On Low-Complexity Approximation of Matrices

Alle-Jan van der Veen and Patrick Dewilde
Delft University of Technology
Department of Electrical Engineering
Mekelweg 4
2628 CD Delft, The Netherlands
email: allejan@dutentb.et.tudelft.nl; Tel.: (+31 15) 781442; Fax: (+31 15) 623271

The operation 'multiplication of a vector by a matrix' can be represented by a computational scheme (or model) that acts on the entries of the vector sequentially. The number of intermediate quantities ('states') that are needed in the computations is a measure of the complexity of the model. If that complexity is low, then not only multiplication, but also other operations such as inversion, can be carried out efficiently using the model rather than the original matrix. In the introductory sections, we describe an algorithm to derive a computational model of minimal complexity that gives an exact representation of an arbitrary upper triangular matrix. The main result of the paper is an algorithm for computing an approximating matrix with a model of (much) lower complexity than the original — as low as possible for a given tolerance on the approximation error. As measure for the tolerance we will use a strong norm which we will call the *Hankel norm.* It is a generalization of the Hankel norm which is used in the classical model approximation theory for complex analytical functions.

Running title: Low-complexity approximation of matrices

This research was supported in part by the commission of the EC under the ESPRIT BRA program 6632 (NANA2).
Linear Algebra and its Applications, 205-206:1145-1202, July 1994.

1. INTRODUCTION

1.1. Computational linear algebra and time-varying modeling

In the intersection of linear algebra and system theory is the field of *computational linear algebra*. Its purpose is to find efficient algorithms for linear algebra problems (matrix multiplication, inversion, approximation). A useful model for matrix computations is provided by dynamical system theory. Such a model is often quite natural: in any algorithm which computes a matrix multiplication or inversion, the global operation is decomposed into a sequence of local operations that each act on a limited number of matrix entries (ultimately two), assisted by intermediate quantities that connect the local operations. These quantities can be called the states of the algorithm, and translate to the state of the dynamical system that is the computational model of the matrix operation. Although many matrix operations can be captured this way by some linear dynamical system, our interest is in matrices that possess some kind of structure which allows for efficient ("fast") algorithms: algorithms that exploit this structure. Structure in a matrix is inherited from the origin of the linear algebra problem, and is for our purposes typically due to the modeling of some (physical) dynamical system. Many signal processing applications, inverse scattering problems and least squares estimation problems give rise to structured matrices that can indeed be modeled by a low complexity computational system.

Besides sparse matrices (many zero entries), traditional structured matrices are Toeplitz and Hankel matrices (constant along diagonals or anti-diagonals), which translate to linear time-invariant (LTI) systems. Associated computational algorithms are well-known, *e.g.,* for Toeplitz systems we have Schur recursions for LU- and Cholesky factorization [1], Levinson recursions for factorization of the inverse [2], Gohberg/Semencul recursions for computing the inverse [3], and Schur-based recursions for QR factorization [4]. The resulting algorithms have computing complexity of order $\mathcal{O}(n^2)$ for matrices of size $(n \times n)$, as compared to $\mathcal{O}(n^3)$ for algorithms that do not take the Toeplitz structure into account. Generalizations of the Toeplitz structure are obtained by considering matrices which have a so-called displacement structure [5, 6]: matrices G for which there are (simple) matrices F_1, F_2 such that $G - F_1^* G F_2$ is of low rank. Overviews of inversion and factorization algorithms for such matrices can be found in [7, 8].

The Toeplitz, Hankel and displacement structures give rise to computational models with a low number of inputs and outputs. In this paper, we pursue a complementary notion of structure which we will call the state structure. The state structure applies to upper triangular matrices and is seemingly unrelated to the Toeplitz or displacement structure mentioned above. A first purpose of the computational schemes considered in this paper is to perform a desired linear transformation T on some vector ('input sequence') u,

$$u = [u_1 \quad u_2 \quad \cdots \quad u_n],$$

with an output vector or sequence $y = uT$ as the result. The key idea is that we can associate with this matrix-vector multiplication a computational network that takes u and computes y, and that matrices with a 'small' state structure have a computational network of low complexity so that using the network to compute y is more efficient than computing uT directly. To introduce this notion, consider an upper triangular matrix T along with its inverse,

$$T = \begin{bmatrix} 1 & 1/2 & 1/6 & 1/24 \\ & 1 & 1/3 & 1/12 \\ & & 1 & 1/4 \\ & & & 1 \end{bmatrix} \qquad T^{-1} = \begin{bmatrix} 1 & -1/2 & & \\ & 1 & -1/3 & \\ & & 1 & -1/4 \\ & & & 1 \end{bmatrix}.$$

