Linear Algebra and Its Applications, vol. 203–204, pp. 155–188, Jul. 1994.

Structured Matrices and Unconstrained Rational Interpolation Problems *

Tibor Boros, Ali H. Sayed, and Thomas Kailath

Abstract

We describe a fast recursive algorithm for the solution of an unconstrained rational interpolation problem by exploiting the displacement structure concept. We use the interpolation data to implicitly define a convenient non-Hermitian structured matrix, and then apply a computationally efficient procedure for its triangular factorization. This leads to a transmission line interpretation that makes evident the interpolation properties. We further discuss connections with the Lagrange interpolating polynomial as well as questions regarding the minimality and the admissible degrees of complexity of the solutions.

1 Introduction

Interpolation problems lend themselves for interesting applications in many fields that include, but are not limited to, partial realization, model reduction, Padé approximation, Hankel-norm approximation, H^∞–control, etc. What we shall define here as unconstrained interpolation problems have a very long history, associated with many classical results of Lagrange, Hermite, Prony, Padé, and other famous names. In recent years, several authors have approached these problems from a system-theoretical point of view, where the main idea is to find a two-input two-output linear system, also known as a *generating system*, whose global transfer matrix can be used to obtain a linear fractional parametrization of the family of rational interpolants. In particular, many studies of this type have been made over the last few years by Antoulas, Ball, Gohberg, and their colleagues (see, esp. [1, 2] and the references therein).

The basic (unconstrained) interpolation problem that we treat in this paper can be stated as follows:

Problem 1.1 (Unconstrained Interpolation) *Given the array of complex pairs*

$$(\alpha_i, \beta_i), \quad i \in \{0, 1, \ldots, n-1\}, \quad i \neq j \Longrightarrow \alpha_i \neq \alpha_j, \quad |\beta_i| < \infty,$$

find all rational interpolants $y(z) = n(z)/d(z)$ such that

*This work was supported in part by the Air Force Office of Scientific Research, Air Force Systems Command under Contract AFOSR91-0060 and by the Army Research Office under contract DAAL03-89-K-0109.

- $\beta_i = y(\alpha_i)$, *and*
- $y(z)$ *is irreducible (i.e.,* $n(z)$ *and* $d(z)$ *are coprime).*

■

Antoulas and Anderson solved this problem by exploiting the rank properties of a Loewner matrix constructed from the interpolation data [3, 4]. Ball, Gohberg and Rodman gave a linear fractional parametrization of all interpolants in terms of a rational transfer matrix $\mathbf{\Theta}(z)$ by using the so-called residual interpolation method (see [2] and [5, Chapter 16]).

In many applications the interpolants are also required to satisfy certain minimality constraints. The complexity of a rational solution $y(z)$ is measured in terms of its McMillan degree, which is defined by

$$\delta\{y(z)\} = \max\{\deg n(z), \deg d(z)\} \quad .$$

This motivates the consideration of the following two additional questions.

Problem 1.2 (Minimal Interpolation)

- *How to determine the admissible degrees of complexity of the rational interpolants?*
- *How to determine the minimal degree of complexity and the minimal interpolants?*

■

Apparently, the minimal interpolation problem was first solved by Antoulas and Anderson in the Loewner matrix setting [3]. By using a different approach, Antoulas and Willems [1] showed how to give a linear fractional parametrization in terms of a column reduced polynomial matrix obtained via the coprime factorization of a suitable rational matrix function. This approach was extended to the matrix case and combined with the residual interpolation method in [6]. Further developments, at a more abstract level, were initiated by invoking the so-called behavioral framework for linear systems [7].

In this paper, we shall pursue an approach related to the computationally-oriented solution put forward for rational analytic interpolation problems in [8, 9, 10]. The key point is an *efficient recursive algorithm* for the factorization of matrices possessing displacement structure. This fast algorithm can be naturally associated with a cascade of first-order sections. Such cascades always have certain interpolation properties because of the fact that linear systems have "transmission zeros": certain inputs at certain frequencies yield zero outputs. More specifically, each section of the cascade can be characterized by a rational transfer matrix $\Theta_i(z)$ say, that has a left zero-direction vector g_i at a frequency f_i, viz.,

$$g_i\,\Theta_i(f_i) = \begin{pmatrix} g_{i0} & g_{i1} \end{pmatrix} \begin{pmatrix} \Theta_{i,11}(f_i) & \Theta_{i,12}(f_i) \\ \Theta_{i,21}(f_i) & \Theta_{i,22}(f_i) \end{pmatrix} = \mathbf{0} \ ,$$

which makes evident (with the proper partitioning of the row vector g_i and the matrix function $\Theta_i(z)$) the following (local) interpolation property: $g_{i0}\,\Theta_{i,12}(f_i)\,\Theta_{i,22}^{-1}(f_i) = -g_{i1}$. Hence, one way of solving an interpolation problem is to show how to construct an appropriate cascade so that the local interpolation properties of the elementary sections combine in such a way that the cascade yields a solution to the global interpolation problem.

The matrix R that we factorize is specified by the so-called displacement equation

$$R - F\,R\,A^* = G\,J\,B^* \quad ,$$

where F and G are directly constructed from the interpolation data, and the arrays A and B are chosen so as to simplify the recursion and to impose further constraints on the interpolating functions. For example, a particular choice leads to a cascade that implements the Lagrange solution. Another choice leads to a cascade whose transfer matrix is column reduced, which is useful in answering the minimality questions. More generally, each particular choice $\{A, B\}$ offers a different way of characterizing the family of all solutions.

As we shall explain in Section 6 below, our work is closely related to the earlier results of Ball, Gohberg, and Rodman (see, esp. Theorem 5.4.1–2 in [5]). From this perspective, our main contribution is in providing a recursive version of their global formulas. Although the following discussion can be extended to the multiple point case as well as the vector case (to be discussed elsewhere), we restrict ourselves here, for brevity and clarity of presentation, to the distinct point and scalar case.

2 Triangular Factorization and the Generalized Schur Algorithm

We start by reviewing some basic results concerning the triangular factorization of non-Hermitian matrices. We then present the array form of the generalized Schur algorithm (see, e.g., [9, 10, 11, 12] for more details on the subject).

Consider a *strongly regular* (i.e., all leading minors are non-zero) non-Hermitian $n \times n$ matrix R. The assumption of strong regularity guarantees the existence of a triangular factorization of form $R = L\,D^{-1}\,U^*$ where

$$L = \begin{pmatrix} \tilde{l}_0 & \tilde{l}_1 & \dots & \tilde{l}_{n-1} \end{pmatrix}, \quad \text{and} \quad U = \begin{pmatrix} \tilde{u}_0 & \tilde{u}_1 & \dots & \tilde{u}_{n-1} \end{pmatrix}$$

are $n \times n$ lower triangular matrices, i.e.,

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

and $D = \text{diag}\{d_0, d_1, \dots, d_{n-1}\}$, where $d_i \neq 0$. The columns of L and U as well as the diagonal elements of D can be computed by the well-known *Gauss/Schur reduction procedure.*

Algorithm 2.1 (Schur Reduction Procedure) *Start the recursion with* $R_0 = R$ *and repeat for all* $i \in \{0, 1, \dots, n-1\}$*:*

$$l_i = \begin{pmatrix} r_{00}^{(i)} \\ r_{10}^{(i)} \\ \vdots \\ r_{n-i-1,0}^{(i)} \end{pmatrix}, \qquad u_i^* = \begin{pmatrix} r_{00}^{(i)} & r_{01}^{(i)} & \dots & r_{0,n-i-1}^{(i)} \end{pmatrix}, \qquad d_i = r_{00}^{(i)},$$

$$R_i - l_i\, d_i^{-1}\, u_i^* = \begin{pmatrix} 0 & 0 & \ldots & 0 \\ 0 & & & \\ \vdots & & R_{i+1} & \\ 0 & & & \end{pmatrix},$$

where the elements of R_i are denoted by $R_i = \left[r_{jk}^{(i)}\right]_{j,k=0}^{n-i-1}$. ■

The matrix R_{i+1} is called the *Schur complement* of $r_{00}^{(i)}$ in R_i. The Schur reduction procedure usually requires $\mathcal{O}(n^3)$ operations. If the matrix R is "structured" then fast algorithms can be devised in order to reduce the computational burden.

We say that R is structured if it satisfies a displacement equation of the form

$$R - F\,R\,A^* = G\,J\,B^* \ , \tag{1}$$

where

- F and A are appropriate lower triangular matrices, whose diagonal entries will be denoted by $\{f_i\}$ and $\{a_i\}$, respectively:

$$F = \begin{pmatrix} f_0 & & & \\ x & f_1 & & \\ \vdots & \vdots & \ddots & \\ x & x & \ldots & f_{n-1} \end{pmatrix}, \qquad A = \begin{pmatrix} a_0 & & & \\ x & a_1 & & \\ \vdots & \vdots & \ddots & \\ x & x & \ldots & a_{n-1} \end{pmatrix}.$$

- G and B are $n \times r$ ($r \ll n$ usually) so-called generator matrices.
- J is a signature matrix of the form

$$J = I_p \oplus -I_q = \begin{pmatrix} I_p & 0 \\ 0 & -I_q \end{pmatrix}, \qquad p + q = r.$$

We further assume that the matrix equation (1) has a unique solution, i.e., that the following conditions are satisfied

$$1 - f_i\, a_j^* \neq 0 \qquad \forall\, i, j \in \{0, 1, \ldots, n-1\}.$$

The generators G and B are not unique; the column dimension r of the minimal generators is called the *displacement rank* of R with respect to F and A.

