A Recursive Method for Solving Unconstrained Tangential Interpolation Problems

Tibor Boros, Ali H. Sayed, *Member, IEEE*, and Thomas Kailath, *Life Fellow, IEEE*

***Abstract*— An efficient recursive solution is presented for the one-sided unconstrained tangential interpolation problem. The method relies on the triangular factorization of a certain structured matrix that is implicitly defined by the interpolation data. The recursive procedure admits a physical interpretation in terms of discretized transmission lines. In this framework the generating system is constructed as a cascade of first-order sections. Singular steps occur only when the input data is contradictory, i.e., only when the interpolation problem does not have a solution. Various pivoting schemes can be used to improve numerical accuracy or to impose additional constraints on the interpolants. The algorithm also provides coprime factorizations for all rational interpolants and can be used to solve polynomial interpolation problems such as the general Hermite matrix interpolation problem. A recursive method is proposed to compute a column-reduced generating system that can be used to solve the *minimal* tangential interpolation problem.**

***Index Terms*—Interpolation, matrix decomposition, numerical stability, polynomial matrices, rational functions, rational matrices.**

I. Introduction

Several problems in control, circuit theory, and digital filter design can be reduced to the solution of matrix rational interpolation problems which have been widely studied (see, especially, [1]–[8]). This paper treats left-sided tangential interpolation problems with and without minimality constraints. Applications occur, for example, in minimal partial realization [1]–[3] and in the $\boldsymbol{Q}$-parameterization of stabilizing controllers for unstable plants [9]–[12].

In its simplest form, an interpolation problem would ask for rational functions $y(z)$ that meet the interpolation conditions $y(\alpha_k) = w_k$ for given complex numbers α_k and w_k $(k = 0, 1, \cdots, m-1)$. The interpolants can further be required to have minimal complexity measured in terms of their McMillan degree. An extension of this problem to the vector case would ask for $p \times q$ rational matrix functions $\boldsymbol{Y}(z)$ that satisfy tangential interpolation conditions of the form

$$\boldsymbol{v}_k \boldsymbol{Y}(\alpha_k) = \boldsymbol{w}_k$$

where α_k are given complex numbers and where $\boldsymbol{v}_k$ and $\boldsymbol{w}_k$ are $1 \times p$ and $1 \times q$ complex vectors, respectively.

This paper deals with a generalization of these problems that imposes tangential interpolation conditions both on $\boldsymbol{Y}(z)$ and on its derivatives. We describe an efficient recursive algorithm for computing the rational interpolants and show how to handle minimality constraints recursively. This will be achieved by using a generalized Schur-type algorithm originally developed for the fast triangular factorization of structured matrices and by exploiting some degrees of freedom in its description (see [13] for a review on matrix factorization). Relations to earlier work will be presented after a formal problem statement.

A. Problem Statement

Let $\mathbb{C}(z)$ denote the field of scalar rational functions of a single variable $z \in \mathbb{C}$ and $\mathbb{C}^{p\times q}(z)$ the linear space of $p\times q$ rational matrix functions defined over $\mathbb{C}(z)$. Furthermore, let $\boldsymbol{\mathcal{T}}^k$ be an operator that maps $\boldsymbol{Y}(z) \in \mathbb{C}^{p\times q}(z)$ into an upper triangular Toeplitz block-matrix, $\boldsymbol{\mathcal{T}}^k_{\boldsymbol{Y}}(z) \in \mathbb{C}^{kp\times kq}(z)$, as shown in (1) at the bottom of the next page, where $\boldsymbol{Y}^{(i)}(z)$ stands for the ith derivative of $\boldsymbol{Y}(z)$.

Problem 1.1 (Unconstrained Tangential Interpolation): Consider a set of points $\{\alpha_k\}_{k=0}^{m-1}$ and two sets of row vectors $\{\boldsymbol{v}_k\}_{k=0}^{m-1}$ and $\{\boldsymbol{w}_k\}_{k=0}^{m-1}$ such that

$$\alpha_k \in \mathbb{C} \tag{2a}$$
$$\boldsymbol{v}_k = [\boldsymbol{v}_{k,1} \quad \boldsymbol{v}_{k,2} \quad \cdots \quad \boldsymbol{v}_{k,r_k}], \quad \boldsymbol{v}_{k,i} \in \mathbb{C}^{1\times p} \tag{2b}$$
$$\boldsymbol{w}_k = [\boldsymbol{w}_{k,1} \quad \boldsymbol{w}_{k,2} \quad \cdots \quad \boldsymbol{w}_{k,r_k}], \quad \boldsymbol{w}_{k,i} \in \mathbb{C}^{1\times q}. \tag{2c}$$

1) Given the nodes α_k and the associated vectors $\boldsymbol{v}_k$ and $\boldsymbol{w}_k$, find all rational interpolants $\boldsymbol{Y}(z) \in \mathbb{C}^{p\times q}(z)$ that are analytic at $z = \alpha_k$ and satisfy the interpolation conditions

$$\boldsymbol{v}_k \boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{Y}}(\alpha_k) = \boldsymbol{w}_k, \qquad \text{for all } k. \tag{3}$$

2) Given the extraction points $\{\xi_0, \xi_1, \cdots\}$, evaluate $\{\boldsymbol{Y}(\xi_0), \boldsymbol{Y}(\xi_1), \cdots\}$ for any particular solution $\boldsymbol{Y}(z)$.

In the problem statement, we use the adjective *unconstrained* because no other restrictions, such as minimality or boundedness, are imposed on the rational interpolants. In particular, apart from the analyticity conditions at α_k,

Manuscript received November 4, 1994; revised December 6, 1995 and August 4, 1996. Recommended by Associate Editor, P. Van Dooren. This work was supported by the Army Research Office under contract DAAH-93-G-0029 and by a grant from NSF under Award MIP-9409319. This work was initiated while A. H. Sayed was a Research Associate in the Information Systems Laboratory.

T. Boros is with ArrayComm, Inc., San Jose, CA 95134 USA.

A. H. Sayed was with Stanford University, on leave from Escola Politécnica, da Universidade de São Paulo, Brazil. He is now with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA.

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

Publisher Item Identifier S 0018-9286(99)02125-X.

there are no other constraints on the location of poles of the interpolants. At the same time, in this paper we also investigate the following *constrained* interpolation problem.

Problem 1.2 (Minimal Tangential Interpolation): Given the nodes α_k and the direction vectors $\boldsymbol{v}_k$ and $\boldsymbol{w}_k$ as in (2), find all rational matrix functions $\boldsymbol{Y}(z) \in \mathbb{C}^{p\times q}(z)$ so that

- $\boldsymbol{Y}(z)$ is analytic at $z = \alpha_k$ and satisfies the interpolation conditions (3);
- the complexity of $\boldsymbol{Y}(z)$ (measured in terms of its McMillan degree) is as small as possible.

B. Connections to Earlier Work

Prior work on the unconstrained tangential interpolation problem has been largely carried out by Ball *et al*. [4], [5]. The main result in [4] states that the family of all rational functions that satisfy (3) can be parameterized in terms of a certain linear fractional map. Specifically, it is possible to translate the interpolation data into a so-called *left null pair* $\{\boldsymbol{F}, \boldsymbol{G}\}$ that describes the zero structure of a $(p+q)\times(p+q)$ rational matrix function denoted by

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

Reference [4, Th. 5.1.2] then states that $\boldsymbol{Y}(z)$ satisfies the interpolation conditions (3) if and only if, one can write

$$\begin{aligned} \boldsymbol{Y}(z) = {} & [\boldsymbol{\Theta}_{11}(z)\boldsymbol{P}(z) + \boldsymbol{\Theta}_{12}(z)\boldsymbol{Q}(z)] \\ & \cdot [\boldsymbol{\Theta}_{21}(z)\boldsymbol{P}(z) + \boldsymbol{\Theta}_{22}(z)\boldsymbol{Q}(z)]^{-1} \end{aligned} \tag{4}$$

for some rational matrices $\boldsymbol{P}(z)$ and $\boldsymbol{Q}(z)$. To compute a suitable generating system $\boldsymbol{\Theta}(z)$ (which incidentally is called the *resolvent matrix* by the Odessa school of operator theory), one has to first construct a so-called *right pole pair* $\{\boldsymbol{A}, \boldsymbol{B}\}$ so that the solution $\boldsymbol{R}$ of the Sylvester equation

$$\boldsymbol{FR} - \boldsymbol{RA} = \boldsymbol{GB}$$

is invertible. Then, $\boldsymbol{\Theta}(z)$ can be obtained from a global state-space formula that involves $\boldsymbol{F}, \boldsymbol{G}, \boldsymbol{A}, \boldsymbol{B}$, and $\boldsymbol{R}^{-1}$ (see [5, pp. 23–24, 74, and 103] for the exact definition of left null pairs, right pole pairs, and null-pole triples).

In this paper, we present a different *recursive* method which can be used to compute the generating system as a product of elementary first-order rational matrix functions. The recursive technique allows us to update $\boldsymbol{\Theta}(z)$ whenever a new interpolation point is added to the input data set. The aforementioned algorithm was first studied in connection with a rather different problem, *viz.*, the triangular factorization problem for non-Hermitian matrices possessing *displacement structure* [9], [14]–[13].

The factorization of a non-Hermitian matrix can be naturally associated with two $(p+q)$-input $(p+q)$-output feedforward cascade systems denoted by $\boldsymbol{\Theta}(z)$ and $\boldsymbol{\Gamma}(z)$. Each step of the algorithm determines first-order (lattice) sections in each of the two cascades. The elementary sections obtained this way have transmission zeros: certain inputs at certain frequencies yield zero outputs (this is a general property of any linear system). When the sections are designed appropriately, these "*local*" transmission zeros combine to yield a "*global*" transmission zero (see Proposition 4.2 below) which can be used to solve unconstrained rational interpolation problems. This approach has been successfully used in various other interpolation problems as well (see, e.g., [6] for Schur-type and [15] for unconstrained interpolation problems).

The matrix $\boldsymbol{R}$ that we factor here is *implicitly* determined via a non-Hermitian displacement equation of the form[1]

$$\boldsymbol{R} - \boldsymbol{FRA}^* = \boldsymbol{GJB}^* \tag{5}$$

where $\boldsymbol{F}, \boldsymbol{G}$, and $\boldsymbol{J}$ are constructed directly from the interpolation data [as shown in (10)], while $\boldsymbol{A}$ and $\boldsymbol{B}$ are free parameters that can be chosen to guarantee that no breakdowns occur in the recursive algorithm. In contrast to the methods in [4] and [15], the pair $\{\boldsymbol{A}, \boldsymbol{B}\}$ does not have to be known in advance; the relevant entries can be chosen "*on the fly*" when they are needed in the algorithm (see Algorithm 6.1). The additional degrees of freedom in $\{\boldsymbol{A}, \boldsymbol{B}\}$ can be used to impose various constrains on the rational interpolants (see Section VIII). We further note that in this approach $\boldsymbol{R}$ does not have to be invertible or strongly regular, or even explicitly known.

The main results on minimal interpolation problems to this date appear in [1] and [3] where it is shown that, in the special case when the transfer function $\boldsymbol{\Theta}(z)$ is a *column-reduced* polynomial matrix, it is possible to extract the admissible degrees of complexity as well as the minimal degree of complexity from the linear fractional parameterization formula (4). In the scalar case, Antoulas *et al*. suggested first finding the Lagrange interpolating polynomial and then applying long division (Euclidean algorithm) to obtain $\boldsymbol{\Theta}(z)$ in column-

[1] In principle, the signature matrix $\boldsymbol{J}$ on the right-hand side of (5) could be merged into $\boldsymbol{G}$ or $\boldsymbol{B}$. However, the present form allows us to remain consistent with Schur-type interpolation problems where the underlying displacement equation can be written as $\boldsymbol{R} - \boldsymbol{FRF}^* = \boldsymbol{GJG}^*$, i.e., $\boldsymbol{A} = \boldsymbol{F}$ and $\boldsymbol{B} = \boldsymbol{G}$ [13], [6].

$$\mathcal{T}^k\colon \boldsymbol{Y}(z) \mapsto \mathcal{T}^k_{\boldsymbol{Y}}(z) = \begin{bmatrix} \boldsymbol{Y}(z) & \frac{1}{1!}\boldsymbol{Y}^{(1)}(z) & \frac{1}{2!}\boldsymbol{Y}^{(2)}(z) & \cdots & \frac{1}{(k-1)!}\boldsymbol{Y}^{(k-1)}(z) \\ & \boldsymbol{Y}(z) & \frac{1}{1!}\boldsymbol{Y}^{(1)}(z) & \cdots & \frac{1}{(k-2)!}\boldsymbol{Y}^{(k-2)}(z) \\ & & \boldsymbol{Y}(z) & \cdots & \frac{1}{(k-3)!}\boldsymbol{Y}^{(k-3)}(z) \\ & & & \ddots & \vdots \\ & & & & \boldsymbol{Y}(z) \end{bmatrix} \tag{1}$$

reduced form [1], [2]. Since it is difficult to extend this method to the tangential case, one is obliged first to find a general transfer matrix $\boldsymbol{\Theta}(z)$ by using an unconstrained algorithm. In the next step the generating system $\boldsymbol{\Theta}(z)$ must be transformed into column-reduced form via a sequence of elementary (unimodular) transformations [16]. A detailed algorithm for the construction of a column-reduced rational matrix function from a given null-pole triple has recently appeared in [7]. The corresponding solution set is parameterized via a global state-space formula, but it is not immediately obvious how to update the solutions when a new pole or zero is added to the input data set. In our approach, a suitable column-reduced transfer function can be obtained *recursively* by combining the non-Hermitian generator recursion with a special pivoting scheme.

C. Applications

In order to motivate applications we briefly mention the following two problems that arise in control and circuit theory.

Problem 1.3 (Minimal Partial Realization [3], [5], [16]): Let $\boldsymbol{W}_0, \boldsymbol{W}_1, \cdots, \boldsymbol{W}_{L-1} \in \mathbb{C}^{p\times q}$ be the partial impulse response of a p-input q-output linear multivariable system. Find all admissible transfer functions $\boldsymbol{Y}(z) \in \mathbb{C}^{p\times q}(z)$ that match the measured data $\boldsymbol{W}_i$ for $i = 0, \cdots, L-1$. Which of these rational models have the minimal McMillan degree?

Solution: Introducing the arrays

$$\begin{aligned}\boldsymbol{\mathcal{V}} &= [\boldsymbol{I}_p \quad \boldsymbol{0}_{p\times p} \quad \cdots \quad \boldsymbol{0}_{p\times p}]\\ \boldsymbol{\mathcal{W}} &= [\boldsymbol{W}_0 \quad \boldsymbol{W}_1 \quad \cdots \quad \boldsymbol{W}_{L-1}]\end{aligned}$$

and applying the relation $\boldsymbol{W}_i = (1/i!)\boldsymbol{Y}^{(i)}(0)$ leads to the following tangential interpolation problem: Find all rational matrix functions $\boldsymbol{Y}(z) \in \mathbb{C}^{p\times q}(z)$ that satisfy

$$\boldsymbol{\mathcal{V}}\boldsymbol{T}_{\boldsymbol{Y}}^{L}(0) = \boldsymbol{\mathcal{W}}. \tag{6}$$

It is apparent that each row of $\boldsymbol{\mathcal{V}}$ and $\boldsymbol{\mathcal{W}}$ corresponds to a tangential interpolation constraint on the rational function $\boldsymbol{Y}(z)$. Each constraint involves the derivatives of $\boldsymbol{Y}(z)$ up to the $(L-1)$th order. Thus, the interpolation data set consists of p nodes $\alpha_0 = \alpha_1 = \cdots = \alpha_{p-1} = 0$ with corresponding multiplicities $r_0 = r_1 = \cdots = r_{p-1} = L$.

In the scalar case $p = q = 1$, thus (6) collapses to one tangential constraint at $\alpha_0 = 0$, which involves the derivatives of $y(z) \in \mathbb{C}(z)$ up to the $(L-1)$th order

$$\begin{aligned}&[1 \quad 0 \quad \cdots \quad 0]\\ &\quad\cdot\begin{bmatrix} Y(0) & \frac{1}{1!}y^{(1)}(0) & \cdots & \frac{1}{(L-1)!}y^{(L-1)}(0)\\ & y(0) & \cdots & \frac{1}{(L-2)!}y^{(L-2)}(0)\\ & & \ddots & \vdots\\ & & & y(0)\end{bmatrix}\\ &= [w_0 \quad w_1 \quad \cdots \quad w_{L-1}].\end{aligned}$$

Computing the solutions of the interpolation problem (6) by using a generalized Schur-type algorithm is explained in Sections IV–VI. Selecting the minimal interpolants from this solution set is discussed in Section IX. ■

The next problem is borrowed from [12] (see also [5, Ch. 23], [10], [11], and [17]).

Problem 1.4 ($\boldsymbol{Q}$-Parameterization for Unstable Plants): Let $\boldsymbol{P}(z) \in \mathbb{C}^{p\times q}(z)$ $(p \le q)$ be a strictly proper rational plant, having unstable poles at α_k with corresponding multiplicities r_k $(k = 0, \cdots, m-1)$. Assume, furthermore, that the Laurent expansion of $\boldsymbol{P}(z)$ at $z = \alpha_k$ is given by

$$\boldsymbol{P}(z) = \sum_{j=-r_k}^{\infty} \boldsymbol{P}_{k,j}\cdot(z-\alpha_k)^j, \qquad \boldsymbol{P}_{k,j} \in \mathbb{C}^{p\times q}.$$

Find all controllers $\boldsymbol{C}(z) \in \mathbb{C}^{q\times p}$ so that each entry of the closed-loop transfer matrix

$$\begin{aligned}\boldsymbol{H}_{eu}(z) &= \begin{bmatrix}\boldsymbol{I}_p - \boldsymbol{PQ} & -(\boldsymbol{I}_p - \boldsymbol{PQ})\boldsymbol{P}\\ \boldsymbol{Q} & \boldsymbol{I}_q - \boldsymbol{QP}\end{bmatrix}\\ \boldsymbol{Q} &= \boldsymbol{C}(\boldsymbol{I}_p + \boldsymbol{PC})^{-1}\end{aligned}$$

is stable.

