) & \tilde{r}_{n-1}(x_2) & \cdots & \tilde{r}_{n-1}(x_n) \\ \vdots & \vdots & & \vdots \\ \tilde{r}_1(x_1) & \tilde{r}_1(x_2) & \cdots & \tilde{r}_1(x_n) \\ \tilde{r}_0(x_1) & \tilde{r}_0(x_2) & \cdots & \tilde{r}_0(x_n) \end{bmatrix} \cdot D,$$

with

$$D = diag\,(H'(x_i)) = diag\left(\frac{1}{\Pi_{\substack{k=1\\k\neq i}}^{n}(x_k - x_i)}\right),$$

and using the recurrence relations

$$\begin{bmatrix} \tilde{F}_k(x) \\ \tilde{r}_k(x) \end{bmatrix} = \begin{bmatrix} 0 \\ b_n \end{bmatrix}$$

$$\begin{bmatrix} \tilde{F}_k(x) \\ \tilde{r}_k(x) \end{bmatrix} = \begin{bmatrix} \alpha_{n-k+1}^T & \frac{1}{\delta_{n-k+1}}\gamma_{n-k+1}^T \\ \delta_{n-k}\beta_{n-k+1}^T & \delta_{n-k}x + \frac{\theta_{n-k+1}}{\delta_{n-k+1}} \end{bmatrix} \begin{bmatrix} \tilde{F}_k(x) \\ \tilde{r}_k(x) \end{bmatrix} + \begin{bmatrix} 0 \\ \delta_{n-k}b_{n-k} \end{bmatrix}$$

where the perturbations b_k are defined by

$$H(x) = \prod_{k=1}^{n}(x - x_k) = b_0 r_0(x) + \cdots + b_n r_n(x),$$

to evaluate the generalized Horner polynomials $\tilde{R} = \{\tilde{r}_0(x),\, \tilde{r}_1(x),\, \ldots,\, \tilde{r}_{n-1}(x)\}$ (of Procedure 2.1) at each node x_k.

The proof is a straightforward application of Proposition 7.1 and the reversal of the (H,m)–quasiseparable filter structure pictured in Figure 6, and an algebraic proof can be found in [4] (and for the $(H,1)$–quasiseparable case in [3]).

Similarly, the proof of the following corollary is seen by using Proposition 7.1 and the reversal of the (H, m)–semiseparable filter structure, which is pictured in Figure 7. An algebraic proof of this in the $(H, 1)$–semiseparable case appeared in [3].

Corollary 7.3. *Let R be a system of (H, m)–semiseparable polynomials given in terms of recurrence relation coefficients, and $H(x)$ a monic polynomial with exactly n distinct roots. Then the (H, m)–semiseparable Vandermonde matrix $V_R(x)$ whose nodes $\{x_k\}$ are the zeros of $H(x)$ can be inverted as*

$$V_R(x)^{-1} = \begin{bmatrix} \tilde{r}_{n-1}(x_1) & \tilde{r}_{n-1}(x_2) & \cdots & \tilde{r}_{n-1}(x_n) \\ \vdots & \vdots & & \vdots \\ \tilde{r}_1(x_1) & \tilde{r}_1(x_2) & \cdots & \tilde{r}_1(x_n) \\ \tilde{r}_0(x_1) & \tilde{r}_0(x_2) & \cdots & \tilde{r}_0(x_n) \end{bmatrix} \cdot D,$$

with

$$D = diag\,(H'(x_i)) = diag\left(\frac{1}{\Pi_{\substack{k=1\\k\neq i}}^{n}(x_k - x_i)}\right),$$

and using the recurrence relations

$$\begin{bmatrix} \tilde{G}_0(x) \\ \tilde{r}_0(x) \end{bmatrix} = \begin{bmatrix} -b_n\beta_n^T \\ b_n \end{bmatrix}$$