One of the major facts of the displacement structure theory is that the successive Schur complements of a structured matrix inherit its displacement structure (see [10]). More precisely, if R_i is the Schur complement of the leading $i \times i$ block in R, then R_i satisfies a displacement equation similar to (1), viz.,

$$R_i - F_i\, R_i\, A_i^* = G_i\, J\, B_i^*, \qquad i \in \{0, 1, \ldots, n-1\},$$

where F_i and A_i are obtained by deleting the first i columns and rows of F and A, respectively. The generator matrices G_i and B_i can be obtained recursively by the so-called non-Hermitian generalized Schur algorithm [10, 11].

Algorithm 2.2 (Generalized Schur Algorithm) *The generators of the successive Schur complements satisfy the following recursion: start with $G_0 = G$, $B_0 = B$, and repeat*

$$\begin{aligned}
\begin{pmatrix} \mathbf{0} \\ G_{i+1} \end{pmatrix} &= \Phi_i\, G_i\, \Theta_i \begin{pmatrix} \mathbf{0}_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{r-j-1} \end{pmatrix} + G_i\, \Theta_i \begin{pmatrix} I_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{r-j-1} \end{pmatrix}, \\
\begin{pmatrix} \mathbf{0} \\ B_{i+1} \end{pmatrix} &= \Psi_i\, B_i\, \Gamma_i \begin{pmatrix} \mathbf{0}_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{r-j-1} \end{pmatrix} + B_i\, \Gamma_i \begin{pmatrix} I_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{r-j-1} \end{pmatrix},
\end{aligned} \tag{2}$$

where Φ_i and Ψ_i are defined by

$$\Phi_i = (F_i - f_i\, I)\,(I - a_i^*\, F_i)^{-1}\,, \qquad \Psi_i = (A_i - a_i\, I)\,(I - f_i^*\, A_i)^{-1}\,, \tag{3}$$

and Θ_i and Γ_i are constant matrices chosen so as to satisfy $\Theta_i\, J\, \Gamma_i^* = J$, and such that the generators G_i and B_i are reduced to proper form. This means that Θ_i and Γ_i transform the first row of G_i (denoted by g_i) and the first row of B_i (denoted by b_i) to the forms

$$g_i\, \Theta_i = \begin{pmatrix} 0 & \dots & 0 & x & 0 & \dots & 0 \end{pmatrix}, \qquad b_i\, \Gamma_i = \begin{pmatrix} 0 & \dots & 0 & x & 0 & \dots & 0 \end{pmatrix}, \tag{4}$$

with a single non-zero entry in the same column position, say the j-th position. ■

The rotation matrices Θ_i and Γ_i can be implemented in a variety of ways, e.g., by using suitable variations of elementary Householder projections, Givens rotations, or hyperbolic transformations. Moreover, the generator recursion (2) has the following simple array interpretation, as depicted in Figure 1:

1. Choose Θ_i and Γ_i that reduce G_i and B_i to proper form.
2. Multiply G_i by Θ_i and B_i by Γ_i.
3. Multiply the j-th column of $G_i\, \Theta_i$ by Φ_i and the j-th column of $B_i\, \Gamma_i$ by Ψ_i, and keep all other columns unaltered.
4. These steps result in G_{i+1} and B_{i+1}.

If the generators G_i and B_i are known then the triangular factors of R can be obtained as

$$\left.\begin{aligned}
l_i &= (I_i - a_i^*\, F_i)^{-1} G_i\, J\, b_i^* \\
u_i &= (I_i - f_i^*\, A_i)^{-1} B_i\, J\, g_i^* \\
d_i &= \tfrac{g_i\, J\, b_i^*}{1 - f_i\, a_i^*}
\end{aligned}\right\} \quad i \in \{0, 1, \dots, n-1\}.$$

This method requires only $\mathcal{O}(n^2)$ operations.

Remark: Notice that the matrix R is strongly regular if and only if $r_{00}^{(i)} = d_i = \frac{g_i\, J\, b_i^*}{1 - f_i a_i^*} \neq 0$, $i \in \{0, 1, \dots, n-1\}$.

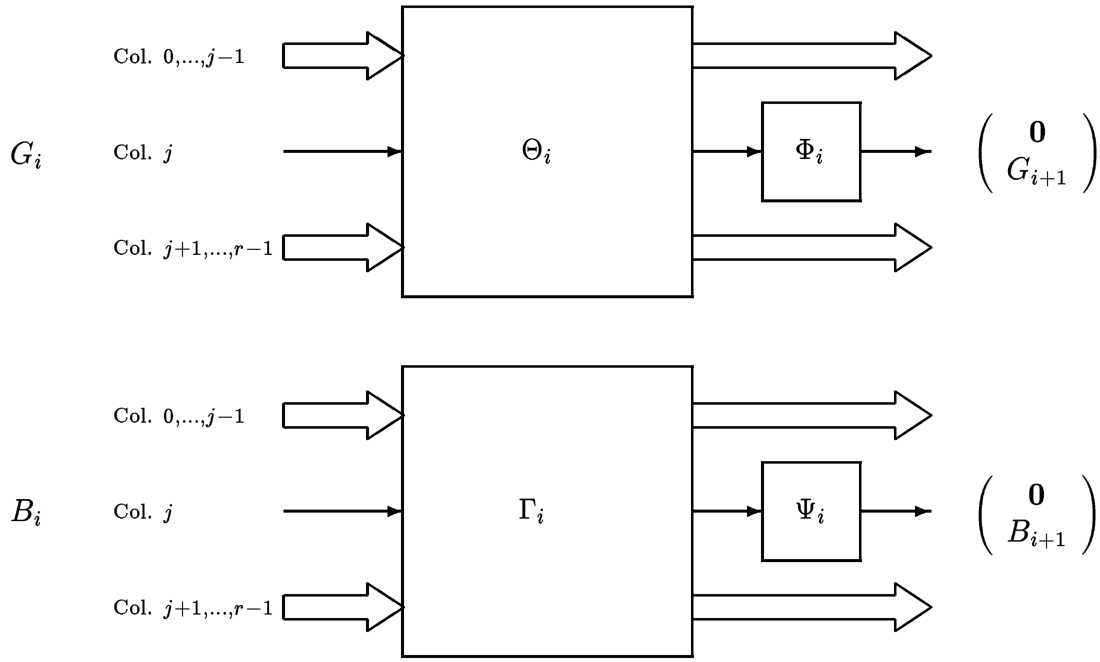


Figure 1: Array form of the generalized Schur algorithm.

3 Cascade Systems: Blocking and Interpolation Properties

As further discussed in [8, 9, 10, 11], with each step of the generalized Schur algorithm we can associate two first-order discrete-time r-input r-output systems with transfer functions

$$\Theta_i(z) = \Theta_i \begin{pmatrix} I_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathcal{B}_{\Theta,i}(z) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{r-j-1} \end{pmatrix} ,$$

$$\Gamma_i(w) = \Gamma_i \begin{pmatrix} I_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathcal{B}_{\Gamma,i}(w) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{r-j-1} \end{pmatrix} ,$$

where

$$\mathcal{B}_{\Theta,i}(z) = \frac{z - f_i}{1 - a_i^* z} , \qquad \mathcal{B}_{\Gamma,i}(w) = \frac{w - a_i}{1 - f_i^* w} . \tag{5}$$

These transfer functions are jointly J-unitary in the following sense: $\Theta_i(z)\, J\, \Gamma_i^*(w) = J$ on $z\, w^* = 1$. Also notice that the diagonal elements of F and A determine the zero and pole structures of $\Theta_i(z)$, respectively.

After performing n steps of the generator recursion, we obtain two discrete-time cascades with transfer functions

$$\begin{aligned} \boldsymbol{\Theta}(z) &= \Theta_1(z)\, \Theta_2(z) \ldots \Theta_n(z), \\ \boldsymbol{\Gamma}(w) &= \Gamma_1(w)\, \Gamma_2(w) \ldots \Gamma_n(w), \end{aligned} \tag{6}$$

which inherit the generalized J-unitary property:

$$\mathbf{\Theta}(z)\, J\, \mathbf{\Gamma}^*(w) = J \qquad \text{on} \qquad z\, w^* = 1. \tag{7}$$

Our purpose is to show how to choose suitable matrices F, A, G, and B, to define a displacement structure as in (1), and how to recursively construct the associated cascades $\mathbf{\Theta}(z)$ and $\mathbf{\Gamma}(w)$ in order to parametrize all solutions of the interpolation problem.

For the moment we note two simple but important facts about the cascades. First, observe that the determinants of the matrix functions $\mathbf{\Theta}(z)$ and $\mathbf{\Gamma}(w)$ can be expressed as

$$\det \mathbf{\Theta}(z) \sim \frac{\prod_{i=1}^{n}(z - f_i)}{\prod_{i=1}^{n}(1 - a_i^*\, z)} \ , \qquad \det \mathbf{\Gamma}(w) \sim \frac{\prod_{i=1}^{n}(w - a_i)}{\prod_{i=1}^{n}(1 - f_i^*\, w)} \ , \tag{8}$$

where '$\sim$' denotes proportionality. This readily leads to the following result, to be used in later sections.

Lemma 3.1 *The transfer matrix $\mathbf{\Theta}(z)$ will be a polynomial matrix if A is strictly lower triangular, i.e., $a_i = 0,\ i \in \{0, 1, \ldots, n-1\}$. Under this condition we have*

$$\mathcal{B}_{\Theta,i}(z) = z - f_i \ , \qquad \mathcal{B}_{\Gamma,i}(w) = \frac{w}{1 - f_i^*\, w} \ ,$$

$$\det \mathbf{\Theta}(z) \sim \prod_{i=1}^{n}(z - f_i) \ , \qquad \det \mathbf{\Gamma}(w) \sim \frac{w^n}{\prod_{i=1}^{n}(1 - f_i^*\, w)} \ .$$

■

The second important fact about the cascades is that the first order sections $\Theta_i(z)$ and $\Gamma_i(w)$ exhibit obvious blocking properties [8, 9, 10].