Solution: It can be shown (see [12] for the details) that $\boldsymbol{H}_{eu}(z)$ will be stable if and only if the rational matrix function $\boldsymbol{Q}(z)$ is stable (i.e., analytic *outside* the open unit disc) and satisfies the following tangential interpolation conditions for $k = 0, 1, \cdots, m-1$:

$$\begin{aligned}&[\boldsymbol{P}_{k,-r_k} \quad \boldsymbol{P}_{k,-r_k+1} \quad \cdots \quad \boldsymbol{P}_{k,0} \quad \cdots \quad \boldsymbol{P}_{k,r_k-1}]\boldsymbol{T}_{\boldsymbol{Q}}^{2r_k}(\alpha_k)\\ &\quad = [\overbrace{\boldsymbol{0}}^{1} \quad \cdots \quad \boldsymbol{0} \quad \overbrace{\boldsymbol{I}_p}^{r_k+1} \quad \boldsymbol{0} \quad \cdots \quad \overbrace{\boldsymbol{0}}^{2r_k}].\end{aligned} \tag{7}$$

The stable rational solutions of (7) can be parameterized by using the Hermite interpolating polynomial (see, e.g., [5] and [18]). Computing the Hermite polynomial via a generalized Schur-type algorithm is discussed in Section VIII below. ■

D. Preliminaries

Let $\mathbb{C}[z]$ denote the ring of scalar polynomials of a single variable $z \in \mathbb{C}$, and $\mathbb{C}^{p\times q}[z]$ the module of $p\times q$ polynomial matrices defined over the ring $\mathbb{C}[z]$. The expression $\boldsymbol{Y}(z) = \boldsymbol{N}(z)\boldsymbol{D}^{-1}(z)$ where $\boldsymbol{N}(z)$ and $\boldsymbol{D}(z)$ are polynomial matrices, is called a *matrix fraction description* of the rational function $\boldsymbol{Y}(z)$.

A polynomial matrix is called *column-reduced* if its leading column coefficient matrix has full rank. A square polynomial matrix is column-reduced if and only if, the sum of its column degrees is equal to the degree of its determinant.

The complexity of a polynomial matrix $\boldsymbol{Y}(z)$ is measured by its *McMillan degree* $\delta\{\boldsymbol{Y}(z)\}$. In particular, $\delta\{\boldsymbol{Y}(z)\}$ can be determined by transforming $\boldsymbol{Y}(z)$ into column-reduced form and taking the sum of the column degrees. By definition, the McMillan degree of a rational matrix function $\boldsymbol{Y}(z) = \boldsymbol{N}_R(z)\boldsymbol{D}_R^{-1}(z)$ is equal to the McMillan degree of the associated polynomial matrix $[\boldsymbol{N}_R^T(z) \quad \boldsymbol{D}_R^T(z)]^T$ (see, e.g., [16]).

Let $\boldsymbol{F} \in \mathbb{C}^{n\times n}$ be a constant matrix and let $\phi(z) \in \mathbb{C}(z)$ be a scalar function that is analytic at the eigenvalues of $\boldsymbol{F}$. The value of $\phi(\cdot)$ on the matrix $\boldsymbol{F}$ can be defined via the *Riesz* formula

$$\phi(\boldsymbol{F}) = \frac{1}{2\pi i}\int_{\gamma}(z\boldsymbol{I}_n - \boldsymbol{F})^{-1}\phi(z)\,dz \tag{8}$$

where γ is a rectifiable simple contour that surrounds the spectrum of $\boldsymbol{F}$ (see, e.g., [5, p. 597]). It is especially easy to evaluate this integral when $\boldsymbol{F}$ is a Jordan matrix. For example

$$\begin{aligned}\boldsymbol{F} = \begin{bmatrix} f_0 & 1 & & \\ & f_0 & \ddots & \\ & & \ddots & 1 \\ & & & f_0\end{bmatrix} &\rightarrow \phi(\boldsymbol{F}) \\ = \begin{bmatrix} \phi(f_0) & \frac{1}{1!}\phi^{(1)}(f_0) & \cdots & \frac{1}{(n-1)!}\phi^{(n-1)}(f_0) \\ & \phi(f_0) & \ddots & \vdots \\ & & \ddots & \frac{1}{1!}\phi^{(1)}(f_0) \\ & & & \phi(f_0)\end{bmatrix} \\ = \boldsymbol{T}_{\phi}^{n}(f_0).\end{aligned}$$

Finally, observe that the operator $\boldsymbol{T}^k$ defined in (1) satisfies the relation

$$\boldsymbol{T}^k_{\boldsymbol{Y}_0\boldsymbol{Y}_1}(z) = \boldsymbol{T}^k_{\boldsymbol{Y}_0}(z)\boldsymbol{T}^k_{\boldsymbol{Y}_1}(z) \tag{9}$$

for any two rational matrices $\boldsymbol{Y}_0(z) \in \mathbb{C}^{p\times q}(z)$ and $\boldsymbol{Y}_1(z) \in \mathbb{C}^{q\times s}(z)$. By induction, this statement can be extended to the product of a countable number of rational matrices. In particular, $\boldsymbol{T}^k$: $\mathbb{C}^{p\times p}(z) \rightarrow \mathbb{C}^{kp\times kp}(z)$ is an algebra homomorphism.

This paper is divided into nine sections. In Section II, we state a sufficient and necessary condition for the solvability of Problem 1.1. In Sections III and IV we briefly review the non-Hermitian forward generator recursion and the associated cascade system interpretation. Section V deals with various parameterization schemes for the family of rational interpolants. The main algorithm that solves Problem 1.1 appears in Section VI, while Section VII contains the physical interpretation of the results in terms of discretized transmission lines. Section VIII deals with polynomial interpolation problems such as the general Hermite matrix interpolation problem. In Section IX, we obtain a generating system with a column-reduced transfer matrix which gives a nice solution to the minimal rational interpolation problem. Concluding remarks are given in Section X.

It may be useful to note that, when specialized to the scalar case, the procedures of this paper exhibit several improvements over the scalar algorithm that we had presented earlier in [15]. In particular, here we have a simpler procedure for avoiding breakdowns and for obtaining column-reduced generating systems.

II. Solvability Condition

Problem 1.1 may fail to admit a solution when the interpolation data is contradictory. The solvability issues of one- and two-sided tangential interpolation problems have been analyzed in [19] by using a residual interpolation framework. In this section we present a more direct algebraic approach, which shows that solving a tangential interpolation problem is equivalent to solving a matrix Padé approximation problem where the Taylor coefficients obey a set of linear constraints.

Thus let $\boldsymbol{F}, \boldsymbol{G}$, and $\boldsymbol{J}$ be three arrays constructed from the interpolation data as

$$\boldsymbol{F} = \begin{bmatrix}\boldsymbol{\mathcal{J}}_0 & & & \\ & \boldsymbol{\mathcal{J}}_1 & & \\ & & \ddots & \\ & & & \boldsymbol{\mathcal{J}}_{m-1}\end{bmatrix}$$

$$\boldsymbol{G} = \begin{bmatrix}\boldsymbol{V}_0 & \boldsymbol{W}_0 \\ \boldsymbol{V}_1 & \boldsymbol{W}_1 \\ \vdots & \vdots \\ \boldsymbol{V}_{m-1} & \boldsymbol{W}_{m-1}\end{bmatrix} \equiv [\boldsymbol{V} \quad \boldsymbol{W}], \qquad \boldsymbol{J} = \begin{bmatrix}\boldsymbol{I}_p & \\ & -\boldsymbol{I}_q\end{bmatrix} \tag{10}$$

where $\boldsymbol{\mathcal{J}}_k \in \mathbb{C}^{r_k\times r_k}, \boldsymbol{V}_k \in \mathbb{C}^{r_k\times p}$, and $\boldsymbol{W}_k \in \mathbb{C}^{r_k\times q}$ are defined by

$$\boldsymbol{\mathcal{J}}_k = \begin{bmatrix}\alpha_k & & & \\ 1 & \alpha_k & & \\ & \ddots & \ddots & \\ & & 1 & \alpha_k\end{bmatrix}, \qquad \boldsymbol{V}_k = \begin{bmatrix}\boldsymbol{v}_{k,1} \\ \boldsymbol{v}_{k,2} \\ \vdots \\ \boldsymbol{v}_{k,r_k}\end{bmatrix}$$

$$\boldsymbol{W}_k = \begin{bmatrix}\boldsymbol{w}_{k,1} \\ \boldsymbol{w}_{k,2} \\ \vdots \\ \boldsymbol{w}_{k,r_k}\end{bmatrix}.$$

The triplet $\{\boldsymbol{\mathcal{J}}_{r_k}(\alpha_k), \boldsymbol{V}_k, \boldsymbol{W}_k\}$ carries all information about the kth tangential interpolation constraint. The following statement is now valid.

Proposition 2.1 (Solvability Condition): Problem 1.1 is solvable if and only if

$$\operatorname{Im}\boldsymbol{W} \in \operatorname{Im}[\boldsymbol{V} \quad \boldsymbol{F}\boldsymbol{V} \quad \boldsymbol{F}^2\boldsymbol{V} \quad \cdots \quad \boldsymbol{F}^{r_{\max}-1}\boldsymbol{V}] \tag{11}$$

where $r_{\max} = \max\{r_0, r_1, \cdots, r_{m-1}\}$. ■

In system theoretical terms, the unconstrained tangential problem is solvable if and only if the columns of $\boldsymbol{W}$ lie in the *controllable subspace* of the pair $\{\boldsymbol{F}, \boldsymbol{V}\}$. If the pair $\{\boldsymbol{F}, \boldsymbol{V}\}$ is controllable then Problem 1.1 can be solved for any right-hand side $\boldsymbol{W}$. If $\{\boldsymbol{F}, \boldsymbol{V}\}$ is not controllable but the columns of $\boldsymbol{W}$ lie in the controllable subspace of $\{\boldsymbol{F}, \boldsymbol{V}\}$ then the interpolation data is redundant. Finally, if the columns of $\boldsymbol{W}$ do not lie in the controllable subspace of $\{\boldsymbol{F}, \boldsymbol{V}\}$ then the interpolation data is contradictory. A sufficient condition for the controllability of $\{\boldsymbol{F}, \boldsymbol{V}\}$ can be formulated in terms of the direction vectors $\boldsymbol{v}_{k,j}$ as follows:

Lemma 2.2 (Controllability): If $\boldsymbol{v}_{k,1} \neq \boldsymbol{0}$ for all k, and the vectors $\boldsymbol{v}_{k,1}, \boldsymbol{v}_{k+1,1}, \cdots, \boldsymbol{v}_{k+l,1}$ are linearly independent whenever $\alpha_k = \alpha_{k+1} = \cdots = \alpha_{k+l}$, then the pair $\{\boldsymbol{F}, \boldsymbol{V}\}$ is controllable. ■

In particular, the interpolation problem is always solvable when the nodes α_k are distinct and $\boldsymbol{v}_{k,1} \neq \boldsymbol{0}$ for all k.

Proof of Proposition 2.1 and Lemma 2.2—Part A: Let us first consider the special case when the interpolation points coincide. Assume therefore that

$$\alpha_0 = \alpha_1 = \cdots = \alpha_{m-1} = \overline{\alpha}. \qquad (12)$$

The kth tangential constraint can be rewritten as

$$\begin{bmatrix} \boldsymbol{v}_{k,1} & & & & \\ \boldsymbol{v}_{k,2} & \boldsymbol{v}_{k,1} & & & \\ \boldsymbol{v}_{k,3} & \boldsymbol{v}_{k,2} & \boldsymbol{v}_{k,1} & & \\ \vdots & \vdots & \vdots & \ddots & \\ \boldsymbol{v}_{k,r_k} & \boldsymbol{v}_{k,r_k-1} & \boldsymbol{v}_{k,r_k-2} & \cdots & \boldsymbol{v}_{k,1} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{Y}(\overline{\alpha}) \\ \boldsymbol{Y}^{(1)}(\overline{\alpha}) \\ \frac{1}{2!}\boldsymbol{Y}^{(2)}(\overline{\alpha}) \\ \vdots \\ \frac{1}{(r_k-1)!}\boldsymbol{Y}^{(r_k-1)}(\overline{\alpha}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_{k,1} \\ \boldsymbol{w}_{k,2} \\ \boldsymbol{w}_{k,3} \\ \vdots \\ \boldsymbol{w}_{k,r_k} \end{bmatrix}$$

or more compactly as

$$[\boldsymbol{V}_k \quad \boldsymbol{Z}_{r_k}\boldsymbol{V}_k \quad \boldsymbol{Z}_{r_k}^2\boldsymbol{V}_k \quad \cdots \quad \boldsymbol{Z}_{r_k}^{r_k-1}\boldsymbol{V}_k] \cdot \begin{bmatrix} \boldsymbol{Y}(\overline{\alpha}) \\ \vdots \\ \frac{1}{(r_k-1)!}\boldsymbol{Y}^{(r_k-1)}(\overline{\alpha}) \end{bmatrix} = \boldsymbol{W}_k$$

where $\boldsymbol{Z}_{r_k}$ denotes the $r_k \times r_k$ lower triangular shift matrix with ones below the main diagonal. The set of all interpolation conditions can be expressed as shown in the equation at the bottom of the page, or, equivalently, as

$$[\boldsymbol{V} \quad \boldsymbol{Z}\boldsymbol{V} \quad \boldsymbol{Z}^2\boldsymbol{V} \quad \cdots \quad \boldsymbol{Z}^{r_{\max}-1}\boldsymbol{V}] \cdot \begin{bmatrix} \boldsymbol{Y}(\overline{\alpha}) \\ \boldsymbol{Y}^{(1)}(\overline{\alpha}) \\ \vdots \\ \frac{1}{(r_{\max}-1)!}\boldsymbol{Y}^{(r_{\max}-1)}(\overline{\alpha}) \end{bmatrix} = \boldsymbol{W} \qquad (13)$$

where $\boldsymbol{Z} = \boldsymbol{Z}_{r_0} \oplus \boldsymbol{Z}_{r_1} \oplus \cdots \oplus \boldsymbol{Z}_{r_{m-1}}$. Problem 1.1 can now be reformulated as follows: Find all rational matrix functions $\boldsymbol{Y}(z)$ that can be expanded at $z = \overline{\alpha}$ as

$$\boldsymbol{Y}(z) = \boldsymbol{Y}(\overline{\alpha}) + \boldsymbol{Y}^{(1)}(\overline{\alpha})(z-\overline{\alpha}) + \cdots + \frac{1}{(r_{\max}-1)!} \cdot \boldsymbol{Y}^{(r_{\max}-1)}(\overline{\alpha})(z-\overline{\alpha})^{(r_{\max}-1)} + \mathcal{O}\{(z-\overline{\alpha})^{r_{\max}}\}$$

for some coefficients $\{\boldsymbol{Y}^{(i)}(\overline{\alpha})\}_{i=0}^{r_{\max}-1}$ that satisfy the linear constraints (13).

Obtaining $\boldsymbol{Y}(z)$ from a given set of coefficients $\boldsymbol{Y}^{(i)}(\overline{\alpha})$ is the well-studied matrix Padé approximation problem, which always has a solution.[2] Thus Problem 1.1 is solvable if and only if the linear system (13) is solvable. The solvability condition for (13) is, in turn, given by

$$\operatorname{Im} \boldsymbol{W} \in \operatorname{Im} [\boldsymbol{V} \quad \boldsymbol{Z} \quad \boldsymbol{V} \quad \boldsymbol{Z}^2\boldsymbol{V} \quad \cdots \quad \boldsymbol{Z}^{(r_{\max}-1)}\boldsymbol{V}]. \qquad (14)$$

Multiplication by a nonsingular matrix from the right leaves the column space of $[\boldsymbol{V} \quad \boldsymbol{Z}\boldsymbol{V} \quad \cdots \quad \boldsymbol{Z}^{(r_{\max}-1)}\boldsymbol{V}]$ unaltered. Therefore, (14) can be re-expressed in terms of $\boldsymbol{F} = \boldsymbol{Z} + \overline{\alpha}\boldsymbol{I}$ as

$$\begin{aligned} \operatorname{Im} \boldsymbol{W} \in & [\boldsymbol{V} \quad \boldsymbol{Z}\boldsymbol{V} \quad \cdots \quad \boldsymbol{Z}^{r_{\max}-1}V] \\ & \cdot \begin{bmatrix} \boldsymbol{I} & \overline{\alpha}\boldsymbol{I} & \overline{\alpha}^2\boldsymbol{I} & \cdots & \overline{\alpha}^{r_{\max}-1}\boldsymbol{I} \\ & \boldsymbol{I} & 2\overline{\alpha}\boldsymbol{I} & \cdots & (r_{\max}-1)\overline{\alpha}^{r_{\max}-2}\boldsymbol{I} \\ & & \boldsymbol{I} & \cdots & \binom{r_{\max}-1}{2}\overline{\alpha}^{r_{\max}-3}\boldsymbol{I} \\ & & & \ddots & \vdots \\ & & & & \boldsymbol{I} \end{bmatrix} \\ = & [\boldsymbol{V} \quad \boldsymbol{F}\boldsymbol{V} \quad \cdots \quad \boldsymbol{F}^{r_{\max}-1}\boldsymbol{V}]. \end{aligned}$$

This proves Proposition 2.1 in the particular case when (12) is valid. It is easy to see that the coefficient matrix in (13) has full rank whenever the vectors $\boldsymbol{v}_{0,1}, \boldsymbol{v}_{1,1}, \cdots, \boldsymbol{v}_{m-1,1}$ are linearly independent. This proves Lemma 2.2.