The inverse of T is sparse, which is an indication of a 'small' state structure. Computational networks for the computation $y = uT$ are depicted in figure 1. The computations in the network are split into sections, which we will call *stages*, where the k-th stage consumes u_k and produces y_k. The dependence of y_k on

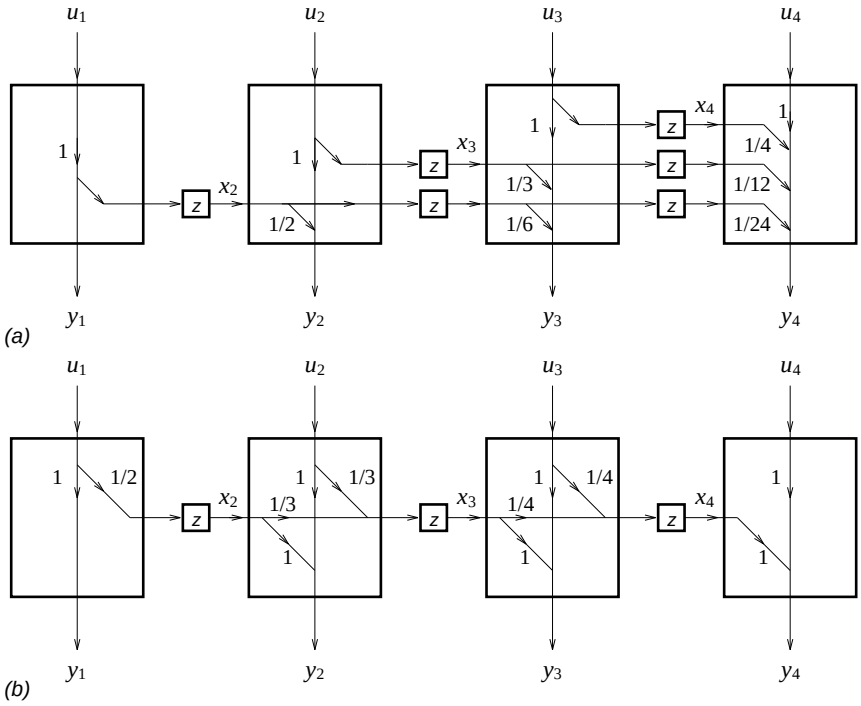

Figure 1. Computational networks corresponding to T. (*a*) Direct (trivial) realization, (*b*) minimal realization.

u_i, ($i < k$) introduces intermediate quantities x_k called *states*. At each point k the processor in the stage at that point takes its input data u_k from the input sequence u and computes a new output data y_k which is part of the output sequence y generated by the system. To execute the computation, the processor will use some remainder of its past history, *i.e.*, the state x_k, which has been computed by the previous stages and which was temporarily stored in registers indicated by the symbol z. The complexity of the computational network is equal to the number of states at each point. The total number of multiplications required in the minimal realization (figure 1(*b*)) that are different from 1 is 5, as compared to 6 in a direct computation using T (figure 1(*a*)). Although we have gained only one multiplication here, for a less moderate example, say an $(n \times n)$ upper triangular matrix with $n = 10000$ and $d \ll n$ states at each point, the number of multiplications in the network is in the order of $\mathcal{O}(d^2 n)$ and can even be further reduced to $\mathcal{O}(4dn)$, instead of $\mathcal{O}(1/2\, n^2)$ for a direct computation using T. Note however that the number of states can vary from one point to the other, depending on the nature of T. In the example above, the number of states entering the network at point 1 is zero, and the number of states leaving the network at point 4 is also zero. If we would change the value of one of the entries of the 2×2 submatrix in the upper-right corner of T to a different value, then, in the minimal network, two states would have been required to connect stage 2 to stage 3.