Lemma 3.2 (Blocking Property) *Each first-order section $\Theta_i(z)$ (resp. $\Gamma_i(w)$) has a transmission zero at f_i (resp. a_i) along the zero direction g_i (resp. b_i). That is,*

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

Proof: It follows from (4) that

$$\begin{aligned} g_i\, \Theta_i(f_i) &= g_i\, \Theta_i \begin{pmatrix} I_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathcal{B}_{\Theta,i}(f_i) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{r-j-1} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{0} & x & \mathbf{0} \end{pmatrix} \begin{pmatrix} I_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{r-j-1} \end{pmatrix} = \mathbf{0}, \end{aligned}$$

Similarly, $b_i\, \Gamma_i(a_i) = \mathbf{0}$. ■

We now verify that these local blocking properties are inherited by the feed-forward cascades $\mathbf{\Theta}(z)$ and $\mathbf{\Gamma}(w)$. For this purpose, we focus on $\mathbf{\Theta}(z)$ and consider the special case of a diagonal matrix F, which will be of interest in this paper; the same result can be extended to more general forms of F and will be discussed elsewhere:

$$F = \begin{pmatrix} f_0 & & & \\ & f_1 & & \\ & & \ddots & \\ & & & f_{n-1} \end{pmatrix}, \qquad A = \begin{pmatrix} a_0 & & & \\ x & a_1 & & \\ \vdots & \vdots & \ddots & \\ x & x & \dots & a_{n-1} \end{pmatrix}.$$

The next lemma states that the rows of the generator matrix G are zero directions of the transfer matrix $\mathbf{\Theta}(z)$ at the 'frequencies' f_i. This statement can be justified by the following reasoning. The first row of G annihilates $\mathbf{\Theta}(z)$ at $z = f_0$ due to the local blocking property of $\Theta_0(z)$. When the second row of G enters the cascade, we get the first row of G_1 at the input of $\Theta_1(z)$ (due to the generator recursion), which then annihilates $\mathbf{\Theta}(z)$ at $z = f_1$, and so on. This argument is made precise in the proof of the following result.

Lemma 3.3 (Interpolation Property) *The cascade $\mathbf{\Theta}(z)$ has a global interpolation property at f_i, viz.,*

$$e_i \, G \, \mathbf{\Theta}(f_i) = \mathbf{0},$$

where $e_i = \begin{pmatrix} \mathbf{0}_i & 1 & \mathbf{0}_{n-i-1} \end{pmatrix}$.

Proof: The matrices Φ_i are clearly diagonal since F is diagonal. It then readily follows from the generator recursion (2) that the first row of G_i is given by

$$\begin{aligned} g_i &= e_0 \, G_i \\ &= \frac{f_i - f_{i-1}}{1 - a_{i-1}^* f_i} \, e_1 \, G_{i-1} \, \Theta_{i-1} \begin{pmatrix} \mathbf{0}_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{r-j-1} \end{pmatrix} + e_1 \, G_{i-1} \, \Theta_{i-1} \begin{pmatrix} I_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{r-j-1} \end{pmatrix} \\ &= e_1 \, G_{i-1} \, \Theta_{i-1} \begin{pmatrix} I_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathcal{B}_{\Theta,i}(f_i) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{r-j-1} \end{pmatrix} = e_1 \, G_{i-1} \, \Theta_{i-1}(f_i) \, . \end{aligned}$$

By induction we obtain,

$$g_i = e_1 \, G_{i-1} \, \Theta_{i-1}(f_i) = e_2 \, G_{i-2} \, \Theta_{i-2}(f_i) \, \Theta_{i-1}(f_i) = \ldots = e_i \, G \, \Theta_0(f_i) \, \Theta_1(f_i) \, \ldots \, \Theta_{i-1}(f_i).$$

But $g_i \, \Theta_i(f_i) = \mathbf{0}$ due to the local blocking property of the i-th section $\Theta_i(z)$. Hence, $e_i \, G \, \Theta_0(f_i) \, \Theta_1(f_i) \, \ldots \, \Theta_{i-1}(f_i) \, \Theta_i(f_i) = \mathbf{0}$, and by (6), $e_i \, G \, \mathbf{\Theta}_i(f_i) = \mathbf{0}$. ■

4 The Unconstrained Interpolation Problem

We now apply the previous theory to the unconstrained rational interpolation problem that was introduced in Section 1. For this purpose we consider the following special choices of F, G, and J that are constructed directly from the interpolation data,

$$F = \begin{pmatrix} \alpha_0 & & & \\ & \alpha_1 & & \\ & & \ddots & \\ & & & \alpha_{n-1} \end{pmatrix}, \quad G = \begin{pmatrix} 1 & -\beta_0 \\ 1 & -\beta_1 \\ \vdots & \vdots \\ 1 & -\beta_{n-1} \end{pmatrix}, \quad J = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{9}$$

For the moment we choose B arbitrarily and A as a *strictly* lower triangular matrix,

$$A = \begin{pmatrix} 0 & & & & & \\ x & 0 & & & & \\ x & x & 0 & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ x & x & x & \dots & 0 & \\ x & x & x & \dots & x & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} x & x \\ x & x \\ \vdots & \vdots \\ x & x \end{pmatrix}. \tag{10}$$

The strict lower triangularity condition on A assures that $\mathbf{\Theta}(z)$ will be a *polynomial* rather than rational matrix, which will be relevant to our later analysis. More specifically (see Proposition 4.1 below), a polynomial $\mathbf{\Theta}(z)$ will allow us to express all rational interpolants as $y(z) = n(z)/d(z)$, where $n(z)$ and $d(z)$ are coprime polynomials. In Section 7 (see Proposition 7.1) we give a particular choice for A and B that ensures the strong regularity of the matrix R in the non-Hermitian displacement equation $R - F\,R\,A^* = G\,J\,B^*$. Other choices for A and B are also possible and may be useful when additional factors and constraints are important (see e.g., [13]).

Let us now specialize the generalized Schur algorithm to the arrays given in (9) and (10). Here we assume that A and B are chosen so that the generator recursion is strongly regular. We denote the elements of the first rows of the generators G_i and B_i by

$$g_i = \begin{pmatrix} g_{i0} & g_{i1} \end{pmatrix}, \qquad b_i = \begin{pmatrix} b_{i0} & b_{i1} \end{pmatrix}.$$

It follows from the strong regularity assumption that $g_i\,J\,b_i \neq 0$, i.e., either $g_{i0} \neq 0$ and $b_{i0} \neq 0$ or $g_{i1} \neq 0$ and $b_{i1} \neq 0$. Therefore, at each stage of the recursion there are two possible ways of transforming g_i and b_i to proper form.

Case A If $g_{i0} \neq 0$ and $b_{i0} \neq 0$ then we can define the *reflection coefficients* k_i, l_i as

$$k_i = g_{i1}/g_{i0}, \qquad l_i = b_{i1}/b_{i0},$$

and the transformation matrices Θ_i and Γ_i can be chosen as

$$\Theta_i = \begin{pmatrix} 1 & -k_i \\ -l_i^* & 1 \end{pmatrix}, \qquad \Gamma_i = \frac{1}{1 - k_i^*\,l_i} \begin{pmatrix} 1 & -l_i \\ -k_i^* & 1 \end{pmatrix}.$$

These transformations pivot with the entries g_{i0} and b_{i0} and introduce new zero entries at the $(0,1)$ positions of G_i and B_i:

$$g_i\,\Theta_i = \begin{pmatrix} x & 0 \end{pmatrix}, \qquad b_i\Gamma_i = \begin{pmatrix} x' & 0 \end{pmatrix}.$$

This gives rise to the feed-forward lattice line sections that are depicted in Figure 2.

Case B If $g_{i1} \neq 0$ and $b_{i1} \neq 0$ then the reflection coefficients and the rotation matrices can be chosen as

$$k_i = g_{i0}/g_{i1}, \qquad l_i = b_{i0}/b_{i1},$$

and

$$\Theta_i = \begin{pmatrix} 1 & -l_i^* \\ -k_i & 1 \end{pmatrix}, \qquad \Gamma_i = \frac{1}{1 - k_i\,l_i^*} \begin{pmatrix} 1 & -k_i^* \\ -l_i & 1 \end{pmatrix}.$$

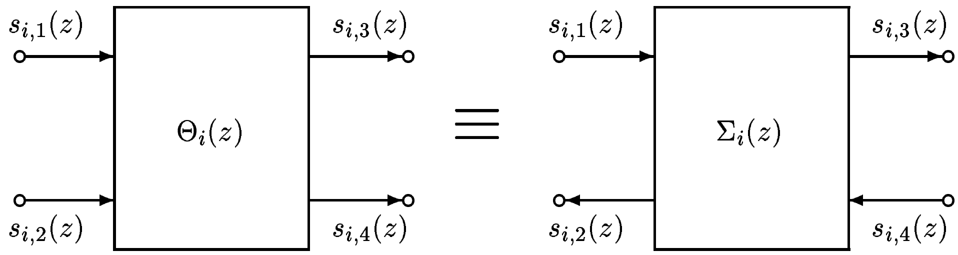


Figure 2: Feed-forward lattice sections in the $r = 2$ case when we pivot with the left-most entries g_{i0} and b_{i0}.