Part B: In general, the interpolation data contains several sets of coinciding nodes. Without restriction of generality, assume that the interpolation points can be arranged into L subsets as

$$\begin{aligned} \alpha_0 = \alpha_1 = \cdots = \alpha_{m_0-1} &= \overline{\alpha}_0 \\ \alpha_{m_0} = \alpha_{m_0+1} = \cdots = \alpha_{m_0+m_1-1} &= \overline{\alpha}_1 \\ &\vdots \\ \alpha_{m_0+\cdots+m_{L-2}} = \cdots = \alpha_{m_0+\cdots+m_{L-1}-1} &= \overline{\alpha}_{L-1} \end{aligned} \qquad (15)$$

where the nodes $\{\overline{\alpha}_\ell\}_{\ell=0}^{L-1}$ are distinct. Now introduce the auxiliary indexes

$$\mu_0 = 0, \qquad \mu_\ell = \sum_{i=0}^{\ell-1} m_i, \qquad \ell = 1, 2, \cdots L-1$$

[2] Also note that the matrix Padé problem can be reduced to a set of independent scalar Padé problems.

$$\begin{bmatrix} \boldsymbol{V}_0 & \boldsymbol{Z}_{r_0}\boldsymbol{V}_0 & \boldsymbol{Z}_{r_0}^2\boldsymbol{V}_0 & \cdots & \boldsymbol{Z}_{r_0}^{r_{\max}-1}\boldsymbol{V}_0 \\ \boldsymbol{V}_1 & \boldsymbol{Z}_{r_1}\boldsymbol{V}_1 & \boldsymbol{Z}_{r_1}^2\boldsymbol{V}_1 & \cdots & \boldsymbol{Z}_{r_1}^{r_{\max}-1}\boldsymbol{V}_1 \\ \vdots & \vdots & \vdots & & \vdots \\ \boldsymbol{V}_{m-1} & \boldsymbol{Z}_{r_{m-1}}\boldsymbol{V}_{m-1} & \boldsymbol{Z}_{r_{m-1}}^2\boldsymbol{V}_{m-1} & \cdots & \boldsymbol{Z}_{r_{m-1}}^{r_{\max}-1}\boldsymbol{V}_{m-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{Y}(\overline{\alpha}) \\ \boldsymbol{Y}^{(1)}(\overline{\alpha}) \\ \vdots \\ \frac{1}{(r_{\max}-1)!}\boldsymbol{Y}^{(r_{\max}-1)}(\overline{\alpha}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{W}_0 \\ \boldsymbol{W}_1 \\ \vdots \\ \boldsymbol{W}_{m-1} \end{bmatrix}$$

and define the arrays $\{\overline{\boldsymbol{F}}_\ell, \overline{\boldsymbol{V}}_\ell, \overline{\boldsymbol{W}}_\ell\}_{\ell=0}^{L-1}$ as

$$\overline{\boldsymbol{F}}_\ell = \begin{bmatrix} \mathcal{J}_{\mu_\ell} & & & \\ & \mathcal{J}_{\mu_\ell+1} & & \\ & & \ddots & \\ & & & \mathcal{J}_{\mu_\ell+m_\ell-1} \end{bmatrix}$$

$$\overline{\boldsymbol{V}}_\ell = \begin{bmatrix} \boldsymbol{V}_{\mu_\ell} \\ \boldsymbol{V}_{\mu_\ell+1} \\ \vdots \\ \boldsymbol{V}_{\mu_\ell+m_\ell-1} \end{bmatrix}, \qquad \overline{\boldsymbol{W}}_\ell = \begin{bmatrix} \boldsymbol{W}_{\mu_\ell} \\ \boldsymbol{W}_{\mu_\ell+1} \\ \vdots \\ \boldsymbol{W}_{\mu_\ell+m_\ell-1} \end{bmatrix}.$$

By repeating the arguments of Part A for each subset in (15), Problem 1.1 can be reformulated as follows: Given the distinct nodes $\{\overline{\alpha}_\ell\}_{\ell=0}^{L-1}$, find all rational matrix functions $\boldsymbol{Y}(z)$ that can be expanded at $z = \overline{\alpha}_\ell$ as

$$\begin{aligned}\boldsymbol{Y}(z) = &\boldsymbol{Y}(\overline{\alpha}_\ell) + \boldsymbol{Y}^{(1)}(\overline{\alpha}_\ell)(z-\overline{\alpha}_\ell) + \cdots + \frac{1}{(r_{\max}-1)!} \\ &\cdot \boldsymbol{Y}^{(r_{\max}-1)}(\overline{\alpha}_\ell)(z-\overline{\alpha}_\ell)^{(r_{\max}-1)} \\ &+ \boldsymbol{\mathcal{O}}\{(z-\overline{\alpha}_\ell)^{r_{\max}}\}\end{aligned}$$

for some coefficients $\{\boldsymbol{Y}^{(i)}(\overline{\alpha}_\ell)\}_{i=0}^{r_{\max}-1}$ that satisfy the linear constraints

$$[\overline{\boldsymbol{V}}_\ell \quad \overline{\boldsymbol{Z}}_\ell\overline{\boldsymbol{V}}_\ell \quad \overline{\boldsymbol{Z}}_\ell^2\overline{\boldsymbol{V}}_\ell \quad \cdots \quad \overline{\boldsymbol{Z}}_\ell^{r_{\max}-1}\overline{\boldsymbol{V}}_\ell] \cdot \begin{bmatrix} \boldsymbol{Y}(\overline{\alpha}_\ell) \\ \boldsymbol{Y}^{(1)}(\overline{\alpha}_\ell) \\ \vdots \\ \frac{1}{(r_{\max}-1)!}\boldsymbol{Y}^{(r_{\max}-1)}(\overline{\ell}) \end{bmatrix} = \overline{\boldsymbol{W}}_\ell \tag{16}$$

where $\overline{\boldsymbol{Z}}_\ell = \boldsymbol{Z}_{\mu_\ell} \oplus \boldsymbol{Z}_{\mu_\ell+1} \oplus \cdots \oplus \boldsymbol{Z}_{\mu_\ell+r_\ell-1}$. If the coefficients $\boldsymbol{Y}^{(i)}(\overline{\alpha}_\ell)$ are known, then the rational interpolants $\boldsymbol{Y}(z)$ can be obtained by solving a multiple-point matrix Padé problem which is guaranteed to have a solution. Therefore, Problem 1.1 is solvable if and only if

$$\operatorname{Im}\overline{\boldsymbol{W}}_\ell \in \operatorname{Im}[\overline{\boldsymbol{V}}_\ell \quad \overline{\boldsymbol{Z}}_\ell\overline{\boldsymbol{V}}_\ell \quad \overline{\boldsymbol{Z}}_\ell^2\overline{\boldsymbol{V}}_\ell \quad \cdots \quad \overline{\boldsymbol{Z}}_\ell^{(r_{\max}-1)}\overline{\boldsymbol{V}}_\ell] \quad \text{for all } \ell. \tag{17}$$

If the assumptions of Lemma 2.2 are valid, then the coefficient matrix in (16) has full rank and therefore (17) holds for any $\overline{\boldsymbol{W}}_\ell$. This concludes the proof of Lemma 2.2. By using nonsingular transformations it can be shown that (17) is equivalent to

$$\operatorname{Im}\overline{\boldsymbol{W}}_\ell \in \operatorname{Im}[\overline{\boldsymbol{V}}_\ell \quad \overline{\boldsymbol{F}}_\ell\overline{\boldsymbol{V}}_\ell \quad \overline{\boldsymbol{F}}_\ell^2\overline{\boldsymbol{V}}_\ell \quad \cdots \quad \overline{\boldsymbol{F}}_\ell^{(r_{\max}-1)}\overline{\boldsymbol{V}}_\ell] \quad \text{for all } \ell. \tag{18}$$

The equivalence of (18) and (11) follows immediately from the fact that

$$\boldsymbol{F} = \begin{bmatrix} \overline{\boldsymbol{F}}_0 & & & \\ & \overline{\boldsymbol{F}}_1 & & \\ & & \ddots & \\ & & & \overline{\boldsymbol{F}}_{L-1} \end{bmatrix}, \quad \boldsymbol{V} = \begin{bmatrix} \overline{\boldsymbol{V}}_0 \\ \overline{\boldsymbol{V}}_1 \\ \vdots \\ \overline{\boldsymbol{V}}_{L-1} \end{bmatrix}$$

and

$$\boldsymbol{W} = \begin{bmatrix} \overline{\boldsymbol{W}}_0 \\ \overline{\boldsymbol{W}}_1 \\ \vdots \\ \overline{\boldsymbol{W}}_{L-1} \end{bmatrix}.$$

This proves Proposition 2.1 in the general case. ■

III. Triangular Factorization and the Generalized Schur Algorithm

The key step in our approach to rational interpolation is via an apparently unrelated matrix factorization algorithm that we review in this section. To this end, let $\boldsymbol{R} \in \mathbb{C}^{n\times n}$ be a *structured* matrix that satisfies a displacement equation of the form

$$\boldsymbol{R} - \boldsymbol{F}\boldsymbol{R}\boldsymbol{A}^* = \boldsymbol{G}\boldsymbol{J}\boldsymbol{B}^* \tag{19}$$

where $\boldsymbol{G}, \boldsymbol{B} \in \mathbb{C}^{n\times r}$ $(n \geq r)$ are full-rank generator matrices, $\boldsymbol{J} \in \mathbb{C}^{r\times r}$ is a signature matrix of the form

$$\boldsymbol{J} = \boldsymbol{I}_p \oplus -\boldsymbol{I}_q = \begin{bmatrix} \boldsymbol{I}_p & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I}_q \end{bmatrix}, \qquad p+q=r$$

and $\boldsymbol{F}, \boldsymbol{A} \in \mathbb{C}^{n\times n}$ are lower triangular matrices with diagonal entries $\{f_0, f_1, \cdots, f_{n-1}\}$ and $\{a_0, a_1, \cdots, a_{n-1}\}$, respectively. The quantity $r = \operatorname{rank}\{\boldsymbol{R} - \boldsymbol{F}\boldsymbol{R}\boldsymbol{A}^*\}$ is called the *displacement rank* of $\boldsymbol{R}$ with respect to $\boldsymbol{F}$ and $\boldsymbol{A}$. In the rest of the paper we shall assume that $1 - f_i a_j^* \neq 0$ for all i and j so that (19) has a unique solution for $\boldsymbol{R}$. In connection with interpolation problems $\boldsymbol{F}$ will be a Jordan matrix as defined in (10) so that

$$\begin{aligned} f_0 = f_1 = \cdots = f_{r_0-1} &= \alpha_0 \\ f_{r_0} = f_{r_0+1} = \cdots = f_{r_0+r_1-1} &= \alpha_1 \\ &\vdots \\ f_{r_0+\cdots+r_{m-2}} = \cdots = f_{r_0+\cdots+r_{m-2}+r_{m-1}-1} &= \alpha_{m-1}. \end{aligned} \tag{20}$$

The classical method to compute the triangular factors of $\boldsymbol{R}$ is provided by the well-known Gauss/Schur reduction procedure

$$\begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & \boldsymbol{R}_{i+1} & \\ 0 & & & \end{bmatrix} = \boldsymbol{R}_i - \boldsymbol{l}_i d_i^{-1}\boldsymbol{u}_i^*, \quad i = 0, 1, \cdots, n-1$$
$$\boldsymbol{R}_0 = \boldsymbol{R}$$

where $\boldsymbol{l}_i, \boldsymbol{u}_i$, and d_i denote the first column, the first row, and the upper-left corner element of $\boldsymbol{R}_i$. Intrinsically this method requires $\mathcal{O}(n^3)$ additions and multiplications. A faster procedure can be obtained by exploiting the fact that the ith Schur complement $\boldsymbol{R}_i$ inherits the displacement structure of

$\boldsymbol{R}$, i.e., it satisfies a displacement equation of the form

$$\boldsymbol{R}_i - \boldsymbol{F}_i\boldsymbol{R}_i\boldsymbol{A}_i^* = \boldsymbol{G}_i\boldsymbol{J}\boldsymbol{B}_i^*$$

where $\{\boldsymbol{G}_i, \boldsymbol{B}_i, \boldsymbol{F}_i, \boldsymbol{A}_i\}$ can be obtained from $\{\boldsymbol{G}, \boldsymbol{B}, \boldsymbol{F}, \boldsymbol{A}\}$ via the following algorithm [13]:

Algorithm 3.1 (Generalized Schur Algorithm)[3]*:* Start with $\boldsymbol{G}_0 = \boldsymbol{G}, \boldsymbol{B}_0 = \boldsymbol{B}$, and repeat the following steps for $i = 0, 1, \cdots, n-1$.

1) Obtain $\boldsymbol{F}_i$ and $\boldsymbol{A}_i$ by deleting the first i rows and columns of $\boldsymbol{F}$ and $\boldsymbol{A}$.
2) Choose $r\times r$ matrices $\boldsymbol{\Theta}_i$ and $\boldsymbol{\Gamma}_i$ that satisfy $\boldsymbol{\Theta}_i\boldsymbol{J}\boldsymbol{\Gamma}_i^* = \boldsymbol{J}$ and transform the generators $\boldsymbol{G}_i$ and $\boldsymbol{B}_i$ to proper form. This means that $\boldsymbol{\Theta}_i$ and $\boldsymbol{\Gamma}_i$ reduce the first row of $\boldsymbol{G}_i$ (denoted by $\boldsymbol{g}_{i,0}$) and the first row of $\boldsymbol{B}_i$ (denoted by $\boldsymbol{b}_{i,0}$) to the forms

$$\begin{aligned} \boldsymbol{g}_{i,0}\boldsymbol{\Theta}_i &= [\,0 \quad \cdots \quad 0 \quad \overset{\overset{j}{\smile}}{*} \quad 0 \quad \cdots \quad 0\,] \\ \boldsymbol{b}_{i,0}\boldsymbol{\Gamma}_i &= [\,0 \quad \cdots \quad 0 \quad \overset{\overset{j}{\smile}}{*} \quad 0 \quad \cdots \quad 0\,] \end{aligned} \tag{21}$$

with a single nonzero entry in the same column position, say the jth position $j \in \{1, 2, \cdots, r\}$.
3) Multiply the jth column of $(\boldsymbol{G}_i\boldsymbol{\Theta}_i)$ by $\phi_i(\boldsymbol{F}_i) = (\boldsymbol{F}_i - f_i\boldsymbol{I}_{n-i})(\boldsymbol{I}_{n-i} - a_i^*\boldsymbol{F}_i)^{-1}$ and the jth column of $(\boldsymbol{B}_i\boldsymbol{\Gamma}_i)$ by $\psi_i(\boldsymbol{A}_i) = (\boldsymbol{A}_i - a_i\boldsymbol{I}_{n-i})(\boldsymbol{I}_{n-i} - f_i^*\boldsymbol{A}_i)^{-1}$ [the functions $\phi(z)$ and $\psi(z)$ are defined in (24)].
4) Delete the first row of the resulting arrays to obtain the new generators $\boldsymbol{G}_{i+1}$ and $\boldsymbol{B}_{i+1}$.

■

The *generator recursion* that describes how to obtain $\boldsymbol{G}_{i+1}$ and $\boldsymbol{B}_{i+1}$ from $\boldsymbol{G}_i$ and $\boldsymbol{B}_i$ can be written in a compact form as

$$\begin{aligned} \begin{bmatrix} \mathbf{0} \\ \boldsymbol{G}_{i+1} \end{bmatrix} &= \phi_i(\boldsymbol{F}_i)\boldsymbol{G}_i\boldsymbol{\Theta}_i \begin{bmatrix} \mathbf{0}_{j-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{r-j} \end{bmatrix} + \boldsymbol{G}_i\boldsymbol{\Theta}_i \\ &\quad \cdot \begin{bmatrix} \boldsymbol{I}_{j-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I}_{r-j} \end{bmatrix} \end{aligned} \tag{22a}$$

$$\begin{aligned} \begin{bmatrix} \mathbf{0} \\ \boldsymbol{B}_{i+1} \end{bmatrix} &= \psi_i(\boldsymbol{A}_i)\boldsymbol{B}_i\boldsymbol{\Gamma}_i \begin{bmatrix} \mathbf{0}_j & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{r-j-1} \end{bmatrix} + \boldsymbol{B}_i\boldsymbol{\Gamma}_i \\ &\quad \cdot \begin{bmatrix} \boldsymbol{I}_{j-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I}_{r-j} \end{bmatrix}. \end{aligned} \tag{22b}$$

These formulas have the following simple array interpretations:

$$\boldsymbol{G}_i = \begin{bmatrix} * & * & * \\ * & * & * \\ \vdots & \vdots & \vdots \\ * & * & * \end{bmatrix} \xrightarrow{\boldsymbol{\Theta}_i} \begin{bmatrix} \mathbf{0} & * & \mathbf{0} \\ * & * & * \\ \vdots & \vdots & \vdots \\ * & * & * \end{bmatrix}$$

$$\xrightarrow{\phi_i(F_i)\times j\text{th col.}} \begin{bmatrix} \mathbf{0} & 0 & \mathbf{0} \\ * & * & * \\ \vdots & \vdots & \vdots \\ * & * & * \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{G}_{i+1} \end{bmatrix}$$

$$\boldsymbol{B}_i = \begin{bmatrix} * & * & * \\ * & * & * \\ \vdots & \vdots & \vdots \\ * & * & * \end{bmatrix} \xrightarrow{\boldsymbol{\Gamma}_i} \begin{bmatrix} \mathbf{0} & * & \mathbf{0} \\ * & * & * \\ \vdots & \vdots & \vdots \\ * & * & * \end{bmatrix}$$

$$\xrightarrow{\psi_i(A_i)\times j\text{th col.}} \begin{bmatrix} \mathbf{0} & 0 & \mathbf{0} \\ * & * & * \\ \vdots & \vdots & \vdots \\ * & * & * \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{B}_{i+1} \end{bmatrix}.$$

The nonsingular transformations $\boldsymbol{\Theta}_i$ and $\boldsymbol{\Gamma}_i$ can be implemented in a variety of ways, e.g., by using a suitable sequence of elementary Householder projections or Givens rotations, and hyperbolic transformations. When the arrays $\boldsymbol{F}$ and $\boldsymbol{A}$ are sparse (e.g., diagonal or bidiagonal), the generalized Schur algorithm requires only $\mathcal{O}(rn^2)$ operations. The triangular factors of $\boldsymbol{R}$ can be computed from $\boldsymbol{G}_i, \boldsymbol{B}_i$ as

$$\begin{aligned} \boldsymbol{l}_i &= (\boldsymbol{I}_i - a_i^*\boldsymbol{F}_i)^{-1}\boldsymbol{G}_i\boldsymbol{J}\boldsymbol{b}_i^*, \\ \boldsymbol{u}_i &= (\boldsymbol{I}_i - f_i^*\boldsymbol{A}_i)^{-1}\boldsymbol{B}_i\boldsymbol{J}\boldsymbol{g}_i^*, \qquad d_i = \frac{\boldsymbol{g}_i\boldsymbol{J}\boldsymbol{b}_i^*}{1 - f_i a_i^*}. \end{aligned}$$

The generalized Schur algorithm may break down if the matrix $\boldsymbol{R}$ is not strongly regular; this issue is addressed in Section VI.