$$\begin{bmatrix} \tilde{G}_k(x) \\ \tilde{r}_k(x) \end{bmatrix} = \begin{bmatrix} \alpha_{n-k}^T & \gamma_{n-k}^T \\ \delta_{n-k}\beta_{n-k}^T & \delta_{n-k} \end{bmatrix} \begin{bmatrix} \tilde{G}_k(x) \\ \left(x + \frac{\theta_{n-k+1}}{\delta_{n-k+1}}\right)\tilde{r}_k(x) + b_{n-k} \end{bmatrix}$$

where the perturbations b_k are defined by

$$H(x) = \prod_{k=1}^{n}(x - x_k) = b_0 r_0(x) + \cdots + b_n r_n(x),$$

to evaluate the generalized Horner polynomials $\tilde{R} = \{\tilde{r}_0(x), \tilde{r}_1(x), \ldots, \tilde{r}_{n-1}(x)\}$ (of Procedure 2.1) at each node x_k.

8. Conclusions

In this paper, we use the language of signal flow graphs, typically used in applications, to answer purely mathematical questions regarding the class of quasiseparable matrices. Two new filter classes were introduced, and the connection between Horner and generalized Horner polynomials and reversing the flow of a signal flow graph were exploited to solve three mathematical questions.

References

[1] G. Ammar, W. Gragg, and L. Reichel. An analogue for the Szegö polynomials of the Clenshaw algorithm. *J. Computational Appl. Math.*, 46:211–216, 1993.

[2] T. Bella, Y. Eidelman, I. Gohberg, and V. Olshevsky. Classifications of three–term and two–term recurrence relations via subclasses of quasiseparable matrices. *submitted to SIAM Journal of Matrix Analysis (SIMAX)*, 2007.

[3] T. Bella, Y. Eidelman, I. Gohberg, V. Olshevsky, and E. Tyrtyshnikov. Fast Traub–like inversion algorithm for Hessenberg order one quasiseparable Vandermonde matrices. *submitted to Journal of Complexity*, 2007.

[4] T. Bella, Y. Eidelman, I. Gohberg, V. Olshevsky, E. Tyrtyshnikov, and P. Zhlobich. A Traub–like algorithm for Hessenberg–quasiseparable–Vandermonde matrices of arbitrary order. *accepted*, 2008.

[5] T. Bella, V. Olshevsky, and P. Zhlobich. Classifications of recurrence relations via subclasses of (H, m)–quasiseparable matrices. *submitted*, 2008.

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

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

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

[9] D. Calvetti and L. Reichel. Fast inversion of Vandermonde–like matrices involving orthogonal polynomials. *BIT*, 1993.

[10] C. Clenshaw. A note on summation of Chebyshev series. *M.T.A.C.*, 9:118–120, 1955.

[11] Y. Eidelman and I. Gohberg. Linear complexity inversion algorithms for a class of structured matrices. *Integral Equations and Operator Theory*, 35:28–52, 1999.

[12] I. Gohberg and V. Olshevsky. Fast inversion of Chebyshev–Vandermonde matrices. *Numerische Mathematik*, 67:71 – 92, 1994.

[13] 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.

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

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

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

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

[18] V. Olshevsky. Eigenvector computation for almost unitary Hessenberg matrices and inversion of Szegö-Vandermonde matrices via discrete transmission lines. *Linear Algebra and Its Applications*, 285:37–67, 1998.

[19] V. Olshevsky. Associated polynomials, unitary Hessenberg matrices and fast generalized Parker-Traub and Bjorck-Pereyra algorithms for Szegö-Vandermonde matrices.

Structured Matrices: Recent Developments in Theory and Computation (D.Bini, E. Tyrtyshnikov, P. Yalamov., Eds.), pages 67–78, 2001.

[20] J. Traub. Associated polynomials and uniform methods for the solution of linear problems. *SIAM Review*, 8:277 – 301, 1966.

T.Bella* and V.Olshevsky† and P.Zhlobich†