In this case g_{i1} and b_{i1} are the pivoting elements, and the transformations introduce zero entries at the $(0, 0)$ positions of G_i and B_i:

$$g_i\,\Theta_i = \begin{pmatrix} 0 & x \end{pmatrix}, \qquad b_i\Gamma_i = \begin{pmatrix} 0 & x' \end{pmatrix}.$$

The corresponding feed-forward lattice sections are shown in Figure 3.

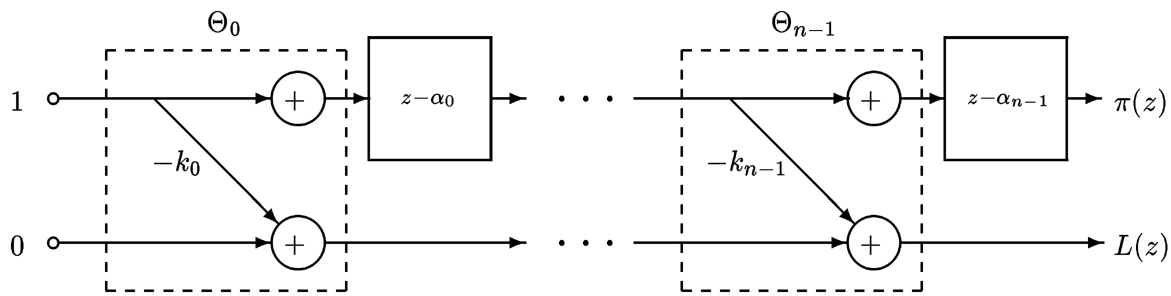


Figure 3: Feed-forward lattice sections in the $r = 2$ case when we pivot with the right-most entries g_{i1} and b_{i1}.

After n steps of the generator recursion we obtain a cascade $\mathbf{\Theta}(z)$, which is composed of first-order sections $\Theta_i(z)$ of either of the types considered in Cases A and B above. The transfer matrix $\mathbf{\Theta}(z)$ can be partitioned as follows:

$$\mathbf{\Theta}(z) = \begin{pmatrix} \Theta_{11}(z) & \Theta_{12}(z) \\ \Theta_{21}(z) & \Theta_{22}(z) \end{pmatrix}.$$

Now, we can re-express the interpolation property (9) in the form

$$\begin{aligned} \Theta_{11}(\alpha_i) - \beta_i\,\Theta_{21}(\alpha_i) = 0, \\ \Theta_{12}(\alpha_i) - \beta_i\,\Theta_{22}(\alpha_i) = 0, \end{aligned} \tag{11}$$

which implies (along with the distinct points constraint, $\alpha_i \neq \alpha_j$) that $\Theta_{21}(z)$ and $\Theta_{22}(z)$ do not have common zeros. Otherwise, the determinant of $\mathbf{\Theta}(z)$ would have multiple roots at these points, which contradicts (8).

We can further verify the well known fact that the rational interpolants are given by a linear fractional transformation (see e.g., [1], [5, Chapter 16], and [6]).

Proposition 4.1 (Linear Fractional Parametrization) *The family of irreducible rational interpolants can be parametrized as*

$$y(z) = \frac{p(z)\,\Theta_{11}(z) + q(z)\,\Theta_{12}(z)}{p(z)\,\Theta_{21}(z) + q(z)\,\Theta_{22}(z)} \;, \tag{12}$$

where $p(z)$ *and* $q(z)$ *are coprime polynomials such that*

$$p(\alpha_i)\,\Theta_{21}(\alpha_i) + q(\alpha_i)\,\Theta_{22}(\alpha_i) \neq 0, \quad i \in \{0, 1, \ldots, n-1\}. \tag{13}$$

Proof: By substituting (9) into (11) we obtain

$$\Theta_{11}(\alpha_i) - \beta_i\,\Theta_{21}(\alpha_i) = 0,\ \Theta_{12}(\alpha_i) - \beta_i\,\Theta_{22}(\alpha_i) = 0,\ i \in \{0, 1, \ldots, n-1\}.$$

Adding $p(\alpha_i)$ times the first equation to $q(\alpha_i)$ times the second equation yields

$$p(\alpha_i)\,\Theta_{11}(\alpha_i) + q(\alpha_i)\,\Theta_{12}(\alpha_i) = \beta_i\,\Big(\ p(\alpha_i)\,\Theta_{21}(\alpha_i) + q(\alpha_i)\,\Theta_{22}(\alpha_i)\ \Big).$$

Thus condition (13) justifies that $y(z)$ in (12) is an admissible interpolant. The coprimeness of $p(z)$ and $q(z)$ together with condition (13) imply the irreducibility of $y(z)$.

Conversely, if $y(z) = n(z)/d(z)$ is an irreducible interpolant then choose $p(z)$ and $q(z)$ as follows:

$$\begin{aligned}
\begin{pmatrix} p(z) \\ q(z) \end{pmatrix} &= \begin{pmatrix} \Theta_{11}(z) & \Theta_{12}(z) \\ \Theta_{21}(z) & \Theta_{22}(z) \end{pmatrix}^{-1} \begin{pmatrix} n(z) \\ d(z) \end{pmatrix} = \\
&= \frac{1}{\det \boldsymbol{\Theta}(z)} \begin{pmatrix} \Theta_{22}(z) & -\Theta_{12}(z) \\ -\Theta_{21}(z) & \Theta_{11}(z) \end{pmatrix} \begin{pmatrix} n(z) \\ d(z) \end{pmatrix} = \\
&\sim \frac{1}{\prod_{i=0}^{n-1}(z - \alpha_i)} \begin{pmatrix} n(z)\,\Theta_{22}(z) - d(z)\,\Theta_{12}(z) \\ -n(z)\,\Theta_{21}(z) + d(z)\,\Theta_{11}(z) \end{pmatrix}.
\end{aligned}$$

By the interpolation property (9) we have

$$\begin{aligned}
n(\alpha_i)\,\Theta_{22}(\alpha_i) - d(\alpha_i)\,\Theta_{12}(\alpha_i) = d(\alpha_i)\,\Big(\ \beta_i\,\Theta_{22}(\alpha_i) - \Theta_{12}(\alpha_i)\ \Big) &= 0, \\
-n(\alpha_i)\,\Theta_{21}(\alpha_i) + d(\alpha_i)\,\Theta_{11}(\alpha_i) = d(\alpha_i)\,\Big(\ -\beta_i\,\Theta_{21}(\alpha_i) + \Theta_{11}(\alpha_i)\ \Big) &= 0,
\end{aligned}$$

which shows that $p(z)$ and $q(z)$ are polynomials. The coprimeness of $p(z)$ and $q(z)$ follows from the irreducibility of $y(z)$. Since β_i is finite, $d(\alpha_i) \neq 0$, and condition (13) holds. ■

Our main result is that we have an efficient $\mathcal{O}(n^2)$ algorithm for obtaining the basic cascade $\boldsymbol{\Theta}(z)$.

Algorithm 4.1 (Solution of the Rational Interpolation Problem)

Step 1 *Form the arrays* F *and* G *from the input data as shown in (9).*

Step 2 *Choose a strictly lower triangular $n \times n$ matrix A and a suitable $n \times 2$ matrix B so that the solution R of the non-Hermitian displacement equation (1) is strongly regular.*

Step 3 *Carry out the generalized Schur algorithm with F, A, G, B, and J. Determine the cascade $\mathbf{\Theta}(z)$.*

Step 4 *Use formula (12) to parametrize all solutions of the interpolation problem.*

■

The existence of coprime polynomials $p(z)$ and $q(z)$ that satisfy (13) is guaranteed by the coprimeness of $\Theta_{21}(z)$ and $\Theta_{22}(z)$. Note that if $\Theta_{12}(z)$ and $\Theta_{22}(z)$ do not share common zeros at α_i, $i \in \{0, 1, \ldots, n-1\}$, then we can choose $p(z) = 0$ and $q(z) = 1$, which leads to

$$y(z) = \frac{\Theta_{12}(z)}{\Theta_{22}(z)}\,.$$

Similarly, if $\Theta_{11}(z)$ and $\Theta_{21}(z)$ do not share common zeros at α_i, $i \in \{0, 1, \ldots, n-1\}$, then we can choose $p(z) = 1$ and $q(z) = 0$, which leads to

$$y(z) = \frac{\Theta_{11}(z)}{\Theta_{21}(z)}\,.$$

We should remark that the generalized Schur algorithm yields two cascades $\mathbf{\Theta}(z)$ and $\mathbf{\Gamma}(w)$. In this section we focused on the interpolation properties of $\mathbf{\Theta}(z)$ only. It can be shown, however, that $\mathbf{\Gamma}(w)$ also satisfies interpolation conditions. The implications of this duality will be discussed elsewhere.

Furthermore, additional constraints on the interpolants, such as analyticity and norm constraints, can be introduced by choosing $A = F$, $B = G$, and considering the appropriate Hermitian structured matrix that satisfies a displacement equations of the form

$$R - F\,R\,F^* = G\,J\,G^*.$$

Under these circumstances the generator recursion leads to a J-lossless cascade, viz., $\mathbf{\Theta}(z)$ is analytic in $|z| \leq 1$ and

$$\mathbf{\Theta}(z)\,J\,\mathbf{\Theta}^*(z) = J \quad \text{on} \quad |z| = 1.$$

For more details on analytic interpolation and on the Hermite-Fejér problem along these lines see [8, 9, 10].