IV. Cascade Systems

The generator recursion (22) can be described in terms of transfer functions as shown in [9], [13], and [14]. Similar models have also appeared in the inverse scattering theory of lossy transmission lines [20]. In the function domain each step of (22) can be associated with two r-input r-output first-order systems (see Fig. 1) with transfer functions

$$\boldsymbol{\Theta}_i(z) = \boldsymbol{\Theta}_i \begin{bmatrix} \boldsymbol{I}_{j-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \phi_i(z) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I}_{r-j} \end{bmatrix}$$

and

$$\boldsymbol{\Gamma}_i(z) = \boldsymbol{\Gamma}_i \begin{bmatrix} \boldsymbol{I}_{j-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \psi_i(z) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I}_{r-j} \end{bmatrix} \tag{23}$$

where $\boldsymbol{\Theta}_i$ and $\boldsymbol{\Gamma}_i$ are obtained at the ith step of the generalized Schur algorithm, and the Möbius transformations $\phi_i(z)$ and $\psi_i(z)$ are defined as

$$\phi_i(z) = \frac{z - f_i}{1 - a_i^* z}, \qquad \psi_i(z) = \frac{z - a_i}{1 - f_i^* z}. \tag{24}$$

Note that $\phi_i(z)$ and $\psi_i(z)$ satisfy $\phi_i(z)\psi_i(1/z^*)^* = 1$. Therefore, the transfer functions $\boldsymbol{\Theta}_i(z)$ and $\boldsymbol{\Gamma}_i(z)$ obey the relation $\boldsymbol{\Theta}_i(z)\boldsymbol{J}\boldsymbol{\Gamma}_i(1/z^*)^* = \boldsymbol{J}$.

[3] If $\boldsymbol{R}$ is a Toeplitz matrix and $\boldsymbol{F} = \boldsymbol{A} = \boldsymbol{Z}$, then the presented scheme collapses to a now well-known algorithm of Schur [13].

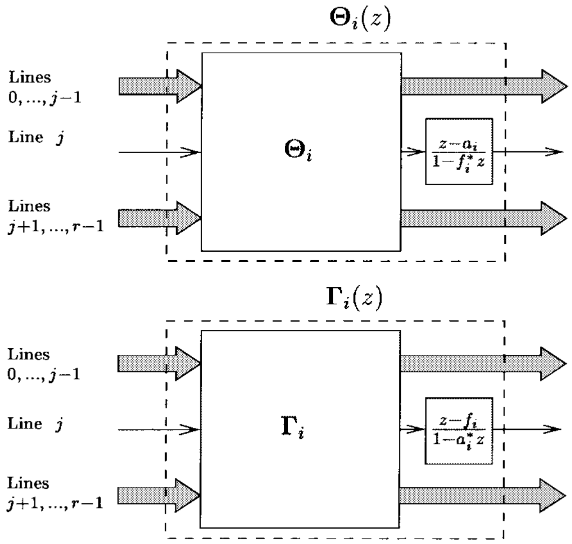


Fig. 1. Lattice filter interpretation of the ith step of Algorithm 3.1.

After n consecutive steps of Algorithm 3.1 we obtain two cascade systems

$$\boldsymbol{\Theta}(z) = \boldsymbol{\Theta}_0(z)\boldsymbol{\Theta}_1(z)\cdots\boldsymbol{\Theta}_{n-1}(z)$$

and

$$\boldsymbol{\Gamma}(z) = \boldsymbol{\Gamma}_0(z)\boldsymbol{\Gamma}_1(z)\cdots\boldsymbol{\Gamma}_{n-1}(z) \tag{25}$$

that satisfy

$$\boldsymbol{\Theta}(z)\boldsymbol{J}\boldsymbol{\Gamma}\left(\frac{1}{z^*}\right)^* = \boldsymbol{J}. \tag{26}$$

The determinants of the transfer functions $\boldsymbol{\Theta}(z)$ and $\boldsymbol{\Gamma}(z)$ can be readily expressed as

$$\det \boldsymbol{\Theta}(z) \sim \prod_{i=1}^{n} \frac{(z-f_i)}{(1-a_i^* z)}, \qquad \det \boldsymbol{\Gamma}(z) \sim \prod_{i=1}^{n} \frac{(z-a_i)}{(1-f_i^* z)} \tag{27}$$

where "$\sim$" denotes proportionality. This shows that the zero- and pole locations of $\boldsymbol{\Theta}(z)$ and $\boldsymbol{\Gamma}(z)$ are uniquely determined by the diagonal elements of $\boldsymbol{F}_i$ and $\boldsymbol{A}_i$. The next lemma follows readily from (23) and (25).

Lemma 4.1 (Polynomial Transfer Matrices): If $a_i = 0$ (respectively, $f_i = 0$) for all i then $\boldsymbol{\Theta}(z)$ [respectively, $\boldsymbol{\Gamma}(z)$] is a polynomial (rather than rational) transfer matrix. ■

The main objective of this section is to show that the cascade systems $\boldsymbol{\Theta}(z)$ and $\boldsymbol{\Gamma}(z)$ inherently satisfy certain interpolation conditions. In particular, we claim that the following statement is valid.

Proposition 4.2 (Global Interpolation Properties): Let $\boldsymbol{F}, \boldsymbol{G}$, and $\boldsymbol{J}$ be as shown in (10), and let $\boldsymbol{A}$ and $\boldsymbol{B}$ be suitable matrices so that Algorithm 3.1 terminates after the nth step. Then, $\boldsymbol{\Theta}(z)$ obtained via (23) and (25) satisfies the homogeneous interpolation conditions

$$\begin{aligned}&[\boldsymbol{v}_{k,1} \quad \boldsymbol{w}_{k,1} \quad \boldsymbol{v}_{k,2} \quad \boldsymbol{w}_{k,2} \quad \cdots \quad \boldsymbol{v}_{k,r_k} \quad \boldsymbol{w}_{k,r_k}]\\ &\qquad \cdot \boldsymbol{T}_{\boldsymbol{\Theta}}^{r_k}(\alpha_k) = \boldsymbol{0}\end{aligned} \tag{28}$$

for $k \in \{0, 1, \cdots, m-1\}$. ■

For the proof, we shall need a lemma that characterizes the local transmission properties of the ith first-order section $\boldsymbol{\Theta}_i(z)$. To this end, note that the matrices $\boldsymbol{F}_i$ obtained via the non-Hermitian generator recursion inherit the Jordan structure of $\boldsymbol{F}$. In particular, if s is the smallest index so that $i < r_0 + r_1 + \cdots + r_s$, then the eigenvalues of $\boldsymbol{F}_i$ are $\alpha_s, \alpha_{s+1}, \cdots, \alpha_{m-1}$ (with multiplicities $r_0 + r_1 + \cdots + r_s - i, r_{s+1}, r_{s+2}, \cdots, r_{m-1}$, respectively).

Lemma 4.3 (Local Interpolation Properties): Let $\boldsymbol{F}$ be given by (10) and let $\boldsymbol{\Theta}_i(z)$ be obtained at the ith step of the generalized Schur algorithm as shown in (23). Moreover, let s be the smallest index so that $i < r_0 + r_1 + \cdots + r_s$, and introduce the auxiliary indexes

$$\begin{aligned}\nu_{s+1} &= r_0 + r_1 + \cdots + r_s - i,\\ \nu_{k+1} &= \nu_k + r_k, \qquad k = s+1, s+2, \cdots, m-2.\end{aligned}$$

The first-order section $\boldsymbol{\Theta}_i(z)$ then satisfies the following (local) interpolation conditions

$$\begin{aligned}&[\boldsymbol{g}_{i,0} \quad \boldsymbol{g}_{i,1} \quad \cdots \quad \boldsymbol{g}_{i,\nu_{s+1}-1}]\boldsymbol{T}_{\boldsymbol{\Theta}_i}^{\nu_{s+1}}(\alpha_s)\\ &\qquad = [\boldsymbol{0} \quad \boldsymbol{g}_{i+1,0} \quad \boldsymbol{g}_{i+1,1} \quad \cdots \quad \boldsymbol{g}_{i+1,\nu_{s+1}-2}]\end{aligned} \tag{29}$$

$$\begin{aligned}&[\boldsymbol{g}_{i,\nu_k} \quad \boldsymbol{g}_{i,\nu_k+1} \quad \cdots \quad \boldsymbol{g}_{i,\nu_k+r_k-1}]\boldsymbol{T}_{\boldsymbol{\Theta}_i}^{r_k}(\alpha_k)\\ &\qquad = [\boldsymbol{g}_{i+1,\nu_k-1} \quad \boldsymbol{g}_{i+1,\nu_k} \quad \cdots \quad \boldsymbol{g}_{i+1,\nu_k+r_k-2}]\end{aligned} \tag{30}$$

for $k \in \{s+1, s+2, \cdots, m-1\}$ where $\boldsymbol{g}_{i,\ell}$ denotes the ℓth row of $\boldsymbol{G}_i$.

Proof of Lemma 4.3: Evaluate the consecutive rows of the generator recursion (22). The 0th row yields[4]

$$\begin{aligned}\boldsymbol{0} &= \phi_i(\alpha_s)\boldsymbol{g}_{i,0}\boldsymbol{\Theta}_i \begin{bmatrix} 0 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix} + \boldsymbol{g}_{i,0}\boldsymbol{\Theta}_i \begin{bmatrix} \boldsymbol{I}_{j-1} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & 0 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{r-j}\end{bmatrix}\\ &= \boldsymbol{g}_{i,0}\boldsymbol{\Theta}_i \begin{bmatrix} \boldsymbol{I}_{j-1} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \phi_i(\alpha_s) & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{r-j}\end{bmatrix}\\ &= \boldsymbol{g}_{i,0}\boldsymbol{\Theta}_i(\alpha_s)\end{aligned} \tag{31}$$

where we used the fact that the upper-left corner element of $\phi_i(\boldsymbol{F}_i)$ is exactly $\phi_i(\alpha_s)$.

Now let ℓ be an arbitrary index so that $0 < \ell < \nu_{s+1}$. By using the Riesz formula (8), it can be shown that the ℓth row of $\phi_i(\boldsymbol{F}_i)$ is given by

$$\left[\frac{1}{\ell!}\phi_i^{(\ell)}(\alpha_s) \quad \frac{1}{(\ell-1)!}\phi_i^{(\ell-1)}(\alpha_s) \quad \cdots \quad \phi_i(\alpha_s) \quad 0 \quad \cdots \quad 0\right].$$

Thus

$$\begin{aligned}\boldsymbol{g}_{i+1,\ell-1} &= \left[\frac{1}{\ell!}\phi_i^{(\ell)}(\alpha_s) \quad \cdots \quad \phi_i^{(1)}(\alpha_s) \quad \phi_i(\alpha_s)\right] \begin{bmatrix} \boldsymbol{g}_{i,0}\\ \boldsymbol{g}_{i,1}\\ \vdots\\ \boldsymbol{g}_{i,\ell}\end{bmatrix}\\ &\qquad \cdot \boldsymbol{\Theta}_i \begin{bmatrix} 0 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix} + \boldsymbol{g}_{i,\ell}\boldsymbol{\Theta}_i \begin{bmatrix} \boldsymbol{I}_{j-1} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & 0 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{r-j}\end{bmatrix}\\ &= \boldsymbol{g}_{i,0}\frac{1}{\ell!}\boldsymbol{\Theta}_i^{(\ell)}(\alpha_s) + \boldsymbol{g}_{i,1}\frac{1}{(\ell-1)!}\boldsymbol{\Theta}_i^{(\ell-1)}(\alpha_s) + \cdots\\ &\qquad + \boldsymbol{g}_{i,\ell}\boldsymbol{\Theta}_i(\alpha_s).\end{aligned} \tag{32}$$

[4] Physically, (31) means that the first-order section $\boldsymbol{\Theta}_i(z)$ has a transmission zero at $z = \alpha_s$, and the associated zero direction is given by $\boldsymbol{g}_{i,0}$.

Equations (31) and (32) validate (29). In a similar manner, evaluating the ℓth row of (22) for all ℓ that satisfy $\nu_k \le \ell < \nu_{k+1}$ validates (30). ∎

Proof of Proposition 4.2: Let $\bar{r}_0 = 0$, and $\bar{r}_k = r_0 + r_1 + \cdots + r_{k-1}$. If $k > 0$, then the recursive application of (30) yields

$$\begin{aligned}[\boldsymbol{g}_{0,\bar{r}_k} \quad \boldsymbol{g}_{0,\bar{r}_k+1} \quad \cdots \quad \boldsymbol{g}_{0,\bar{r}_k+r_k-1}]&\boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{\Theta}_0}(\alpha_k)\boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{\Theta}_1}(\alpha_k) \\ &\cdot \boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{\Theta}_2}(\alpha_k)\cdots\boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{\Theta}_{\bar{r}_k-1}}(\alpha_k) \\ = [\boldsymbol{g}_{1,\bar{r}_k-1} \quad \boldsymbol{g}_{1,\bar{r}_k} \quad \cdots \quad \boldsymbol{g}_{1,\bar{r}_k+r_k-2}]&\boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{\Theta}_1}(\alpha_k) \\ &\cdot \boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{\Theta}_2}(\alpha_k)\cdots\boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{\Theta}_{\bar{r}_k-1}}(\alpha_k) \\ &\vdots \\ = [\boldsymbol{g}_{\bar{r}_k,0} \quad \boldsymbol{g}_{\bar{r}_k,1} \quad \cdots \quad \boldsymbol{g}_{\bar{r}_k,r_k-1}]&. \end{aligned} \tag{33}$$

Using (29) repeatedly r_k times shows that

$$\begin{aligned}[\boldsymbol{g}_{\bar{r}_k,0} \quad \boldsymbol{g}_{\bar{r}_k,1} \quad \boldsymbol{g}_{\bar{r}_k,2} \quad \cdots \quad \boldsymbol{g}_{\bar{r}_k,r_k-1}]&\boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{\Theta}_{\bar{r}_k}}(\alpha_k)\boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{\Theta}_{\bar{r}_k+1}}(\alpha_k) \\ &\cdot \boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{\Theta}_{\bar{r}_k+2}}(\alpha_k)\cdots\boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{\Theta}_{\bar{r}_k+r_k-1}}(\alpha_k) \\ = [\mathbf{0} \quad \boldsymbol{g}_{\bar{r}_k+1,0} \quad \boldsymbol{g}_{\bar{r}_k+1,1} \quad \cdots \quad \boldsymbol{g}_{\bar{r}_k+1,r_k-2}]&\boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{\Theta}_{\bar{r}_k+1}}(\alpha_k) \\ &\cdot \boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{\Theta}_{\bar{r}_k+2}}(\alpha_k)\cdots\boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{\Theta}_{\bar{r}_k+r_k-1}}(\alpha_k) \\ &\vdots \\ = [\mathbf{0} \quad \mathbf{0} \quad \cdots \quad \mathbf{0}]&. \end{aligned} \tag{34}$$

The homomorphism property (9) of the operator $\boldsymbol{\mathcal{T}}^{r_k}$ implies that

$$\begin{aligned}[\boldsymbol{v}_{k,1} \quad \boldsymbol{w}_{k,1} \quad \boldsymbol{v}_{k,2} \quad \boldsymbol{w}_{k,2} \quad \cdots \quad \boldsymbol{v}_{k,r_k} \quad \boldsymbol{w}_{k,r_k}]&\boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{\Theta}}(\alpha_k) \\ = [\boldsymbol{g}_{0,\bar{r}_k} \quad \boldsymbol{g}_{0,\bar{r}_k+1} \quad \cdots \quad \boldsymbol{g}_{0,\bar{r}_k+r_k-1}]&\boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{\Theta}_0}(\alpha_k) \\ &\cdot \boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{\Theta}_1}(\alpha_k)\cdots\boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{\Theta}_{n-1}}(\alpha_k). \end{aligned} \tag{35}$$

Equation (28) follows immediately from (35)–(33). ∎

The interpolation properties of the dual cascade $\boldsymbol{\Gamma}(z)$ can be analyzed in a similar fashion. Here we only note that the zero structure of $\boldsymbol{\Theta}(z)$ is determined by the left null pair $\{\boldsymbol{F}, \boldsymbol{G}\}$ (as seen from Proposition 4.2) while the pole structure is determined by the right pole pair $\{\boldsymbol{A}^*, \boldsymbol{J}\boldsymbol{B}^*\}$. At the same time, it follows from (26) that $\{\boldsymbol{F}^*, \boldsymbol{J}\boldsymbol{G}^*\}$ is a right pole pair, and $\{\boldsymbol{A}, \boldsymbol{B}\}$ is a left null pair for the dual lattice $\boldsymbol{\Gamma}(z)$ (see [5, Ch. 1–3] for a thorough study of the null and pole structure of analytic and meromorphic matrix functions).