5 Physical Interpretation

In this section we describe a useful interpretation of the general formula (12) by examining more closely the signal flow properties of the elementary sections. As we mentioned before, the generating system $\mathbf{\Theta}(z)$ is obtained as the cascade connection of feed-forward lattice sections of the types shown in Figures 2 and 3. Each section can be thought of as a first-order linear system that communicates with its environment via four signals denoted by $s_{i,1}$, $s_{i,2}$, $s_{i,3}$, and $s_{i,4}$ (see Figure 4). If we consider $s_{i,1}$ and $s_{i,2}$ as input signals, and $s_{i,3}$ and

$s_{i,4}$ as output signals, then the i-th section can be described in the frequency domain by the operator $\Theta_i(z)$ as

$$\begin{pmatrix} s_{i,3}(z) & s_{i,4}(z) \end{pmatrix} = \begin{pmatrix} s_{i,1}(z) & s_{i,2}(z) \end{pmatrix} \begin{pmatrix} \Theta_{i,11}(z) & \Theta_{i,12}(z) \\ \Theta_{i,21}(z) & \Theta_{i,22}(z) \end{pmatrix} . \tag{14}$$

However, if we consider $s_{i,1}$ and $s_{i,4}$ as input signals, and $s_{i,3}$ and $s_{i,2}$ as output signals, then the i-th section can be described by the operator $\Sigma_i(z)$ as

$$\begin{pmatrix} s_{i,3}(z) & s_{i,2}(z) \end{pmatrix} = \begin{pmatrix} s_{i,1}(z) & s_{i,4}(z) \end{pmatrix} \begin{pmatrix} \Sigma_{i,11}(z) & \Sigma_{i,12}(z) \\ \Sigma_{i,21}(z) & \Sigma_{i,22}(z) \end{pmatrix} . \tag{15}$$

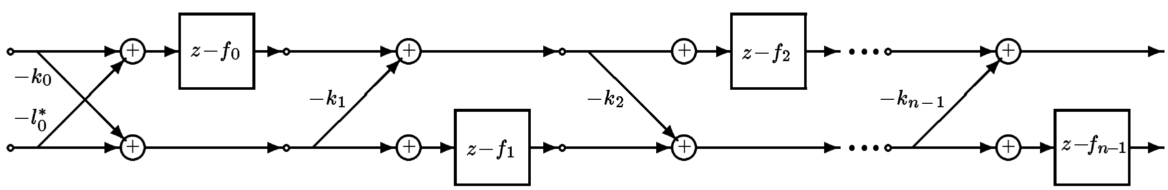


Figure 4: Equivalent descriptions of a first-order linear system.

The transfer functions $\Theta_i(z)$ and $\Sigma_i(z)$ are related as follows:

$$\Sigma_i(z) = \begin{pmatrix} \Theta_{i,11}(z) - \Theta_{i,12}(z)\,\Theta_{i,22}^{-1}(z)\,\Theta_{i,21}(z) & -\Theta_{i,12}(z)\,\Theta_{i,22}^{-1}(z) \\ \Theta_{i,22}^{-1}(z)\,\Theta_{i,21}(z) & \Theta_{i,22}^{-1}(z) \end{pmatrix}$$

(notice that $\Theta_{i,22}^{-1}(z)$ always exists). The two descriptions are equivalent in the sense that if the signals satisfy (14) then they also satisfy (15), and vice-versa.

In circuit theory the matrix $\Sigma_i(z)$ is known as the scattering matrix relating the incident signals $s_{i,1}(z)$ and $s_{i,4}(z)$ to the reflected signals $s_{i,3}(z)$ and $s_{i,2}(z)$. In this framework the $\Theta_i(z)$ is known as the chain scattering matrix. In the composite system that is obtained by interconnecting the elementary sections, the chain scattering matrices multiply in the usual way:

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

The matrix $\mathbf{\Sigma}(z)$ that corresponds to $\mathbf{\Theta}(z)$ has to be written as

$$\mathbf{\Sigma}(z) = \Sigma_0(z) \star \Sigma_1(z) \star \,\ldots\, \star \Sigma_{n-1}(z)\,,$$

where '$\star$' denotes the so-called Redheffer star-product defined by

$$\begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix} \star \begin{pmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{pmatrix} \stackrel{\text{def}}{=}$$

$$\begin{pmatrix} X_{11}(I - Y_{12}\,X_{21})^{-1}Y_{11} & X_{12} + X_{11}\,Y_{12}(I - X_{21}\,Y_{12})^{-1}X_{22} \\ Y_{21} + Y_{22}\,X_{21}(I - Y_{12}\,X_{21})^{-1}Y_{11} & Y_{22}(I - X_{21}\,Y_{12})^{-1}X_{22} \end{pmatrix} .$$

The two global frequency-domain descriptions are connected via the relation

$$\mathbf{\Sigma}(z) = \begin{pmatrix} \Theta_{11}(z) - \Theta_{12}(z)\,\Theta_{22}^{-1}(z)\,\Theta_{21}(z) & -\Theta_{12}(z)\,\Theta_{22}^{-1}(z) \\ \Theta_{22}^{-1}(z)\,\Theta_{21}(z) & \Theta_{22}^{-1}(z) \end{pmatrix}.$$

The significance of the $\mathbf{\Sigma}(z)$ description is that we now have the analog of a physical transmission-line with signals flowing in both directions as shown in Figure 5 (for more details in this direction see [14]).

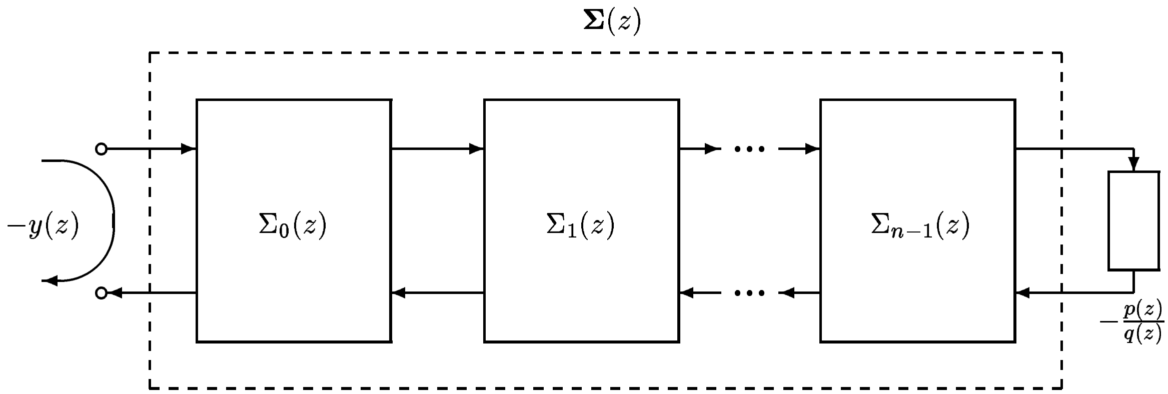


Figure 5: Scattering representation of the interpolating function

Now, we are ready to give a physical interpretation for the linear fractional parametrization formula derived in Proposition 4.1. It is easy to check that the interpolating function $y(z)$ can be obtained as the negative transfer function from the top left input to the bottom left output of the scattering cascade shown in Figure 5 when an appropriate load $-p(z)/q(z)$ is attached to the right-hand-side (assuming $q(z) \neq 0$). When a unit-amplitude impulse is applied at $t = 0$ to the input of the system then the z-transform of the output signal will take on the value β_i at the frequency $z = \alpha_i$, independently of the load at the right-hand-side, as long as the parameters $p(z)$ and $q(z)$ satisfy (13).

In this framework the interpolating function $y(z)$ is constructed recursively by adding a new first-order section to the scattering cascade for each interpolation constraint. The additional section does not influence the interpolation properties of the previous sections; at the same time its reflection coefficients are determined so that the transfer function of the composite system will satisfy the new interpolation constraint.

The transmission-line interpretation can be pursued further to give a physical interpretation of the generalized Schur algorithm as a natural method for solving the *inverse scattering problem* for the line (see the discussions in [14] and also [15]).

6 State-Space Descriptions

In order to make a connection to the work of Ball et al. [2, 5], in this section we present an explicit expression for the transfer matrix $\mathbf{\Theta}(z)$ in terms of the arrays F, G, A, B, J, and the solution R of the displacement equation (1). A similar formula was obtained earlier by Kimura [16] for the analogous half-plane analytic interpolation problem.

In [10, 11] it was shown that the discrete-time linear systems $\Theta_i(z)$ and $\Gamma_i(w)$ can be represented in state-space as

$$\begin{aligned} \Theta_i(z) &= J\, s_i^*\, J + J\, b_i^* \left(z^{-1} - a_i^* \right)^{-1} c_i^*\, J\,, \\ \Gamma_i(w) &= J\, k_i^*\, J + J\, g_i^* \left(w^{-1} - f_i^* \right)^{-1} h_i^*\, J\,, \end{aligned}$$

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

$$\begin{pmatrix} f_i & g_i \\ h_i & k_i \end{pmatrix} \begin{pmatrix} d_i & \mathbf{0} \\ \mathbf{0} & J \end{pmatrix} \begin{pmatrix} a_i & b_i \\ c_i & s_i \end{pmatrix}^* = \begin{pmatrix} d_i & \mathbf{0} \\ \mathbf{0} & J \end{pmatrix} .$$

By combining the time-domain description of the first-order sections we can obtain state-space realizations for the feed-forward cascades in the form

$$\begin{aligned} \mathbf{\Theta}(z) &= J\, S^*\, J + J\, B^* \left(z^{-1} I - A^* \right)^{-1} C^*\, J, \\ \mathbf{\Gamma}(w) &= J\, K^*\, J + J\, G^* \left(w^{-1} I - F^* \right)^{-1} H^*\, J, \end{aligned} \tag{16}$$

where the matrices C, S, H and K satisfy the global embedding relation

$$\begin{pmatrix} F & G \\ H & K \end{pmatrix} \begin{pmatrix} R & \mathbf{0} \\ \mathbf{0} & J \end{pmatrix} \begin{pmatrix} A & B \\ C & S \end{pmatrix}^* = \begin{pmatrix} R & \mathbf{0} \\ \mathbf{0} & J \end{pmatrix} .$$

In fact, it is possible to express C, S, H and K explicitly as

$$\begin{aligned} C &= \Gamma^{-1} J\, G^* \left(I - \tau\, F^* \right)^{-1} R^{-*} \left(\tau\, I - A \right), \\ S &= \Gamma^{-1} \left[I - J\, G^* \left(I - \tau\, F^* \right)^{-1} R^{-*}\, B \right], \\ H &= \Theta^{-1} J\, B^* \left(I - \mu\, A^* \right)^{-1} R^{-1} \left(\mu\, I - F \right), \\ K &= \Theta^{-1} \left[I - J\, B^* \left(I - \mu\, A^* \right)^{-1} R^{-1}\, G \right], \end{aligned}$$

where τ and μ are constants that satisfy $\tau\, \mu^* = 1$, and Γ and Θ are $r \times r$ matrices that satisfy $\Theta\, J\, \Gamma^* = J$. By substituting these expressions into (16) we obtain [10, 11]

$$\begin{aligned} \mathbf{\Theta}(z) &= \left[I - \left(z^{-1} - \tau^* \right) J\, B^* \left(z^{-1} I - A^* \right)^{-1} R^{-1} \left(I - \tau^* F \right)^{-1} G \right] \Theta, \\ \mathbf{\Gamma}(z) &= \left[I - \left(z^{-1} - \mu^* \right) J\, G^* \left(z^{-1} I - F^* \right)^{-1} R^{-*} \left(I - \mu^* A \right)^{-1} B \right] \Gamma. \end{aligned} \tag{17}$$

These formulas generalize certain expressions found in [5, Theorem 4.5.2] on specifying a rational function via left and right null-pole triples. From the above state-space realizations it follows that $\{A^*, -JB^*\}$ is a right pole pair of $\mathbf{\Theta}(z)$, while $\{F, G\}$ is a left pole pair of $J\,\mathbf{\Gamma}^*(z)\,J$. By the global J-unitary property (7) we conclude that $J\,\mathbf{\Gamma}^*(z)\,J = \mathbf{\Theta}^{-1}(z)$, therefore $\{F, G\}$ is also a left null pair of $\mathbf{\Theta}(z)$.

Hence, another way to solve the unconstrained interpolation problem is to compute a generating system $\mathbf{\Theta}(z)$ by using the explicit expression (17). Then the family of rational interpolants can be parametrized in terms of $\mathbf{\Theta}(z)$ as shown in (12). Note that the global method requires the explicit computation of R^{-1}, while the recursive method avoids this step. On the other hand, the global method requires only the invertibility of R, in contrast with the recursive method, which requires strong regularity.

7 Connection to the Lagrange Polynomial

The unconstrained interpolation problem that we studied in the previous section can also be solved by expressing all interpolants in terms of the so-called Lagrange interpolating polynomial as was done, for example, by Antoulas and Anderson in [1, 4, 6]. As a matter of interest, in this section we shall show how to construct this solution recursively by using the generalized Schur algorithm. Recall that we are essentially free to choose A and B so as to guarantee the strong regularity of R. Each such choice would lead to a cascade $\mathbf{\Theta}(z)$ that parametrizes all solutions of the problem as shown in Proposition 4.1. We shall give here a particular choice for A and B that will lead us to the Lagrange solution.

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

$$L_i(\alpha_j) = \begin{cases} 1 & \text{if} \quad j = i \\ 0 & \text{if} \quad j \neq i \end{cases} \qquad i \in \{0, 1, \ldots, n-1\}.$$

That is, each basis polynomial $L_i(z)$ assumes the value 1 at α_i and the value zero at the other points $\alpha_j,\ j \neq i$. Now a polynomial solution to Problem 1.1 can be obtained as a linear combination of $L_0(z), L_1(z), \ldots L_{n-1}(z)$ with coefficients $\beta_0, \beta_1, \ldots \beta_{n-1}$:

$$L(z) = \sum_{i=0}^{n-1} \beta_i \, L_i(z). \tag{18}$$

$L(z)$ is called the *Lagrange interpolating polynomial* and constitutes the unique solution of Problem 1.1 in the space of polynomials of order at most $n-1$.

We can write down an explicit expression for $L_i(z)$ as follows. $L_i(z)$ has zeros at $\alpha_0, \alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_{n-1}$. Therefore,

$$L_i(z) = \frac{\prod_{j \neq i}(z - \alpha_j)}{\prod_{j \neq i}(\alpha_i - \alpha_j)} = \frac{1}{\pi'(\alpha_i)} \, \frac{\pi(z)}{(z - \alpha_i)},$$

where

$$\pi(z) = \prod_{j=0}^{n-1} (z - \alpha_j).$$

Thus, we obtain the celebrated formula

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

It is also known that all rational solutions to Problem 1.1 can be parametrized in terms of $L(z)$.

Lemma 7.1 *$y(z)$ is a rational interpolant if, and only if*

$$y(z) = L(z) + \pi(z)\,\frac{p(z)}{q(z)}\ , \tag{19}$$

for some coprime polynomials $p(z)$ and $q(z)$ such that $q(\alpha_i) \neq 0,\ \ i \in \{0, 1, \ldots, n-1\}$.

Proof: The 'if' part of the statement is obvious since the second term in (19) vanishes at $\alpha_i,\ i \in \{0, 1, \ldots, n-1\}$.

The 'only if' part can be proved as follows. Let us suppose that $y(z) = n(z)/d(z)$ is an irreducible rational interpolant, i.e., $y(\alpha_i) = \beta_i$. Choose

$$\begin{aligned} q(z) &= d(z), \\ p(z) &= \Big(\ y(z) - L(z)\ \Big)\,\frac{d(z)}{\pi(z)} = \frac{n(z) - L(z)\,d(z)}{\pi(z)}\ . \end{aligned}$$

Now, $p(z)$ is a polynomial since the numerator has roots at α_i, i.e.,

$$n(\alpha_i) - L(\alpha_i)\,d(\alpha_i) = n(\alpha_i) - \beta_i\,d(\alpha_i) = 0\ , \quad i \in \{0, 1, \ldots, n-1\}.$$

Moreover, $y(z)$ is irreducible and hence, $p(z)$ and $q(z)$ are coprime. Finally $q(\alpha_i) \neq 0$ because β_i is finite. ■

We now show how to construct the Lagrange interpolating polynomial by using the generalized Schur algorithm.

Proposition 7.1 *Assume that we choose F and G as in (9). If the generator B and the lower triangular matrix A are chosen such that*

$$A = \begin{pmatrix} 0 & 0 & 0 & \ldots & 0 & 0 \\ 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & \ldots & 1 & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{pmatrix},$$

then the matrix R in

$$R - F\,R\,A^* = G\,J\,B^*$$

is strongly regular and the generalized Schur algorithm yields

$$\mathbf{\Theta}(z) = \begin{pmatrix} \pi(z) & L(z) \\ 0 & 1 \end{pmatrix}. \tag{20}$$

Proof: $\mathbf{\Theta}(z)$ is clearly a polynomial matrix since A is strictly lower triangular. From (3) we obtain that

$$\Phi_i \ = \ F_i - \alpha_i\,I = \begin{pmatrix} 0 & & & \\ & (\alpha_{i+1} - \alpha_i) & & \\ & & \ddots & \\ & & & (\alpha_{n-1} - \alpha_i) \end{pmatrix},$$

and Ψ_i is a strictly lower triangular Toeplitz matrix with elements $\{0, 1, \alpha_i, \alpha_i^2, \ldots, \alpha_i^{n-i-2}\}$ in the first column, viz.,

$$\Psi_i \;=\; A_i\,(I-\alpha_i^*\,A_i)^{-1} = \begin{pmatrix} 0 & & & & & & \\ 1 & 0 & & & & & \\ \alpha_i & 1 & \ddots & & & & \\ \alpha_i^2 & \alpha_i & \ddots & 0 & & & \\ \vdots & \alpha_i^2 & \ddots & 1 & 0 & & \\ & & \ddots & \alpha_i & 1 & 0 & \\ \alpha_i^{n-i-2} & & \ldots & \alpha_i^2 & \alpha_i & 1 & 0 \end{pmatrix}.$$

The first step of the generalized Schur algorithm gives: $k_0 = -\beta_0,\;\; l_0 = 0$,

$$\Theta_0(z) = \begin{pmatrix} 1 & -k_0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} z-\alpha_0 & 0 \\ 0 & 1 \end{pmatrix}, \qquad \Gamma_0(w) = \begin{pmatrix} 1 & 0 \\ -k_0^* & 1 \end{pmatrix} \begin{pmatrix} \frac{w}{1-\alpha_0^* w} & 0 \\ 0 & 1 \end{pmatrix},$$

$$G_1 = \begin{pmatrix} \alpha_1-\alpha_0 & \beta_0-\beta_1 \\ \alpha_2-\alpha_0 & \beta_0-\beta_2 \\ \vdots & \vdots \\ \alpha_{n-1}-\alpha_0 & \beta_0-\beta_{n-1} \end{pmatrix}, \qquad B_1 = \begin{pmatrix} 1 & 0 \\ \alpha_0 & 0 \\ \vdots & \vdots \\ \alpha_0^{n-2} & 0 \end{pmatrix}.$$