V. Parameterization of Rational Interpolants

The homogeneous interpolation properties of $\boldsymbol{\Theta}(z)$ described in Proposition 4.2 can be converted into nonhomogeneous properties by using linear fractional maps. In particular, let $\boldsymbol{\Theta}$ and $\boldsymbol{\Gamma}$ be partitioned as

$$\boldsymbol{\Theta}(z) = \begin{array}{cc} \overbrace{\phantom{\boldsymbol{\Theta}_{11}(z)}}^{p} & \overbrace{\phantom{\boldsymbol{\Theta}_{12}(z)}}^{q} \end{array} \atop \begin{bmatrix} \boldsymbol{\Theta}_{11}(z) & \boldsymbol{\Theta}_{12}(z) \\ \boldsymbol{\Theta}_{21}(z) & \boldsymbol{\Theta}_{22}(z) \end{bmatrix} \begin{array}{l} \}p \\ \}q \end{array}$$

$$\boldsymbol{\Gamma}(z) = \begin{array}{cc} \overbrace{\phantom{\boldsymbol{\Gamma}_{11}(z)}}^{p} & \overbrace{\phantom{\boldsymbol{\Gamma}_{12}(z)}}^{q} \end{array} \atop \begin{bmatrix} \boldsymbol{\Gamma}_{11}(z) & \boldsymbol{\Gamma}_{12}(z) \\ \boldsymbol{\Gamma}_{21}(z) & \boldsymbol{\Gamma}_{22}(z) \end{bmatrix} \begin{array}{l} \}p \\ \}q \end{array}.$$

The following statement is then valid.

Proposition 5.1 (Rational Parameterization): Let $\{\boldsymbol{F}, \boldsymbol{G}, \boldsymbol{J}\}$ be as shown in (10), and let $\boldsymbol{\Theta}(z)$ be obtained by executing n steps of the generalized Schur algorithm. Then, all solutions of Problem 1.1 are given by

$$\begin{aligned}\boldsymbol{Y}(z) = &-[\boldsymbol{\Theta}_{11}(z)\boldsymbol{P}(z) + \boldsymbol{\Theta}_{12}(z)\boldsymbol{Q}(z)] \\ &\cdot [\boldsymbol{\Theta}_{21}(z)\boldsymbol{P}(z) + \boldsymbol{\Theta}_{22}(z)\boldsymbol{Q}(z)]^{-1} \end{aligned} \tag{36}$$

where $\boldsymbol{P}(z) \in \mathbb{C}^{p\times q}(z)$ and $\boldsymbol{Q}(z) \in \mathbb{C}^{q\times q}(z)$ are rational parameters that are analytic at α_k and satisfy

$$\det\{\boldsymbol{\Theta}_{21}(\alpha_k)\boldsymbol{P}(\alpha_k) + \boldsymbol{\Theta}_{22}(\alpha_k)\boldsymbol{Q}(\alpha_k)\} \neq 0, \qquad \text{for all } k. \tag{37}$$

∎

The existence of such a parameterization was first proved in [4, Th. 5.1.2] by using a residual interpolation approach (see also [5, Th. 16.4.1]). The generating system $\boldsymbol{\Theta}(z)$ was not obtained recursively in that context. In what follows, we outline a rather different, inductive proof for Proposition 5.1 that relies on the recursive construction of $\boldsymbol{\Theta}(z)$.

Proof—Sufficiency: Suppose $\boldsymbol{P}(z)$ and $\boldsymbol{Q}(z)$ satisfy (37), then (36) can be rewritten as $\boldsymbol{Y}(z) = -\boldsymbol{N}(z)\boldsymbol{D}^{-1}(z)$ where

$$\begin{bmatrix} \boldsymbol{N}(z) \\ \boldsymbol{D}(z) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Theta}_{11}(z) & \boldsymbol{\Theta}_{12}(z) \\ \boldsymbol{\Theta}_{21}(z) & \boldsymbol{\Theta}_{22}(z) \end{bmatrix} \begin{bmatrix} \boldsymbol{P}(z) \\ \boldsymbol{Q}(z) \end{bmatrix}.$$

Now, recall by (9) that $\boldsymbol{\mathcal{T}}^k_{\left[\begin{smallmatrix}\boldsymbol{N}\\ \boldsymbol{D}\end{smallmatrix}\right]}(z) = \boldsymbol{\mathcal{T}}^k_{\boldsymbol{\Theta}}(z)\boldsymbol{\mathcal{T}}^k_{\left[\begin{smallmatrix}\boldsymbol{P}\\ \boldsymbol{Q}\end{smallmatrix}\right]}(z)$ is valid for all $k > 0$. Therefore, by Proposition 4.2

$$\begin{aligned}[\boldsymbol{v}_{k,1} \quad \boldsymbol{w}_{k,1} \quad \boldsymbol{v}_{k,2} \quad \boldsymbol{w}_{k,2} \quad \cdots \quad \boldsymbol{v}_{k,r_k} \quad \boldsymbol{w}_{k,r_k}]&\boldsymbol{\mathcal{T}}^{r_k}_{\left[\begin{smallmatrix}\boldsymbol{N}\\ \boldsymbol{D}\end{smallmatrix}\right]}(\alpha_k) \\ = [\boldsymbol{v}_{k,1} \quad \boldsymbol{w}_{k,1} \quad \boldsymbol{v}_{k,2} \quad \boldsymbol{w}_{k,2} \quad \cdots \quad \boldsymbol{v}_{k,r_k} \quad \boldsymbol{w}_{k,r_k}]& \\ \cdot \boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{\Theta}}(\alpha_k)\boldsymbol{\mathcal{T}}^{r_k}_{\left[\begin{smallmatrix}\boldsymbol{P}\\ \boldsymbol{Q}\end{smallmatrix}\right]}(\alpha_k) = \mathbf{0}& \end{aligned}$$

which implies that

$$\begin{aligned}[\boldsymbol{v}_{k,1} \quad \boldsymbol{v}_{k,2} \quad \cdots \quad \boldsymbol{v}_{k,r_k}]&\boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{N}}(\alpha_k) \\ + [\boldsymbol{w}_{k,1} \quad \boldsymbol{w}_{k,2} \quad \cdots \quad \boldsymbol{w}_{k,r_k}]&\boldsymbol{\mathcal{T}}^{r_k}_{\boldsymbol{D}}(\alpha_k) = \mathbf{0}. \end{aligned}$$

Thus, if $\boldsymbol{D}(\alpha_k)$ is nonsingular then

$$\begin{aligned}[\boldsymbol{v}_{k,1} \quad \boldsymbol{v}_{k,2} \quad \cdots \quad \boldsymbol{v}_{k,r_k}]&\boldsymbol{\mathcal{T}}^{r_k}_{-\boldsymbol{N}\boldsymbol{D}^{-1}}(\alpha_k) \\ = [\boldsymbol{w}_{k,1} \quad \boldsymbol{w}_{k,2} \quad \cdots \quad \boldsymbol{w}_{k,r_k}]&. \end{aligned}$$

Hence the interpolation conditions (3) are satisfied.

Necessity: Let $\boldsymbol{Y}(z)$ be a rational matrix function that satisfies the interpolation conditions (3). We must show that there exist suitable parameters $\boldsymbol{P}(z)$ and $\boldsymbol{Q}(z)$ that satisfy (36) and (37). Indeed, choose

$$\begin{bmatrix} \boldsymbol{P}(z) \\ \boldsymbol{Q}(z) \end{bmatrix} = \boldsymbol{\Theta}^{-1}(z)\begin{bmatrix} \boldsymbol{Y}(z) \\ -\boldsymbol{I}_q \end{bmatrix} = \boldsymbol{J}\boldsymbol{\Gamma}\left(\frac{1}{z^*}\right)^* \boldsymbol{J}\begin{bmatrix} \boldsymbol{Y}(z) \\ -\boldsymbol{I}_q \end{bmatrix}$$

or more explicitly

$$\begin{aligned}\boldsymbol{P}(z) &= \boldsymbol{\Gamma}_{11}\left(\frac{1}{z^*}\right)^*\boldsymbol{Y}(z) + \boldsymbol{\Gamma}_{21}\left(\frac{1}{z^*}\right)^* \\ \boldsymbol{Q}(z) &= -\boldsymbol{\Gamma}_{12}\left(\frac{1}{z^*}\right)\boldsymbol{Y}(z)^* - \boldsymbol{\Gamma}_{22}\left(\frac{1}{z^*}\right)^*. \end{aligned}$$

Then

$$\boldsymbol{\Theta}(z)\begin{bmatrix}\boldsymbol{P}(z)\\ \boldsymbol{Q}(z)\end{bmatrix} = \boldsymbol{\Theta}(z)\boldsymbol{\Theta}^{-1}(z)\begin{bmatrix}\boldsymbol{Y}(z)\\ -\boldsymbol{I}_q\end{bmatrix} = \begin{bmatrix}\boldsymbol{Y}(z)\\ -\boldsymbol{I}_q\end{bmatrix} \tag{38}$$

and therefore (36) is satisfied. Equation (38) further implies that

$$\boldsymbol{\Theta}_{21}(z)\boldsymbol{P}(z) + \boldsymbol{\Theta}_{22}(z)\boldsymbol{Q}(z) = -\boldsymbol{I}_q$$

thus (37) holds. It only remains to prove the analyticity of

$$\begin{bmatrix}\boldsymbol{P}(z)\\ \boldsymbol{Q}(z)\end{bmatrix} = \boldsymbol{\Theta}_{n-1}^{-1}(z)\boldsymbol{\Theta}_{n-2}^{-1}(z)\cdots\boldsymbol{\Theta}_0^{-1}(z)\begin{bmatrix}\boldsymbol{Y}(z)\\ -\boldsymbol{I}_q\end{bmatrix}$$

at $f_0, f_1, \cdots, f_{n-1}$.

Let us proceed by induction. By definition, $[\boldsymbol{Y}^T(z) \quad -\boldsymbol{I}_q]^T$ is analytic at $z = f_i$ for all i. The inductive assumption is that

$$\begin{bmatrix}\boldsymbol{A}(z)\\ \boldsymbol{B}(z)\end{bmatrix} \stackrel{\text{def}}{=} \boldsymbol{\Theta}_{i-1}^{-1}(z)\cdots\boldsymbol{\Theta}_0^{-1}(z)\begin{bmatrix}\boldsymbol{Y}(z)\\ -\boldsymbol{I}_q\end{bmatrix} \tag{39}$$

is analytic at $z \in \{f_0, f_1, \cdots, f_{i-1}\}$. Now one must show that

$$\boldsymbol{\Theta}_i^{-1}(z)\begin{bmatrix}\boldsymbol{A}(z)\\ \boldsymbol{B}(z)\end{bmatrix} = \boldsymbol{\Theta}_i^{-1}(z)\boldsymbol{\Theta}_{i-1}^{-1}(z)\cdots\boldsymbol{\Theta}_0^{-1}(z)\begin{bmatrix}\boldsymbol{Y}(z)\\ -\boldsymbol{I}_q\end{bmatrix} \tag{40}$$

is analytic at $z = f_i$. The inductive assumption along with the expression

$$\boldsymbol{\Theta}_i^{-1}(z) = \begin{bmatrix}\boldsymbol{I}_{j-1} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \dfrac{1-a_i^* z}{z-\alpha_i} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \boldsymbol{I}_{r-j}\end{bmatrix}\boldsymbol{\Theta}_i^{-1}$$

implies that each row of (40) is analytic at f_i with the possible exception of the jth row. In order to prove the analyticity of the jth row observe that

$$\begin{aligned}
&[0 \quad \cdots \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0]\begin{bmatrix}\boldsymbol{I}_{j-1} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \dfrac{1-a_i^* z}{z-f_i} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \boldsymbol{I}_{r-j}\end{bmatrix}\\
&\quad\cdot \boldsymbol{\Theta}_i^{-1}\begin{bmatrix}\boldsymbol{A}(z)\\ \boldsymbol{B}(z)\end{bmatrix}\\
&= \frac{1-a_i^* z}{z-f_i}[0 \quad \cdots \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0]\boldsymbol{\Theta}_i^{-1}\begin{bmatrix}\boldsymbol{A}(z)\\ \boldsymbol{B}(z)\end{bmatrix}\\
&\sim \frac{1-a_i^* z}{z-f_i}\boldsymbol{g}_{i,0}\begin{bmatrix}\boldsymbol{A}(z)\\ \boldsymbol{B}(z)\end{bmatrix}
\end{aligned} \tag{41}$$

where (41) follows from the fact that $\boldsymbol{\Theta}_i$ transforms $\boldsymbol{g}_{i,0}$ into proper form. The proof is concluded by showing that

$$\boldsymbol{g}_{i,0}\begin{bmatrix}\boldsymbol{A}(f_i)\\ \boldsymbol{B}(f_i)\end{bmatrix} = \mathbf{0}. \tag{42}$$

Indeed, let $\overline{r}_k = r_0 + r_1 + \cdots + r_{k-1}$ be the cumulative sum of the indexes r_k, and assume that $\overline{r}_k \le i \le \overline{r}_k + r_k - 1$ for some k. Then (20) implies that $f_i = \alpha_k$, and

$$\begin{aligned}
&[\mathbf{0} \quad \mathbf{0} \quad \cdots \quad \mathbf{0}]\\
&\quad = [\boldsymbol{v}_{k,0} \quad \boldsymbol{w}_{k,0} \quad \boldsymbol{v}_{k,1} \quad \boldsymbol{w}_{k,1} \quad \cdots \quad \boldsymbol{v}_{k,r_k-1} \quad \boldsymbol{w}_{k,r_k-1}]\\
&\qquad \cdot \boldsymbol{T}^{r_k}_{\left[\begin{smallmatrix}\boldsymbol{Y}\\ -\boldsymbol{I}_q\end{smallmatrix}\right]}(\alpha_k) && (43)\\
&\quad = [\boldsymbol{g}_{0,\overline{r}_k} \quad \boldsymbol{g}_{0,\overline{r}_k+1} \quad \cdots \quad \boldsymbol{g}_{0,\overline{r}_k+r_k-1}]\\
&\qquad \cdot \boldsymbol{T}^{r_k}_{\boldsymbol{\Theta}_0\cdots\boldsymbol{\Theta}_{k-1}\left[\begin{smallmatrix}\boldsymbol{A}\\ \boldsymbol{B}\end{smallmatrix}\right]}(\alpha_k) && (44)\\
&\quad = [\boldsymbol{g}_{0,\overline{r}_k} \quad \boldsymbol{g}_{0,\overline{r}_k+1} \quad \cdots \quad \boldsymbol{g}_{0,\overline{r}_k+r_k-1}]\boldsymbol{T}^{r_k}_{\boldsymbol{\Theta}_0\cdots\boldsymbol{\Theta}_{k-1}}(\alpha_k)\\
&\qquad \cdot \boldsymbol{T}^{r_k}_{\left[\begin{smallmatrix}\boldsymbol{A}\\ \boldsymbol{B}\end{smallmatrix}\right]}(\alpha_k) && (45)\\
&\quad = [\mathbf{0} \quad \cdots \quad \boldsymbol{g}_{i,0} \quad \cdots \quad \boldsymbol{g}_{i,\overline{r}_k+r_k-i}]\boldsymbol{T}^{r_k}_{\left[\begin{smallmatrix}\boldsymbol{A}\\ \boldsymbol{B}\end{smallmatrix}\right]}(\alpha_k) && (46)
\end{aligned}$$

where (43) follows from the fact that $\boldsymbol{Y}(z)$ satisfies the interpolation conditions (3), (44) follows from (39), (45) follows from the homomorphism property (9), and (46) can be verified by applying Lemma 4.3 recursively i times. Equation (46), in turn, implies (42). ∎

The parameters $\boldsymbol{P}(z)$ and $\boldsymbol{Q}(z)$ in Proposition 5.1 are not unique. Indeed, let $\boldsymbol{R}(z) \in \mathbb{C}^{q\times q}(z)$ be any rational matrix function analytic at α_k so that $\det \boldsymbol{R}(\alpha_k) \neq 0$. If the parameters $\boldsymbol{P}(z)$ and $\boldsymbol{Q}(z)$ satisfy (37) and correspond to an interpolant $\boldsymbol{Y}(z)$, then the parameters $\boldsymbol{P}(z)\boldsymbol{R}(z)$ and $\boldsymbol{Q}(z)\boldsymbol{R}(z)$ also satisfy (37) and correspond to the same interpolant. It is possible to obtain a slightly different characterization for the family of interpolating functions in terms of *right coprime polynomial* parameters which are *unique* up to a unimodular factor.