By induction we easily verify that at each stage of the recursion we get $l_i = 0$ and hence,

$$\Theta_i(z) = \begin{pmatrix} 1 & -k_i \\ 0 & 1 \end{pmatrix} \begin{pmatrix} z-\alpha_i & 0 \\ 0 & 1 \end{pmatrix}, \qquad \Gamma_i(w) = \begin{pmatrix} 1 & 0 \\ -k_i^* & 1 \end{pmatrix} \begin{pmatrix} \frac{w}{1-\alpha_i^* w} & 0 \\ 0 & 1 \end{pmatrix},$$

$$G_i = \begin{pmatrix} \mu_0^{(i)} & x \\ \mu_1^{(i)} & x \\ \vdots & \vdots \\ \mu_{n-i-1}^{(i)} & x \end{pmatrix}, \qquad B_i = \begin{pmatrix} 1 & 0 \\ x & 0 \\ \vdots & \vdots \\ x & 0 \end{pmatrix},$$

where

$$\mu_j^{(i)} = \prod_{k=0}^{i-1} (\alpha_{j+i}-\alpha_k), \qquad j \in \{0, 1, \ldots, n-i-1\}, \qquad i \in \{1, 2, \ldots, n-1\}.$$

Due to the distinct point constraint we have $\mu_j^{(i)} \neq 0$ for all i, j. Hence, the generator recursion is strongly regular. Notice that in this case we have no freedom in deciding the positions of the pivoting elements, since at each step we get $b_{i1} = 0$ and hence, the algorithm guides us to pivot with the left-most entries b_{i0} and g_{i0}. According to (6), the transfer matrix $\mathbf{\Theta}(z)$ can be written as

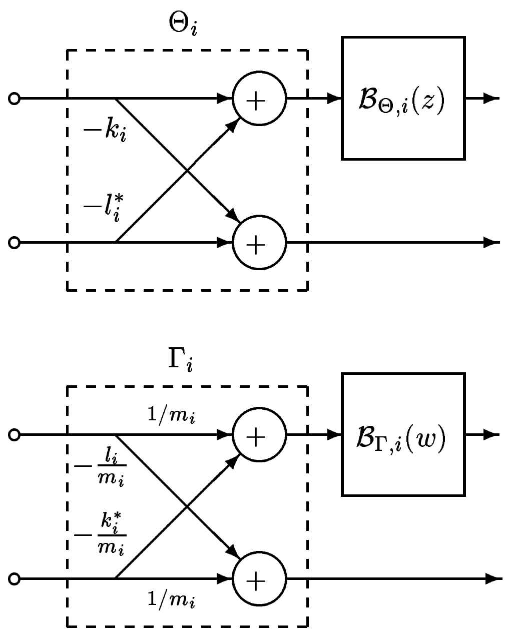
$$\begin{aligned} \mathbf{\Theta}(z) \;&=\; \begin{pmatrix} 1 & -k_0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} z-\alpha_0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -k_1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} z-\alpha_1 & 0 \\ 0 & 1 \end{pmatrix} \cdots \\ &\cdots \begin{pmatrix} 1 & -k_{n-1} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} z-\alpha_{n-1} & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \pi(z) & X(z) \\ 0 & 1 \end{pmatrix}, \end{aligned}$$

where $X(z)$ is a polynomial of degree at most $n-1$. By using the interpolation property (9) we obtain

$$\begin{pmatrix} 1 & -\beta_i \end{pmatrix} \begin{pmatrix} \pi(\alpha_i) & X(\alpha_i) \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \end{pmatrix}, \qquad i \in \{0, 1, \ldots, n-1\},$$

which implies $X(\alpha_i) = \beta_i, \ \forall i$. We readily conclude that $X(z) = L(z)$ since $L(z)$ is unique. ■

Observe that substituting $\mathbf{\Theta}(z)$ into the linear fractional parametrization (12) we get (19).



$$m_i = 1 - l_i\, k_i^*$$

Figure 6: The Lagrange lattice

The cascade system $\mathbf{\Theta}(z)$ associated with the special choices in Proposition 7.1 is depicted in Figure 6. We see that the transfer function from the top left input to the bottom right output is

$$L(z) = -k_0 - k_1(z-\alpha_0) - k_2(z-\alpha_0)(z-\alpha_1) - \ldots - k_{n-1}\prod_{i=0}^{n-2}(z-\alpha_i). \tag{21}$$

This relationship can be interpreted as follows. Let us define the so-called Newton basis $\{P_0(z), P_1(z), \ldots, P_{n-1}(z)\}$,

$$\begin{aligned} P_0(z) &= 1, \\ P_1(z) &= z-\alpha_0, \\ P_2(z) &= (z-\alpha_0)(z-\alpha_1), \end{aligned} \tag{22}$$

$$\vdots$$

$$P_{n-1}(z) \quad = \quad \prod_{i=0}^{n-2}(z-\alpha_i).$$

It then follows from (21) that the Lagrange interpolating polynomial can be represented in this basis as

$$L(z) = \sum_{i=0}^{n-1} -k_i\, P_i(z).$$

In other words, the reflection coefficients $k_0, k_1, \ldots, k_{n-1}$ are the negative coordinates of $L(z)$ in the Newton basis. At the beginning of this section we showed that $\beta_0, \beta_1, \ldots, \beta_{n-1}$ are the coordinates of $L(z)$ in the Lagrange basis. We thus conclude that the generalized Schur algorithm performs a coordinate transformation between the Lagrange basis and the Newton basis.

We further remark that other choices of A and B are also possible, as long as A is strictly lower triangular, the second column of B is zero, and the strong regularity condition is satisfied. In Proposition 7.1 we showed a special convenient (sparse) form for A and B. In the multiple point case, when F is in a general Jordan form, a similar reasoning leads to the Hermite interpolating polynomial, though we shall not show this here.

8 Column Reduced Transfer Matrices

We now show that, under special conditions, the generator recursion gives rise to a column reduced transfer matrix $\mathbf{\Theta}(z)$. In a series of papers [1, 3, 4, 6] Antoulas et al. pointed out that no statements can be made about the degree of complexity of the rational interpolants

$$\begin{pmatrix} n(z) \\ d(z) \end{pmatrix} = \begin{pmatrix} \Theta_{11}(z) & \Theta_{12}(z) \\ \Theta_{21}(z) & \Theta_{22}(z) \end{pmatrix} \begin{pmatrix} p(z) \\ q(z) \end{pmatrix}$$

unless the polynomial matrix $\mathbf{\Theta}(z)$ is column reduced, i.e, the degree of its determinant is equal to the sum of its column indices. If $\mathbf{\Theta}(z)$ is column reduced with column indices κ_1 and $\kappa_2 \geq \kappa_1$, then we can apply the so-called *predictable degree property* [17, p. 387], [18] to claim that the McMillan degree of the interpolants is

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

The main result in [6] establishes that if $\mathbf{\Theta}(z)$ is column reduced then the minimal solution of the unconstrained rational interpolation problem can be obtained by either

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

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

$$y_{\min}(z) = \frac{p(z)\,\Theta_{11}(z) + \Theta_{12}(z)}{p(z)\,\Theta_{21}(z) + \Theta_{22}(z)}, \qquad \begin{array}{c} \deg p(z) \leq \kappa_2 - \kappa_1, \\ p(\alpha_i)\,\Theta_{21}(\alpha_i) + \Theta_{22}(\alpha_i) \neq 0, \end{array}$$

when $\Theta_{11}(z)$ and $\Theta_{21}(z)$ share some common roots. In the first case, there is a minimal solution with complexity κ_1, while in the second case there exists a family of minimal interpolants whose complexities are equal to κ_2. In both cases, there exist infinitely many interpolants with complexity $\delta\{y(z)\} > \kappa_2$.

Notice, however, that (apart from the case when $\deg\{L(z)\} = 0$) the solution $\mathbf{\Theta}(z)$ that was obtained in the previous section,

$$\mathbf{\Theta}(z) = \begin{pmatrix} \pi(z) & L(z) \\ 0 & 1 \end{pmatrix} \tag{23}$$

is not column reduced. In [6] it was suggested to apply Euclid's algorithm to transform (23) to a column reduced form.

We now show that it is possible to directly obtain a column reduced transfer matrix $\mathbf{\Theta}(z)$ via the generalized Schur algorithm, by proper choices of the matrices A and B.

Proposition 8.1 (Column Reduced Transfer Matrix) *Assume that we choose F and G as in (9). If the diagonal and the first subdiagonal elements of A are all zero, i.e., if*

$$A = \begin{pmatrix} 0 & & & & & & & \\ 0 & 0 & & & & & & \\ x & 0 & 0 & & & & & \\ x & x & 0 & 0 & & & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & & & \\ x & x & x & x & \dots & 0 & & \\ x & x & x & x & \dots & 0 & 0 & \\ x & x & x & x & \dots & x & 0 & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} x & x \\ x & x \\ x & x \\ \vdots & \vdots \\ x & x \end{pmatrix},$$

then the Schur algorithm yields a column reduced polynomial matrix $\mathbf{\Theta}(z)$, provided that the solution R of the displacement equation

$$R - F\,R\,A^* = G\,J\,B^*$$

is strongly regular.

Proof: Notice that the first two rows of $\Psi_i = A_i\,(I - \alpha_i^*\,A_i)^{-1}$ will be identically zero. Therefore, at the i-th stage of the Schur recursion, we introduce an additional zero element in the first row of the generator B_{i+1}. This means that in the next stage $l_{i+1} = 0$, and we must switch the position of the pivoting elements. Let us follow this reasoning on a step-by-step basis.

Step 0: This is a completely general step. Without loss of generality, let us pivot with g_{00} and b_{00}.

$$\begin{aligned} \Theta_0(z) &= \begin{pmatrix} 1 & -k_0 \\ -l_0^* & 1 \end{pmatrix} \begin{pmatrix} z - \alpha_0 & 0 \\ 0 & 1 \end{pmatrix}, \\ \Gamma_0(w) &= \frac{1}{1 - k_0^*\,l_0} \begin{pmatrix} 1 & -l_0 \\ -k_0^* & 1 \end{pmatrix} \begin{pmatrix} \frac{w}{1-\alpha_0^*\,w} & 0 \\ 0 & 1 \end{pmatrix}, \end{aligned}$$

$$G_1 = \begin{pmatrix} x & x \\ x & x \\ \vdots & \vdots \\ x & x \end{pmatrix}, \qquad B_1 = \begin{pmatrix} 0 & x \\ x & x \\ \vdots & \vdots \\ x & x \end{pmatrix}.$$

Step 1: Now $l_1 = 0$ and we must pivot with b_{11} and g_{11}. If the strong regularity condition holds then $g_{11} \neq 0$ and $b_{11} \neq 0$. In this case,

$$\Theta_1(z) = \begin{pmatrix} 1 & 0 \\ -k_1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & z - \alpha_1 \end{pmatrix}, \qquad \Gamma_1(w) = \begin{pmatrix} 1 & -k_1^* \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \frac{w}{1-\alpha_1^* w} \end{pmatrix},$$

$$G_2 = \begin{pmatrix} x & x \\ x & x \\ \vdots & \vdots \\ x & x \end{pmatrix}, \qquad B_2 = \begin{pmatrix} x & 0 \\ x & x \\ \vdots & \vdots \\ x & x \end{pmatrix}.$$

Step 2: Now $l_2 = 0$ and we must pivot with b_{20} and g_{20}. If the strong regularity holds then $b_{20} \neq 0$ and $g_{20} \neq 0$. Moreover,

$$\Theta_2(z) = \begin{pmatrix} 1 & -k_2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} z - \alpha_2 & 0 \\ 0 & 1 \end{pmatrix}, \qquad \Gamma_2(w) = \begin{pmatrix} 1 & 0 \\ -k_2^* & 1 \end{pmatrix} \begin{pmatrix} \frac{w}{1-\alpha_2^* w} & 0 \\ 0 & 1 \end{pmatrix},$$

$$G_3 = \begin{pmatrix} x & x \\ x & x \\ \vdots & \vdots \\ x & x \end{pmatrix}, \qquad B_3 = \begin{pmatrix} 0 & x \\ x & x \\ \vdots & \vdots \\ x & x \end{pmatrix}.$$

By induction we obtain that $l_i = 0$, $\forall i \geq 1$. If i is odd then

$$\Theta_i(z) = \begin{pmatrix} 1 & 0 \\ -k_i & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & z - f_i \end{pmatrix}, \quad \Gamma_i(w) = \begin{pmatrix} 1 & -k_i^* \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \frac{w}{1-f_i^* w} \end{pmatrix}.$$

If i is even then

$$\Theta_i(z) = \begin{pmatrix} 1 & -k_i \\ 0 & 1 \end{pmatrix} \begin{pmatrix} z - f_i & 0 \\ 0 & 1 \end{pmatrix}, \quad \Gamma_i(w) = \begin{pmatrix} 1 & 0 \\ -k_i^* & 1 \end{pmatrix} \begin{pmatrix} \frac{w}{1-f_i^* w} & 0 \\ 0 & 1 \end{pmatrix}.$$

The transfer function $\mathbf{\Theta}(z)$ can be expressed as

$$\begin{aligned} \mathbf{\Theta}(z) \quad = \quad & \begin{pmatrix} 1 & -k_0 \\ -l_0^* & 1 \end{pmatrix} \begin{pmatrix} z - f_0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -k_1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & z - f_1 \end{pmatrix} \cdots \\ & \cdots \begin{pmatrix} 1 & -k_i \\ 0 & 1 \end{pmatrix} \begin{pmatrix} z - f_i & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -k_{i+1} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & z - f_{i+1} \end{pmatrix} \cdots \end{aligned}$$

By taking the product of the consecutive blocks it is straightforward to check that $\mathbf{\Theta}(z)$ is column reduced. ■

Observe that at each stage of the recursion we obtain $l_i = 0$ as we did in the case of the Lagrange lattice, but now we cannot keep the pivoting elements at the left-hand-side of the generator. We are guided by the algorithm to switch the positions of the pivoting elements back and forth leading to the Schur lattice depicted in Figure 7. It can be shown that this cascade-synthesis procedure is related to Euclid's algorithm.

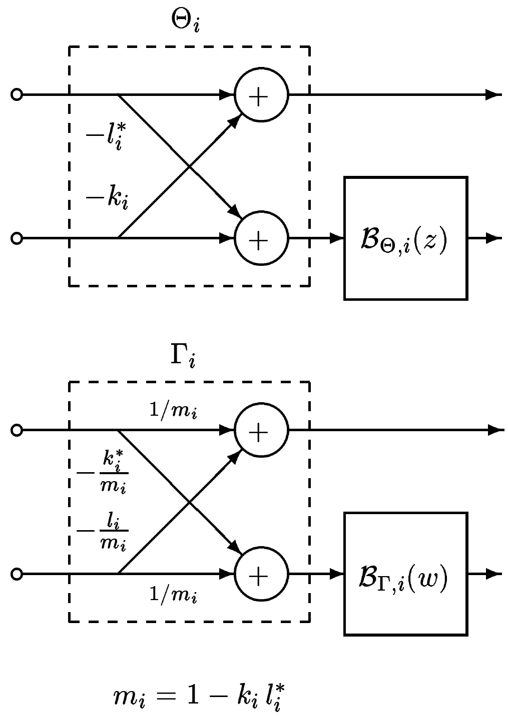


Figure 7: Feed-forward lattice with a column reduced transfer matrix.

9 Concluding Remarks

We have reexamined the much studied scalar unconstrained rational interpolation problem, and described a new, computationally efficient, recursive solution using the notion of displacement structure and the associated generalized Schur algorithm for fast matrix factorization. We showed how to recursively construct a convenient feed-forward cascade that parametrizes all solutions. Special cascades that implement the Lagrange solution, as well as a column reduced solution, were also presented; these allow us to obtain minimal degree interpolants. We derived global state-space formulas for the generating system, and provided a physical interpretation for the interpolating functions.

We may remark that here we limited ourselves to the case of structured matrices with displacement rank $r = 2$. However, the generalized Schur algorithm applies also to higher order displacement ranks and allows us to solve tangential interpolation problems, where the interpolation solution is a matrix function. We also restricted ourselves to the distinct point case, even though F can have a more general Jordan form. These extensions will be discussed elsewhere. In this paper we considered interpolation problems on the unit circle. Similar discussions can be carried out for problems on the line (as done, e.g., in [5]).

References

[1] A. C. Antoulas and J. C. Willems. Minimal rational interpolation and Prony's method. In A. Bensoussan and J. L. Lions, editors, *Analysis and Optimization of Systems*, volume 144 of *Lecture Notes in Control and Information Sciences*, pages 297–306. Springer-Verlag, 1990.

[2] J. A. Ball, I. Gohberg, and L. Rodman. Two-sided Lagrange-Sylvester interpolation problems for rational matrix functions. In *Proceedings of Symposia in Pure Mathematics, Part I*, volume 51, pages 17–83. American Mathematical Society, 1990.

[3] A. C. Antoulas and B. D. O. Anderson. On the scalar rational interpolation problem. *IMA Journal of Mathematical Control & Information*, 3:61–88, 1986.

[4] A. C. Antoulas and B. D. O. Anderson. State-space and polynomial approaches to rational interpolation. In M. A. Kaashoek, J. H. van Schuppen, and A. C. M. Ran, editors, *Realization and Modelling in System Theory, Proceedings of the International Symposium MTNS-89*, volume 1, pages 73–82, Boston, 1990. Birkhäuser.

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

[6] A. C. Antoulas, J. A. Ball, J. Kang, and J. C. Willems. On the solution of the minimal rational interpolation problem. *Linear Algebra and its Applications*, 137/138:511–573, 1990.

[7] A. C. Antoulas and J. C. Willems. A basis-free approach to linear exact modeling. To appear in *IEEE Transactions on Automatic Control*.

[8] A. H. Sayed and T. Kailath. Recursive solutions to rational interpolation problems. In *Proc. IEEE International Symposium on Circuits and Systems*, pages 2376–2379, May 1992.

[9] A. H. Sayed, T. Constantinescu, and T. Kailath. Square-root algorithms for structured matrices, interpolation, and completion problems. To appear in *IMA Volumes in Mathematics and its Applications*.

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

[11] T. Kailath and A. H. Sayed. Fast algorithms for generalized displacement structures. In *Proc. International Symposium on the Mathematical Theory of Networks and Systems, vol. 2*, pages 27–32, June 1991.

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

[13] I. Gohberg, M. A. Kaashoek, and A. C. M. Ran. Regular rational matrix functions with prescribed null and pole data except at infinity. *Linear Algebra and its Applications*, 137/138:387–412, 1990.

[14] T. Kailath. Signal processing applications of some moment problems. In *Proceedings of Symposia in Applied Mathematics*, volume 37, pages 71–109. American Mathematical Society, 1987.

[15] R. Ackner, H. Lev-Ari, and T. Kailath. Transmission-line models for the modified schur algorithm. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 39(4):290–296, April 1992.

[16] H. Kimura. Directional interpolation in the state space. *Systems and Control Letters*, 10:317–324, 1988.

[17] T. Kailath. *Linear Systems.* Prentice Hall Information and System Sciences Series. Prentice Hall, Inc., Englewood Cliffs, N.J. 07632, 1980.

[18] G. D. Forney, Jr. Minimal bases of rational vector spaces, with applications to multivariable linear systems. *SIAM J. Control*, 13:493–520, 1975.