Proposition 5.2 (Polynomial Parameterization): Let $\{\boldsymbol{F},\boldsymbol{G},\boldsymbol{J}\}$ be given by (10), and let $\boldsymbol{\Theta}(z)$ be obtained after executing n steps of the generalized Schur algorithm. Then all solutions of Problem 1.1 are given by

$$\begin{aligned}\boldsymbol{Y}(z) = &-[\boldsymbol{\Theta}_{11}(z)\boldsymbol{P}_R(z) + \boldsymbol{\Theta}_{12}(z)\boldsymbol{Q}_R(z)]\\ &\cdot [\boldsymbol{\Theta}_{21}(z)\boldsymbol{P}_R(z) + \boldsymbol{\Theta}_{22}(z)\boldsymbol{Q}_R(z)]^{-1}\end{aligned} \tag{47}$$

where $\boldsymbol{P}_R(z) \in \mathbb{C}^{p\times q}[z]$ and $\boldsymbol{Q}_R(z) \in \mathbb{C}^{q\times q}[z]$ are right coprime polynomial matrices chosen so as to satisfy

$$\det\{\boldsymbol{\Theta}_{21}(\alpha_k)\boldsymbol{P}_R(\alpha_k) + \boldsymbol{\Theta}_{22}(\alpha_k)\boldsymbol{Q}_R(\alpha_k)\} \neq 0, \quad \text{for all } k. \tag{48}$$

The parameters $\boldsymbol{P}_R(z)$ and $\boldsymbol{Q}_R(z)$ which correspond to a particular interpolating function $\boldsymbol{Y}(z)$ are unique up to a unimodular right factor. ∎

Proof: A simple algebraic proof can be found in [21, pp. 127–128]. ∎

So far, we did not make any particular assumption about the pole structure of the transfer matrix $\boldsymbol{\Theta}(z)$. If we further assume that $\boldsymbol{\Theta}(z)$ is a polynomial matrix (i.e., all of its poles are at infinity), then the linear fractional parameterization formula (47) gives *matrix fraction descriptions* for all rational interpolants $\boldsymbol{Y}(z)$.

Corollary 5.3 (Matrix Fraction Description): Let $\{\boldsymbol{F},\boldsymbol{G},\boldsymbol{J}\}$ be given by (10) and let $\boldsymbol{\Theta}(z)$ be obtained after executing n steps of the generalized Schur algorithm. Assume, furthermore, that $a_i = 0$ for all i. Then $\boldsymbol{\Theta}(z)$ is a polynomial matrix and all solutions of Problem 1.1 can be obtained as $\boldsymbol{Y}(z) =$

$\boldsymbol{N}_R(z)\boldsymbol{D}_R^{-1}(z)$ where the polynomial matrices $\boldsymbol{N}_R(z)$ and $\boldsymbol{D}_R(z)$ are given by

$$\begin{bmatrix}\boldsymbol{N}_R(z)\\ \boldsymbol{D}_R(z)\end{bmatrix}=\begin{bmatrix}\boldsymbol{\Theta}_{11}(z) & \boldsymbol{\Theta}_{12}(z)\\ \boldsymbol{\Theta}_{21}(z) & \boldsymbol{\Theta}_{22}(z)\end{bmatrix}\begin{bmatrix}\boldsymbol{P}_R(z)\\ \boldsymbol{Q}_R(z)\end{bmatrix} \tag{49}$$

for some right coprime polynomial parameters $\boldsymbol{P}_R(z)$ and $\boldsymbol{Q}_R(z)$ that satisfy (48). The coprimeness of $\boldsymbol{P}_R(z)$ and $\boldsymbol{Q}_R(z)$ implies the coprimeness of $\boldsymbol{N}_R(z)$ and $\boldsymbol{D}_R(z)$, and vice-versa. The pair $\{\boldsymbol{N}_R(z), \boldsymbol{D}_R(z)\}$ provides a matrix fraction description for the rational interpolant $\boldsymbol{Y}(z)$. ■

Proof: Lemma 4.1 implies that $\boldsymbol{\Theta}(z)$ is a polynomial matrix whenever $a_i = 0$ for all i. The rest of the statement follows from Proposition 5.2 if we take into account that $\boldsymbol{\Theta}(z)$ is a polynomial matrix. We only need to prove the claim about coprimeness.

Two polynomial matrices are right coprime if and only if, they do not have a common right eigenvector. It follows from (49) that if $\boldsymbol{P}_R(z)$ and $\boldsymbol{Q}_R(z)$ have a common right eigenvector then $\boldsymbol{N}_R(z)$ and $\boldsymbol{D}_R(z)$ also have a common right eigenvector. Thus, the right coprimeness of $\boldsymbol{N}_R(z)$ and $\boldsymbol{D}_R(z)$ implies the right coprimeness of $\boldsymbol{Q}_R(z)$ and $\boldsymbol{P}_R(z)$.

Conversely, assume that $\boldsymbol{P}_R$ and $\boldsymbol{Q}_R$ do not have a common right eigenvector. Since $\boldsymbol{\Theta}(z)$ is invertible on $z \in \mathbb{C}/\{\alpha_0, \alpha_1, \cdots, \alpha_{m-1}\}, \boldsymbol{N}_R(z)$ and $\boldsymbol{D}_R(z)$ cannot have a common right eigenvector over this set. On the other hand, by (48), $\boldsymbol{D}_R(z)$ is invertible over $\{\alpha_0, \alpha_1, \cdots, \alpha_{m-1}\}$. Therefore, it cannot have a right eigenvector over this set either. It follows that $\boldsymbol{N}_R(z)$ and $\boldsymbol{D}_R(z)$ are right coprime. ■

VI. Solution of Problem 1.1

In a triangular factorization problem, the matrix $\boldsymbol{R}$ is given in advance and the main goal is to find lower triangular arrays $\boldsymbol{F}$ and $\boldsymbol{A}$ such that the displacement rank $r = \operatorname{rank}\{\boldsymbol{R} - \boldsymbol{FRA}^*\}$ is as small as possible. If $\boldsymbol{R}$ is strongly regular, its triangular factors can be computed efficiently by using the generalized Schur algorithm. In connection with interpolation problems the situation is somewhat different. The triple $\{\boldsymbol{F}, \boldsymbol{G}, \boldsymbol{J}\}$ is constructed first from the interpolation data as shown in (10). Next, the generator recursion is used to obtain a rational matrix function $\boldsymbol{\Theta}(z)$ that makes it possible to parameterize the solution set. In fact, it is not necessary to compute $\boldsymbol{R}$ explicitly in order to proceed with the recursion. The bottom line of the above discussions is that the generalized Schur algorithm provides a fast recursive method for solving Problem 1.1.

Algorithm 6.1 (Main Algorithm)

Stage I: From the interpolation data, form the arrays $\boldsymbol{J}, \boldsymbol{F}$, and $\boldsymbol{G}$ as shown in (10). Set $\boldsymbol{G}_0 = \boldsymbol{G}$, and $\boldsymbol{F}_0 = \boldsymbol{F}$. Repeat the following steps for $i = 0, 1, \cdots, n-1$.

Step 1: Let $\boldsymbol{g}_{i,0} = [\boldsymbol{g}_{i,0}^a \quad \boldsymbol{g}_{i,0}^b]$ where $\boldsymbol{g}_{i,0}^a$ denotes the first p elements and $\boldsymbol{g}_{i,0}^b$ denotes the last q elements of $\boldsymbol{g}_{i,0}$.

- If $\boldsymbol{g}_{i,0}^a = \boldsymbol{0}$, but $\boldsymbol{g}_{i,0}^b \neq \boldsymbol{0}$, then the interpolation data is contradictory.
- If both $\boldsymbol{g}_{i,0}^a = \boldsymbol{0}$ and $\boldsymbol{g}_{i,0}^b = \boldsymbol{0}$ then the interpolation data is redundant. Set $\boldsymbol{\Theta}_i = \boldsymbol{I}_r$, and go to Step 3.

Step 2: Choose a nonsingular matrix $\boldsymbol{\Theta}_i$ which transforms $\boldsymbol{g}_{i,0}$ into proper form with a single nonzero entry in one of the first p positions:

$$\begin{aligned}\boldsymbol{g}_{i,0}\boldsymbol{\Theta}_i &= [\bar{\boldsymbol{g}}_{i,0}^a \quad \boldsymbol{0}] \quad \text{where}\\ \bar{\boldsymbol{g}}_{i,0}^a &= [0 \quad \cdots \quad 0 \quad * \quad 0 \quad \cdots \quad 0].\end{aligned} \tag{50}$$

Step 3: Choose a_i such that $a_i \neq 1/f_j^*$ and $a_i \neq 1/\xi_\ell^*$ for all ℓ.

Step 4: Obtain $\boldsymbol{\Theta}_i(z)$ as

$$\boldsymbol{\Theta}_i(z) = \boldsymbol{\Theta}_i \begin{bmatrix}\boldsymbol{I}_{j-1} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \dfrac{z-f_i}{1-a_i^* z} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{r-j}\end{bmatrix}.$$

Step 5: Update the generator $\boldsymbol{G}_i$ as

$$\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{G}_{i+1}\end{bmatrix} = \boldsymbol{\Phi}_i\boldsymbol{G}_i\boldsymbol{\Theta}_i\begin{bmatrix}\boldsymbol{0}_{j-1} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & 1 & \boldsymbol{0}\\ \boldsymbol{0} & & \boldsymbol{0}_{r-j}\end{bmatrix} + \boldsymbol{G}_i\boldsymbol{\Theta}_i\begin{bmatrix}\boldsymbol{I}_{j-1} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & 0 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{r-j}\end{bmatrix}$$

where $\boldsymbol{\Phi}_i = (\boldsymbol{F}_i - f_i\boldsymbol{I}_{n-i})(\boldsymbol{I}_{n-i} - a_i^*\boldsymbol{F}_i)^{-1}$.

Stage II: Compute $\boldsymbol{\Theta}(\xi_\ell) = \boldsymbol{\Theta}_0(\xi_\ell)\boldsymbol{\Theta}_1(\xi_\ell)\cdots\boldsymbol{\Theta}_{n-1}(\xi_\ell)$ and evaluate $\boldsymbol{Y}(\xi_\ell) = \boldsymbol{\Theta}_{12}(\xi_\ell)\boldsymbol{\Theta}_{22}^{-1}(\xi_\ell)$ for all ℓ. ■

Remarks:

1) The computational burden of *Stage I* is $\mathcal{O}(rn^2)$ flops. The computational burden of *Stage II* is $\mathcal{O}(nr^2)$ flops for each extraction point ξ_ℓ.
2) *Stage I* of the algorithm can be looked at as a preprocessing step. The generating system is synthesized as the product of elementary first-order sections. This form makes it possible to evaluate any rational interpolant at any given point.
3) There are no break-downs in the algorithm. There is no need to use higher order sections (look-ahead steps).
4) It is not necessary to choose the arrays $\boldsymbol{A}$ and $\boldsymbol{B}$ in advance. It is enough to "*dynamically*" select the scalar a_i at the moment when it is needed in the algorithm.
5) According to (50) the pivot elements are always chosen from the first p positions of $\boldsymbol{g}_i$. This ensures that $\det\boldsymbol{\Theta}_{22} \neq 0$, so that $\boldsymbol{P}_R(z) = \boldsymbol{0}$ and $\boldsymbol{Q}_R(z) = \boldsymbol{I}_q$ satisfy (37). Thus $\boldsymbol{Y}(z) = \boldsymbol{\Theta}_{12}(z)\boldsymbol{\Theta}_{22}^{-1}(z)$ gives a particular solution to the unconstrained interpolation problem.
6) The additional degrees of freedom in $\boldsymbol{\Theta}_i$ and a_i can be used to improve numerical accuracy or to impose additional constraints on the interpolants. Some particular examples include:
   a) $\boldsymbol{\Theta}_i$ can be a unitary matrix (implemented as a sequence of elementary Givens rotations and Householder projections). This choice might be useful from a numerical point of view. In this case $\boldsymbol{\Theta}(z)$ is obtained as the product of simple unitary and diagonal factors.

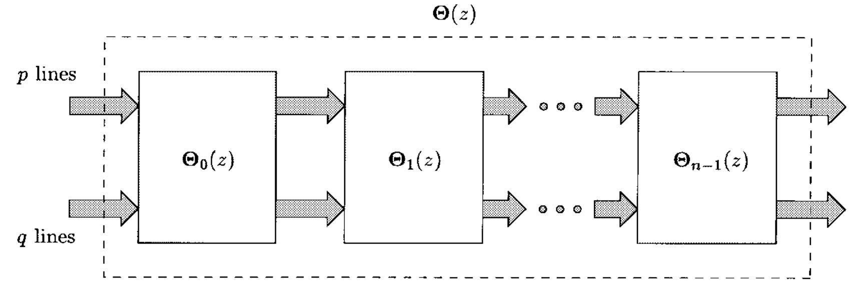


Fig. 2. The feedforward lattice.

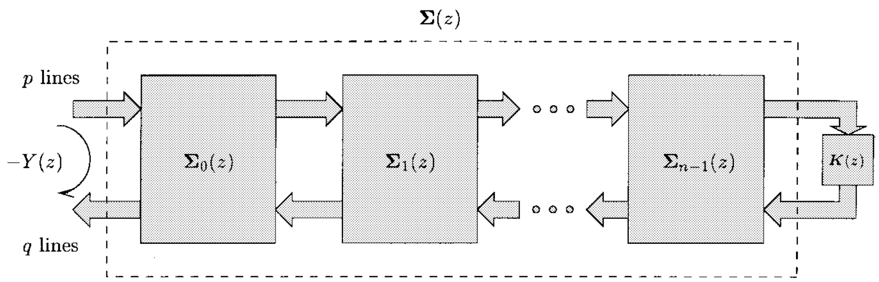


Fig. 3. The feedback lattice.

b) $\boldsymbol{\Theta}_i$ can be chosen in upper triangular form for all i. In this case $\boldsymbol{\Theta}(z)$ is upper triangular, and Propositions 5.1 and 5.2 provide an *affine parameterization* for all rational interpolants (see Section VIII).
c) If $a_i = 0$ for all i then $\boldsymbol{\Theta}(z)$ is a polynomial matrix. This feature can be used to solve polynomial interpolation problems such as the general Hermite problem (see Section VIII) or the minimal interpolation problem (see Section IX). In general, a_i can be used to preassign the poles of $\boldsymbol{\Theta}(z)$.
d) Suppose that the interpolation points α_k lie in the interior of the open unit disc. Under this assumption, the special choice $a_i = f_i$ and a $\boldsymbol{J}$-unitary $\boldsymbol{\Theta}_i$ (satisfying $\boldsymbol{\Theta}_i \boldsymbol{J} \boldsymbol{\Theta}_i^* = \boldsymbol{J}$) lead to a $\boldsymbol{J}$-lossless generating system and to a Schur-type interpolant (analytic and uniformly bounded in the interior of the open unit disc). Such a choice for $\boldsymbol{\Theta}_i$ is possible if and only if, $\boldsymbol{g}_i \boldsymbol{J} \boldsymbol{g}_i^* < 0$ (see [5], [6], and [9] for such constrained interpolation problems).

VII. Physical Interpretation

Algorithm 6.1 corresponds to the synthesis of a p-input q-output feed-forward cascade system with $n = r_0 + r_1 + \cdots + r_{m-1}$ consecutive sections (see Fig. 2). The transfer function $\boldsymbol{\Theta}(z)$ of the feed-forward cascade satisfies the homogeneous interpolation conditions of Proposition 4.2. The linear fractional map of Proposition 5.1 can be interpreted physically by reversing the signal flow of the bottom q lines of the feed-forward lattice and by attaching a load[5] $\boldsymbol{K}(z) = -\boldsymbol{P}_R(z)\boldsymbol{Q}_R^{-1}(z)$ to the right-hand side (see Fig. 3). The I/O description of the first-order sections in the feedback lattice are given by

$$\boldsymbol{\Sigma}_i(z) = \begin{bmatrix} \boldsymbol{\Theta}_{i,11} - \boldsymbol{\Theta}_{i,12}\boldsymbol{\Theta}_{i,22}^{-1}\boldsymbol{\Theta}_{i,21} & -\boldsymbol{\Theta}_{i,12}\boldsymbol{\Theta}_{i,22}^{-1} \\ \boldsymbol{\Theta}_{i,22}^{-1}\boldsymbol{\Theta}_{i,21} & \boldsymbol{\Theta}_{i,22}^{-1} \end{bmatrix}, \quad i \in \{0, 1, \cdots, n-1\}.$$

In circuit theory, $\boldsymbol{\Sigma}_i(z)$ is called the *scattering matrix* while $\boldsymbol{\Theta}_i(z)$ is called the *chain scattering matrix* associated with the ith section. The global transfer function of the feedback lattice is given by

$$\begin{aligned}\boldsymbol{\Sigma}(z) &= \boldsymbol{\Sigma}_0(z) \star \boldsymbol{\Sigma}_1(z) \star \cdots \star \boldsymbol{\Sigma}_{n-1}(z) \\ &= \begin{bmatrix} \boldsymbol{\Theta}_{11} - \boldsymbol{\Theta}_{12}\boldsymbol{\Theta}_{22}^{-1}\boldsymbol{\Theta}_{21} & -\boldsymbol{\Theta}_{12}\boldsymbol{\Theta}_{22}^{-1} \\ \boldsymbol{\Theta}_{22}^{-1}\boldsymbol{\Theta}_{21} & \boldsymbol{\Theta}_{22}^{-1} \end{bmatrix}\end{aligned}$$

where $\star$ denotes the so-called *Redheffer star-product*. Now, the rational interpolant $\boldsymbol{Y}(z)$ is obtained as the negative transfer function from the top-left input lines to the bottom-left output lines of the scattering cascade when the right-hand side is attached to the load $\boldsymbol{K}(z)$.

In fact $\boldsymbol{Y}(z)$ is constructed recursively via the generalized Schur algorithm by attaching additional sections to the feedback cascade. The new sections are introduced in such a way that they do not interfere with the interpolation properties of the preceding sections. The closed-loop transfer function $-\boldsymbol{Y}(z)$ satisfies the prescribed interpolation conditions independently of the load $\boldsymbol{K}(z)$ (as long as $\boldsymbol{K}(z)$ is analytic at the interpolation points).

There exists a strong analogy between the scattering representation of the generalized Schur algorithm and the discretized version of a physical transmission line; this analogy shows that the generalized Schur algorithm nicely solves many inverse scattering problems (see, e.g., [22] and [23]).

[5] If $\boldsymbol{Q}_R(z)$ is not invertible, the load can be described by the implicit relation $\boldsymbol{y}(z)\boldsymbol{Q}_R(z) + \boldsymbol{u}(z)\boldsymbol{P}_R(z) = \boldsymbol{0}$, where $\boldsymbol{u}(z)$ and $\boldsymbol{y}(z)$ denote the input and output variables, respectively.

VIII. Polynomial Interpolation

The primary goal in this section is to parameterize the *polynomial* solutions of Problem 1.1. In order to accomplish this task, we need to first obtain a polynomial generating system in *upper triangular* form.

Proposition 8.1: In Step 3 of Algorithm 6.1, choose $a_i = 0$. Moreover, construct $\boldsymbol{\Theta}_i$ in upper triangular form. This can be done by choosing the left-most nonzero element in $\boldsymbol{g}_{i,0}$ as a pivot element. The above choices lead to

$$\boldsymbol{\Theta}(z) = \begin{bmatrix} \boldsymbol{\Pi}(z) & -\boldsymbol{H}(z) \\ \mathbf{0} & \boldsymbol{I}_q \end{bmatrix} \tag{51}$$

where $\boldsymbol{\Pi}(z) \in \mathbb{C}^{p\times p}[z]$ and $\boldsymbol{H}(z) \in \mathbb{C}^{p\times q}[z]$ are polynomial matrices such that

$$\det \boldsymbol{\Pi}(z) \sim \prod_{i=0}^{m-1} (z-\alpha_k)^{r_k} \quad \text{and} \quad \delta\{\boldsymbol{H}(z)\} \le n-1. \tag{52}$$

■

Proof: Since $a_i = 0$ for all i, it follows from Lemma 4.1 that $\boldsymbol{\Theta}(z)$ is a polynomial matrix. If we choose the left-most nonzero element in $\boldsymbol{g}_i$ as a pivot element, then $\boldsymbol{\Theta}_i$ can be constructed in the following form:

$$\begin{array}{l} [0 \quad \cdots \quad 0 \quad \overset{j}{*} \quad * \quad \cdots \quad *] \\ \cdot \begin{array}{c} \ j \\ \begin{bmatrix} 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & * & \cdots & * \\ 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \end{array} \\ = [0 \quad \cdots \quad 0 \quad \overset{j}{*} \quad 0 \quad \cdots \quad 0] \end{array}$$

where the "$*$" 's denote nontrivial entries. In this case, the first-order factors $\boldsymbol{\Theta}_i(z)$, as well as the transfer function $\boldsymbol{\Theta}(z)$, are upper triangular. Since the pivot elements are chosen from the first p positions, the relation $\boldsymbol{\Theta}_{22}(z) = \boldsymbol{I}_q$ must hold. The statement about the determinant of $\boldsymbol{\Pi}(z)$ follows from (27). Note that the last step does not increase the McMillan degree of $\boldsymbol{H}(z)$. ■

Corollary 8.2 (Affine Parameterization): Assume that $\boldsymbol{\Theta}(z)$ is an upper triangular polynomial matrix as shown in (51). Then all solutions of Problem 1.1 are given by

$$\boldsymbol{Y}(z) = \boldsymbol{H}(z) + \boldsymbol{\Pi}(z)\boldsymbol{P}_R(z)\boldsymbol{Q}_R^{-1}(z) \tag{53}$$

where $\boldsymbol{P}_R(z) \in \mathbb{C}^{p\times q}[z]$ and $\boldsymbol{Q}_R(z) \in \mathbb{C}^{q\times q}[z]$ are right coprime polynomial parameters such that $\det \boldsymbol{Q}_R(\alpha_k) \ne \mathbf{0}$. In particular, $\boldsymbol{H}(z)$ is a polynomial solution of Problem 1.1.

■

Let $\boldsymbol{K}(z) = \boldsymbol{P}_R(z)\boldsymbol{Q}_R^{-1}(z)$ denote the rational parameter that appears in (53). It is evident that (53) establishes a one-to-one correspondence between the set of rational interpolants and the set of rational matrix functions that are analytic at α_k. Moreover, the affine map (53) preserves analyticity, *viz.*, $\boldsymbol{Y}(z)$ is analytic at $z = z_0$ if and only if, $\boldsymbol{K}(z)$ is analytic at $z = z_0$. This feature can be used to parameterize all rational interpolants that are "*stable*" in a certain region.

Corollary 8.3: All solutions of Problem 1.1 that are analytic in a prescribed region Ω are given by

$$\boldsymbol{Y}(z) = \boldsymbol{H}(z) + \boldsymbol{\Pi}(z)\boldsymbol{K}(z) \tag{54}$$

where $\boldsymbol{K}(z) \in \mathbb{C}^{p\times q}(z)$ is some rational parameter that is analytic in $\Omega \cup \{\alpha_0, \alpha_1, \cdots, \alpha_{m-1}\}$. ■

In particular, note that $\boldsymbol{Y}(z)$ in (54) is a polynomial interpolant if and only if, $\boldsymbol{K}(z)$ is a polynomial parameter. Hence, the polynomial solutions of Problem 1.1 can be readily parameterized.

Example: Let $\mathbb{C}_d[z]$ denote the space of scalar polynomials with degree not exceeding d, and let $\mathbb{C}_d^{p\times q}[z]$ denote the space of $p \times q$ matrices whose elements belong to $\mathbb{C}_d[z]$.

Problem 8.4 (Hermite Matrix Interpolation): Let $\{\alpha_k\}_{k=0}^{M-1}$ be a set of distinct points. With each point α_k, associate L_k constant matrices $\{\boldsymbol{\beta}_{kl}\}_{l=0}^{L_k-1} \in \mathbb{C}^{p\times q}$. Find a polynomial matrix $\boldsymbol{\mathcal{H}}(z) \in \mathbb{C}_{L-1}^{p\times q}[z](L = L_0 + L_1 + \cdots + L_{M-1})$ that satisfies

$$\boldsymbol{\mathcal{H}}^{(l)}(\alpha_k) = \boldsymbol{\beta}_{kl}. \tag{55}$$

Problem 8.4 can be recast into a tangential framework by introducing

$$\boldsymbol{\mathcal{V}}_k = [\boldsymbol{I}_p \quad \mathbf{0}_{p\times p} \quad \cdots \quad \mathbf{0}_{p\times p}]$$

and

$$\boldsymbol{\mathcal{W}}_k = [\boldsymbol{\beta}_{k,0} \quad \boldsymbol{\beta}_{k,1} \quad \cdots \quad \boldsymbol{\beta}_{k,L_k-1}]$$

where both $\boldsymbol{\mathcal{V}}_k$ and $\boldsymbol{\mathcal{W}}_k$ contain exactly L_k blocks. Now, (55) can be written as

$$\boldsymbol{\mathcal{V}}_k \boldsymbol{\mathcal{T}}_{\boldsymbol{\mathcal{H}}}^{L_k}(\alpha_k) = \boldsymbol{\mathcal{W}}_k.$$

Equation (55) imposes pqL independent constraints on $\boldsymbol{\mathcal{H}}(z)$. Thus, (55) must have a *unique* solution in the pqL dimensional space $\mathbb{C}_{L-1}^{p\times q}[z]$. This solution is called the *Hermite matrix polynomial*. The following special cases often occur in the literature [24].

- *Simple Hermite problem* (or osculatory interpolation): $L_0 = L_1 = \cdots = L_{k-1} = 2$.
- *Lagrange interpolation:* $L_0 = L_1 = \cdots = L_{k-1} = 1$ (with the Lagrange matrix polynomial as a unique solution).
- *Taylor interpolation:* $M = 1$ $(L_0 \ge 1)$ (with the Taylor matrix polynomial as a unique solution).

Solution of Problem 8.4: Construct the arrays $\{\boldsymbol{F}, \boldsymbol{G}, \boldsymbol{J}\}$ as shown in (10). In this special case Proposition 8.1 gives rise to an upper triangular generating system of the form

$$\boldsymbol{\Theta}(z) = \begin{bmatrix} \pi(z)\cdot \boldsymbol{I}_p & -\boldsymbol{\mathcal{H}}(z) \\ \mathbf{0} & \boldsymbol{I}_q \end{bmatrix}. \tag{56}$$

Due to the special structure of $\boldsymbol{\mathcal{V}}_k$, the (1, 1) entry of (56) can be written as $\boldsymbol{\Pi}(z) = \pi(z) \cdot \boldsymbol{I}_p$ where $\pi(z) = \Pi_{k=0}^{M-1}(z-\alpha_k)^{L_k}$. The pivot elements are chosen exactly L times from each of the first, second, $\cdots p$th positions, and therefore $\boldsymbol{\mathcal{H}}(z)$ cannot have an element whose degree is larger than $L-1$. Thus, by uniqueness, $\boldsymbol{\mathcal{H}}(z)$ must be the Hermite matrix polynomial. ■

In the Appendix two examples are presented for the scalar Hermite and the matrix Taylor problems. In [15] we thoroughly discussed the scalar Lagrange problem.

IX. Minimal Tangential Interpolation

Assume that $\boldsymbol{\Theta}(z)$ is a polynomial matrix. By definition, the McMillan degree of the interpolating function $\boldsymbol{Y}(z) = \boldsymbol{N}_R(z)\boldsymbol{D}_R^{-1}(z)$ is equal to the McMillan degree of the polynomial matrix

$$\begin{bmatrix}\boldsymbol{N}_R(z)\\ \boldsymbol{D}_R(z)\end{bmatrix} = \begin{bmatrix}\boldsymbol{\Theta}_{11}(z) & \boldsymbol{\Theta}_{12}(z)\\ \boldsymbol{\Theta}_{21}(z) & \boldsymbol{\Theta}_{22}(z)\end{bmatrix}\begin{bmatrix}\boldsymbol{P}_R(z)\\ \boldsymbol{Q}_R(z)\end{bmatrix}.$$

In connection with Problem 1.2 one has to minimize the McMillan degree of $[\boldsymbol{N}_R^T(z) \quad \boldsymbol{D}_R^T(z)]^T$ over all possible polynomial matrices $\boldsymbol{P}_R(z)$ and $\boldsymbol{Q}_R(z)$ under the constraint $\det\{\boldsymbol{\Theta}_{21}(\alpha_k)\boldsymbol{P}(\alpha_k) + \boldsymbol{\Theta}_{22}(\alpha_k)\boldsymbol{Q}(\alpha_k)\} \neq 0$. In [2], Antoulas *et al.* pointed out that this problem can be solved by constructing the parameters $\boldsymbol{P}_R(z)$ and $\boldsymbol{Q}_R(z)$ in polynomial echelon form, provided that $\boldsymbol{\Theta}(z)$ is a *column-reduced* polynomial matrix. In what follows, we show how to modify Algorithm 6.1 so as to obtain the generating system in column-reduced form.

Suppose the transfer matrix $\overline{\boldsymbol{\Theta}}_i(z) = \boldsymbol{\Theta}_0(z)\ \boldsymbol{\Theta}_1(z) \cdots \boldsymbol{\Theta}_{i-1}(z)$ of the first i sections is column-reduced (i.e., the degree of the determinant is equal to the sum of the column degrees). The question is how to construct the next section $\boldsymbol{\Theta}_i(z)$ so that the multiplication

$$\overline{\boldsymbol{\Theta}}_{i+1}(z) = \overline{\boldsymbol{\Theta}}_i(z)\boldsymbol{\Theta}_i(z) \tag{57}$$

preserves column-reducedness. By setting $a_i = 0$, the transfer matrix $\boldsymbol{\Theta}_i(z)$ can be expressed as

$$\boldsymbol{\Theta}_i(z) = \boldsymbol{\Theta}_i \begin{bmatrix}\boldsymbol{I}_{j-1} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & z-f_i & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{r-j}\end{bmatrix}.$$

Multiplication by the diagonal factor obviously preserves column-reducedness (it increases both the degree of the determinant and the sum of the column degrees by one). Therefore, we need to concentrate only on the constant factor $\boldsymbol{\Theta}_i$. Multiplication by $\boldsymbol{\Theta}_i$ does not change the degree of $\det\overline{\boldsymbol{\Theta}}_i(z)$. Thus, the objective is to design $\boldsymbol{\Theta}_i$ in such a way that the multiplication in (57) leaves the column degrees of $\overline{\boldsymbol{\Theta}}_i(z)$ unaltered. This can be done in the following manner.

Lemma 9.1 (Column-Reduced Transfer Matrix): Let $\overline{\boldsymbol{\Theta}}_i(z)$ be a column-reduced polynomial matrix with column degrees $\kappa_1, \kappa_2, \cdots, \kappa_r$, and let $\boldsymbol{g}_{i,0} = [g_{i,0}^1 \quad g_{i,0}^2 \quad \cdots \quad g_{i,0}^r]$. Choose an index j so that

$$\kappa_j = \min_{\ell}\{\kappa_\ell\colon\ g_{i,0}^\ell \neq 0\}.$$

In other words, let κ_j be the smallest among the column degrees that correspond to a nonzero entry in $\boldsymbol{g}_{i,0}$. Determine $\boldsymbol{\Theta}_i$ so that

$$\boldsymbol{g}_{i,0}\boldsymbol{\Theta}_i = [g_{i,0}^1 \quad \cdots \quad g_{i,0}^{j-1} \quad g_{i,0}^j \quad g_{i,0}^{j+1} \quad \cdots \quad g_{i,0}^r]$$

$$\cdot\begin{bmatrix}1 &&&&&&\\ &\ddots&&&&&\\ &&1&&&&\\ \vartheta_1 & \cdots & \vartheta_{j-1} & 1 & \vartheta_{j+1} & \cdots & \vartheta_r\\ &&&&1&&\\ &&&&&\ddots&\\ &&&&&&1\end{bmatrix}$$

$$= [0 \quad \cdots \quad 0 \quad g_{i,0}^j \quad 0 \quad \cdots \quad 0] \tag{58}$$

with $\vartheta_\ell = -g_{i,0}^\ell/g_{i,0}^j$. The polynomial matrix $\overline{\boldsymbol{\Theta}}_i(z)\boldsymbol{\Theta}_i$ then remains column-reduced. ■

Proof: Multiplication by $\boldsymbol{\Theta}_i$ means adding the scaled version of the jth column of $\overline{\boldsymbol{\Theta}}_i(z)$ to the other columns. Since the degree of the jth column is not larger than the degree of any other column in the matrix, this operation does not change the column degrees of $\overline{\boldsymbol{\Theta}}_i(z)$. ■

The following algorithm shows how to keep track of the column degrees of the generating system $\boldsymbol{\Theta}(z)$.

Algorithm 9.2 (Minimal Interpolation):

Stage I: Construct $\boldsymbol{F}, \boldsymbol{G}$, and $\boldsymbol{J}$ as shown in (10). Set $\boldsymbol{G}_0 = \boldsymbol{G}, \boldsymbol{F}_0 = \boldsymbol{F}$, and $\kappa_1 = \kappa_2 = \cdots = \kappa_r = 0$. Repeat the following steps for $i = 0, 1, \cdots, n-1$.

Step 1: Same as Step 1) in Algorithm 6.1.

Step 2: Construct $\boldsymbol{\Theta}_i$ as shown in Lemma 9.1. Let j denote the position of the pivot element.

Step 3: Choose $a_i = 0$.

Step 4: Obtain $\boldsymbol{\Theta}_i(z)$ as

$$\boldsymbol{\Theta}_i(z) = \boldsymbol{\Theta}_i \begin{bmatrix}\boldsymbol{I}_{j-1} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & z-f_i & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{r-j}\end{bmatrix}.$$

Step 5: Update the generator $\boldsymbol{G}_i$ as shown in Step 5 of Algorithm 6.1, and set $\kappa_j := \kappa_j + 1$.

Stage II: Now $\boldsymbol{\Theta}(z) = \boldsymbol{\Theta}_0(z)\boldsymbol{\Theta}_1(z)\cdots\boldsymbol{\Theta}_{n-1}(z)$ is column-reduced. with column degrees $\kappa_1, \cdots, \kappa_r$. The family of minimal interpolants can be parameterized by choosing $\boldsymbol{P}(z)$ and $\boldsymbol{Q}(z)$ in a special polynomial echelon form as shown in [2]. ■

In the generic case, the pivot elements can be chosen cyclically from the 1st, 2nd, $\cdots, r$th positions of $\boldsymbol{g}_i$. This leads to a characteristic feedforward cascade where the delay elements lie successively in the first, second, $\cdots, r$th line (see Fig. 6 in the Appendix).

Furthermore, the column degrees $\kappa_1, \kappa_2, \cdots, \kappa_r$ are identical to the controllability indices of the pair $\{\boldsymbol{F}, \boldsymbol{G}\}$. In this manner Algorithm 9.2 can be used to compute the controllability indices of any pair $\{\boldsymbol{F}, \boldsymbol{G}\}$ where $\boldsymbol{F}$ is in Jordan canonical form.

X. Concluding Remarks

We have developed a fast recursive algorithm for solving the left-sided tangential interpolation problem. The basis of the presented method is a generalized Schur-type algorithm originally developed in the context of recursive factorization of non-Hermitian matrices possessing displacement structure. The advantage of this approach is multifold. First, the recursive

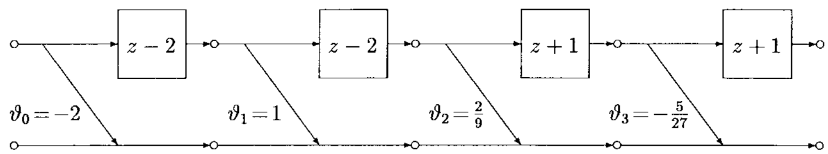


Fig. 4. The Hermite lattice in the scalar case.

algorithm allows us to update the solution whenever a new interpolation point is added to the input data set. Second, the matrix $\boldsymbol{R}$ whose inverse appears in earlier solutions (see Section I-B) does not have to be invertible or strongly regular. There are no break-downs and no singular minors in the algorithm. The generating system is constructed in cascade form by using only first-order sections. Third, the inherent freedom in $\boldsymbol{\Theta}_i$ combined with certain characteristic pivoting schemes can be used to improve numerical accuracy, and impose additional constraints on the interpolants. In particular, we have obtained a recursive solution for the minimal tangential interpolation problem.

APPENDIX

Example 1: Scalar Hermite Interpolation

Let $\alpha_0 = 2$ and $\alpha_1 = -1$. Find a scalar polynomial $h(z) \in \mathbb{C}(z)$ with degree $\deg h(z) \leq 3$ such that

$$h(\alpha_0) = 2, \quad h^{(1)}(\alpha_0) = -1, \quad h(\alpha_1) = 3, \quad h^{(1)}(\alpha_1) = 2.$$

The number of nodes is $m = 2$. The associated multiplicities are $r_0 = 2, r_1 = 2$. The input arrays of the generalized Schur algorithm are

$$\boldsymbol{F} = \left[\begin{array}{cc|cc} 2 & & & \\ 1 & 2 & & \\ \hline & & -1 & \\ & & 1 & -1 \end{array}\right], \qquad \boldsymbol{G} = \left[\begin{array}{c|c} 1 & 2 \\ 0 & -1 \\ \hline 1 & 3 \\ 0 & 2 \end{array}\right]$$

$$\boldsymbol{J} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

i.e., $\boldsymbol{F}$ contains two Jordan blocks of size 2. Four consecutive steps of Algorithm 3.1 yields the generating system

$$\begin{aligned} \boldsymbol{\Theta}(z) &= \boldsymbol{\Theta}_0(z)\boldsymbol{\Theta}_1(z)\boldsymbol{\Theta}_2(z)\boldsymbol{\Theta}_3(z) \\ &= \begin{bmatrix} (z-2)^2(z+1)^2 & -\frac{5}{27}z^3 + \frac{7}{9}z^2 + \frac{1}{9}z - \frac{104}{27} \\ 0 & 1 \end{bmatrix}. \end{aligned}$$

The Hermite interpolating polynomial $h(z) = \frac{5}{27}z^3 - \frac{7}{9}z^2 - \frac{1}{9}z + \frac{104}{27}$ can be physically implemented as the negative transfer function of the feedforward lattice from the top left input to the bottom right output in Fig. 4. Note that the Hermite polynomial can be represented in the Newton basis $\{1, (z-2), (z-2)^2, (z-2)^2(z+1)\}$ as $h(z) = \vartheta_0 + \vartheta_1(z-2) + \vartheta_2(z-2)^2 + \vartheta_3(z-2)^2(z+1)$. The set of all interpolating polynomials can be parameterized as $y_{\mathrm{pol}}(z) = (z-2)^2(z+1)^2P(z) + h(z)$, where $P(z)$ is an arbitrary polynomial. Note that $P(z) \neq 0$ implies $\deg y(z) \geq 4$. Thus the Hermite polynomial has minimal degree in the set of polynomial interpolants. The set of all rational solutions can be parameterized as

$$y(z) = (z-2)^2(z+1)^2\frac{P(z)}{Q(z)} + h(z)$$

where $P(z)$ and $Q(z)$ are polynomials such that $Q(\alpha_k) \neq 0$ for $k = 0, 1$.

Example 2: Matrix Taylor Interpolation:

Let $\alpha_0 = 3$ be a complex point. Find a polynomial matrix $\boldsymbol{T}(z) \in \mathbb{C}_1^{2\times 3}[z]$ such that

$$\boldsymbol{T}(\alpha_0) = \begin{bmatrix} 1 & -2 & 1 \\ -1 & -1 & 2 \end{bmatrix}, \qquad \boldsymbol{T}^{(1)}(\alpha_0) = \begin{bmatrix} 2 & -1 & 3 \\ -2 & 1 & 1 \end{bmatrix}.$$

In the matrix case each point α_k gives rise to two tangential constraints. Therefore, $m = 2, n = 4$, and

$$\boldsymbol{F} = \left[\begin{array}{cc|cc} 3 & & & \\ 1 & 3 & & \\ \hline & & 3 & \\ & & 1 & 3 \end{array}\right], \qquad \boldsymbol{G} = \left[\begin{array}{ccc|cc} 1 & 0 & 1 & -2 & 1 \\ 0 & 0 & 2 & -1 & 3 \\ \hline 0 & 1 & -1 & -1 & 2 \\ 0 & 0 & -2 & 1 & 1 \end{array}\right]$$

$$\boldsymbol{J} = \boldsymbol{I}_3 \oplus -\boldsymbol{I}_2.$$

Four iterations of Algorithm 3.1 yields the generating system shown at the bottom of the page.

Thus, the Taylor matrix polynomial is given by

$$\boldsymbol{T}(z) = \begin{bmatrix} 2z-5 & -z+1 & 3z-8 \\ -2z+5 & +z-4 & z-1 \end{bmatrix}.$$

Fig. 5 depicts a physical implementation of $\boldsymbol{T}(z)$.

$$\boldsymbol{\Theta}(z) = \begin{bmatrix} (z-3)^2 & 0 & -2z+5 & z-1 & -3z+8 \\ 0 & (z-3)^2 & 2z-5 & -z+4 & -z+1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

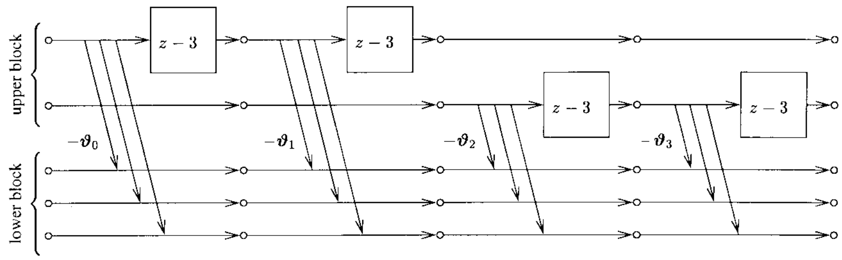


Fig. 5. The Taylor lattice in the matrix case.

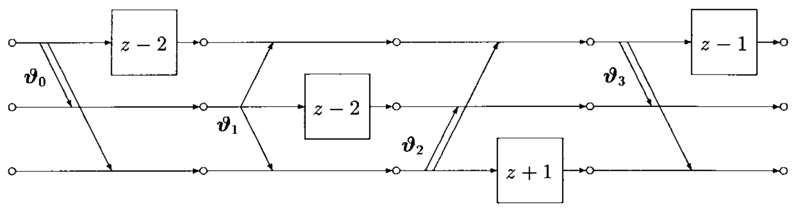


Fig. 6. Feedforward lattice with a column-reduced transfer matrix.

Example 3: Minimal McMillan Degree Interpolation:

Let $\alpha_0 = 2, \alpha_1 = -1$, and $\alpha_2 = 1$ be three points. Find a rational matrix $\boldsymbol{Y}(z) \in \mathbb{C}^{2\times 1}$ with *minimal* McMillan degree such that

$$\begin{aligned}[1 \quad -1 \quad 2 \quad -1]\begin{bmatrix}\boldsymbol{Y}(\alpha_0) & \boldsymbol{Y}^{(1)}(\alpha_0)\\ \mathbf{0} & \boldsymbol{Y}(\alpha_0)\end{bmatrix} &= [0 \quad 1]\\ [3 \quad 1]\boldsymbol{Y}(\alpha_1) &= 3\\ [0 \quad -1]\boldsymbol{Y}(\alpha_2) &= -1.\end{aligned}$$

Now $n = 3, r_0 = 2, r_1 = 1, r_2 = 1$, and the initial arrays are

$$\boldsymbol{F} = \left[\begin{array}{cc|c|c} 2 & & & \\ 1 & 2 & & \\ \hline & & -1 & \\ \hline & & & \end{array}\right], \qquad \boldsymbol{G} = \left[\begin{array}{cc|c} 1 & -1 & 0\\ 2 & -1 & 1\\ \hline 3 & 1 & 3\\ \hline 0 & -1 & -1\end{array}\right]$$

$$\boldsymbol{J} = \boldsymbol{I}_2 \oplus -1.$$

Four iterations of Algorithm 9.2 yield the transfer matrix

$$\begin{aligned}\boldsymbol{\Theta}(z) &= \boldsymbol{\Theta}_0(z)\boldsymbol{\Theta}_1(z)\boldsymbol{\Theta}_2(z)\boldsymbol{\Theta}_3(z)\\ &= \begin{bmatrix}(z-1)(z+10) & 0 & -(z+1)\\ 12(z-1) & z-2 & -(z+1)\\ -13(z-1) & 1 & z+1\end{bmatrix}.\end{aligned}$$

Fig. 6 shows that the delay elements of the lattice sections are located repeatedly in the zeroth, first, and second scalar channel. This characteristic configuration yields a column-reduced transfer matrix. The column degrees of $\boldsymbol{\Theta}(z)$ are $\kappa_0 = 2, \kappa_1 = 1, \kappa_2 = 1$. The minimal interpolants have McMillan degree $\delta\{\boldsymbol{Y}_{\min}(z)\} = 1$, and all of them can be parameterized as

$$\boldsymbol{Y}_{\min}(z) = -\frac{\begin{bmatrix}0\\ z-2\end{bmatrix}P - \begin{bmatrix}z+1\\ z+1\end{bmatrix}Q}{P+(z+1)Q}$$

where $P, Q \in \mathbb{C}$ are scalar parameters such that $P + (\alpha_k + 1)Q \neq 0$ for $k = 0, 1, 2$.

ACKNOWLEDGMENT

The authors would like to thank V. Olshevsky, J. Kamali, and the reviewers of the manuscript for helpful remarks and references.

REFERENCES

[1] A. C. Antoulas and J. C. Willems, “Minimal rational interpolation and Prony’s method,” in *Analysis and Optimization of Systems*, A. Bensoussan and J. L. Lions, Eds., vol. 144, *Lecture Notes in Control and Information Sciences*. New York: Springer-Verlag, 1990, pp. 297–306.

[2] A. C. Antoulas, J. A. Ball, J. Kang, and J. C. Willems, “On the solution of the minimal rational interpolation problem,” *Linear Algebra Its Appl.*, vol. 137/138, pp. 511–573, 1990.

[3] A. C. Antoulas and J. C. Willems, “A behavioral approach to linear exact modeling,” *IEEE Trans. Automat. Contr.*, vol. 38, pp. 1776–1802, Dec. 1993.

[4] J. A. Ball, I. Gohberg, and L. Rodman, “Two-sided Lagrange-Sylvester interpolation problems for rational matrix functions,” in *Proc. Symp. Pure Mathematics*, American Mathematical Soc., 1990, vol. 51, pp. 17–83.

[5] ——, *Interpolation of Rational Matrix Functions*. Basel: Birkhäuser, 1990.

[6] A. H. Sayed, T. Kailath, H. Lev-Ari, and T. Constantinescu, “Recursive solutions of rational interpolation problems via fast matrix factorization,” *Integral Equations and Operator Theory*, vol. 20, pp. 84–118, Sept. 1994.

[7] J. A. Ball, M. A. Kaashoek, G. Groenewald, and J. Kim, “Column reduced rational matrix functions with given null-pole data in the complex plane,” *Linear Algebra Its Appl.*, vol. 203/204, pp. 111–138, 1994.

[8] J. A. Ball and J. Kang, “Matrix polynomial solutions of tangential Lagrange-Sylvester interpolation conditions of low McMillan degree,” *Linear Algebra Its Appl.*, vol. 137/138, pp. 699–746, 1990.

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

[10] S. P. Boyd and C. H. Barratt, *Linear Controller Design*. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[11] M. Vidyasagar, *Control System Synthesis*. Cambridge, MA: MIT, 1985.

[12] J. Kamali, T. Boros, T. Kailath, and G. Franklin, “Q-parametrization for unstable plants: A displacement structure approach,” in *Proc. American Contr, Conf.*, vol. 6, pp. 4401–4402, 1995.

[13] T. Kailath and A. H. Sayed, “Displacement structure: Theory and applications,” *SIAM Rev.*, vol. 37, no. 3, pp. 297–386, 1995.

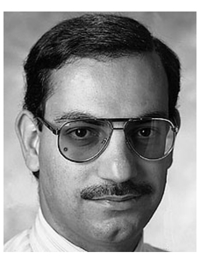
[14] A. H. Sayed and T. Kailath, “Fast algorithms for generalized displacement structures and lossless systems,” *Linear Algebra Its Appl.*, vol. 219, pp. 49–78, Apr. 1995.

[15] T. Boros, A. H. Sayed, and T. Kailath, "Structured matrices and unconstrained rational interpolation problems," *Linear Algebra Its Appl.*, vol. 203/204, pp. 155–188, 1994.
[16] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1980.
[17] M. Green and D. Limebeer, *Linear Robust Control*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
[18] T. Boros, J. Kamali, and T. Kailath, "Analytic interpolation problems and lattice filter design," in *Proc. 28th Asilomar Conf.*, 1994, vol. 1, pp. 368–371.
[19] J. A. Ball, I. Gohberg, and L. Rodman, "Simultaneous residue interpolation for rational matrix functions," *Integral Equations and Operator Theory*, vol. 13, pp. 611–637, 1990.
[20] A. M. Bruckstein, B. C. Levy, and T. Kailath, "Differential methods in inverse scattering," *SIAM J. Appl. Math.*, vol. 45, pp. 312–335, 1985.
[21] T. Boros, "Studies in displacement structure theory," Ph.D. dissertation, Stanford Univ., Stanford, CA, June 1996.
[22] A. M. Bruckstein and T. Kailath, "Inverse scattering for discrete transmission-line models," *SIAM Rev.*, vol. 29, pp. 359–388, 1987.
[23] ——, "An inverse scattering framework for several problems in signal processing," *IEEE ASSP Mag.*, pp. 6–20, Jan. 1987.
[24] P. J. Davis, *Interpolation and Approximation*. New York: Dover, 1975.

Tibor Boros received the Diploma degree with Honors in electrical engineering from the Technical University of Budapest, Hungary, in 1991 and the M.S.E.E and Ph.D degrees in electrical engineering from Stanford University, Stanford, CA, in 1993 and 1996, respectively.

From 1994 to 1995, he was affiliated with the RICOH California Research Center, where he developed image processing algorithms. From 1995 to 1996, he was a Member of the Technical Team at the Wireless Research Group of Watkins-Johnson Company, where he designed smart antenna arrays for mobile communication systems. Currently he is working as a research engineer at ArrayComm, Inc. where he participates in the design and development of wireless local loop systems. His research interests include interpolation theory, control theory, signal processing, and numerical linear algebra.

Dr. Boros received the Hungarian Foundation for Progress in Technology prize in 1989, 1990, and 1991, the Scholarship of the Hungarian Republic in 1990, and the Pro Scientia Medal of the Academic Student Circle in 1991.



Ali H. Sayed (S'90–M'92) was born in Sao Paulo, Brazil. In 1981 he graduated in first place in the National Lebanese Baccalaureat, with the highest score in the history of the examination. In 1987, he received the degree of engineer in electrical engineering from the University of Sao Paulo, Brazil, and was the first place graduate of the School of Engineering. In 1989, he received, with distinction, the M.S. degree in electrical engineering from the University of Sao Paulo, Brazil. In 1992, he received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA.

From September 1992 to August 1993, he was a Research Associate with the Information Systems Laboratory at Stanford University, after which he joined, as an Assistant Professor, the Department of Electrical and Computer Engineering at the University of California, Santa Barbara. Since July 1996, he has been an Associate Professor in the Department of Electrical Engineering at UCLA. His research interests include the areas of adaptive systems, system theory, interplays between signal processing and control methodologies, interpolation theory, and structured computations in systems and mathematics. He has over 90 journal and conference publications.

Dr. Sayed is a member of SIAM and ILAS. He was awarded the Institute of Engineering Prize and the Conde Armando A. Penteado Prize, both in 1987 in Brazil. He is a recipient of the 1996 IEEE Donald G. Fink Prize.

Thomas Kailath (S'57–M'62–F'70–LF'72) was educated in Poona, India, and received the S.M. and Sc.D. degrees from the Massachusetts Institute of Technology in 1959 and 1961, respectively.

From October 1961 to December 1962, he worked at the Jet Propulsion Laboratories, Pasadena, CA, where he also taught part-time at the California Institute of Technology. He then came to Stanford, where he served as Director of the Information Systems Laboratory from 1971 through 1980, as Associate Chairman from 1981 to 1987, and currently holds the Hitachi America Professorship in Engineering. He has also held short term appointments at several institutions around the world. He has worked in a number of areas including information theory, communications, computation, control, signal processing, VLSI design, statistics, linear algebra and operator theory; his recent interests include applications of signal processing, computation and control to problems in semiconductor manufacturing. He is the author of *Linear Systems* (Englewood Cliffs, NJ: Prentice-Hall, 1980) and *Lectures on Wiener and Kalman Filtering* (New York: Springer-Verlag, 1981). He has held Guggenheim and Churchill fellowships, among others.

Dr. Kailath has received awards from the IEEE Information Theory Society, the American Control Council, and the IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING. He served as President of the IEEE Information Theory Society in 1975 and has been awarded honorary doctorates by Linköping University, Sweden, and by Strathclyde University, Scotland. He received the 1995 IEEE Education Medal and the 1996 IEEE Donald G. Fink Prize Award. He is a Fellow of the Institute of Mathematical Statistics, and a member of the National Academy of Engineering and the American Academy of Arts and Sciences.