The computations in the network can be summarized by the following recursion, for $k = 1$ to n:

$$y = uT \qquad \Leftrightarrow \qquad \begin{aligned} x_{k+1} &= x_k A_k + u_k B_k \\ y_k &= x_k C_k + u_k D_k \end{aligned} \tag{1.1}$$

or

$$\begin{bmatrix} x_{k+1} & y_k \end{bmatrix} = \begin{bmatrix} x_k & u_k \end{bmatrix} \mathbf{T}_k\,, \qquad \mathbf{T}_k = \begin{bmatrix} A_k & C_k \\ B_k & D_k \end{bmatrix}$$

in which x_k is the state vector at time k (taken to have d_k entries) A_k is a $d_k \times d_{k+1}$ (possibly non-square) matrix, B_k is a $1 \times d_{k+1}$ vector, C_k is a $d_k \times 1$ vector, and D_k is a scalar. More general computational

networks will have the number of inputs and outputs at each stage to be different from one, and possibly also varying from stage to stage. In the example (figure 1(*b*)), we have as sequence of realization matrices

$$\mathbf{T}_1 = \begin{bmatrix} \cdot & \cdot \\ 1/2 & 1 \end{bmatrix} \qquad \mathbf{T}_2 = \begin{bmatrix} 1/3 & 1 \\ 1/3 & 1 \end{bmatrix} \qquad \mathbf{T}_3 = \begin{bmatrix} 1/4 & 1 \\ 1/4 & 1 \end{bmatrix} \qquad \mathbf{T}_4 = \begin{bmatrix} \cdot & 1 \\ \cdot & 1 \end{bmatrix}$$

where the '$\cdot$' indicates entries that actually have dimension 0 because the corresponding states do not exist. The recursion in equation (1.1) shows that it is a recursion for increasing values of k: the order of computations in the network is strictly from left to right, and we cannot compute y_k unless we know x_k, *i.e.*, unless we have processed $u_1, \cdots, u_{k-1}$. On the other hand, y_k does not depend on $u_{k+1}, \cdots, u_n$. This is a direct consequence of the fact that T has been chosen upper triangular, so that such an ordering of computations is indeed possible.

A link with system theory is obtained when T is regarded as the transfer matrix of a *non-stationary* causal linear system with input u and output $y = uT$. The k-th row of T then corresponds to the impulse response of the system when excited by an impulse at time instant i, that is, the output y due to an input vector u with entries $u_i = \delta_k^i$, where δ_k^i is the Kronecker delta. The case where T has a Toeplitz structure then corresponds with a time-invariant system for which the impulse response due to an impulse at time $i+1$ is just the same as the response due to an impulse at time i, shifted over one position. The computational network is called a state space realization of T, and the number of states at each point of the computational network is called the system order of the realization at that point in time. For time-invariant systems, the state realization can be chosen constant in time. Since for time-varying systems the number of state variables need not be constant in time, but can increase or shrink, it is seen that in this respect the time-varying realization theory is much richer, and that the accuracy of an approximating computational network of T can be varied in time at will.

If the number of state variables is relatively small, then the computation of the output sequence is efficient in comparison with a straight computation of $y = uT$. One example of a matrix with a small state space is the case where T is an upper triangular band-matrix: $T_{ij} = 0$ for $j - i > p$. In this case, the state dimension is equal to or smaller than p. However, the state space model can be much more general, *e.g.*, if a banded upper matrix has an inverse, then this inverse is known to have a sparse state space (of the same complexity) too, as we had in the example above. Moreover, this inversion can be easily carried out by local computations on the realization of T (we assume D_k square; for the general case, see [9]): let $y = uT \Leftrightarrow u = yT^{-1} =: yS$, then

$$\begin{cases} x_{k+1} = x_k A_k + u_k B_k \\ y_k = x_k C_k + u_k D_k \end{cases} \quad \Leftrightarrow \quad \begin{cases} x_{k+1} = x_k (A_k - C_k D_k^{-1} B_k) + y_k D_k^{-1} B_k \\ u_k = -x_k C_k D_k^{-1} + y_k D_k^{-1} \end{cases}$$

so that a model of S is given by

$$\mathbf{S}_k = \begin{bmatrix} A_k - C_k D_k^{-1} B_k & -C_k D_k^{-1} \\ D_k^{-1} B_k & D_k^{-1} \end{bmatrix} \tag{1.2}$$

Observe that the model for $S = T^{-1}$ is obtained in a *local* way from the model of T: $\mathbf{S}_k$ depends only on $\mathbf{T}_k$. The sum and product of matrices with sparse state structure have again a sparse state structure with number of states at each point not larger than the sum of the number of states of its component systems, and computational networks of these compositions (but not necessarily minimal ones) can be easily derived from those of its components. Finally, we mention that a matrix T_1 that is not upper triangular can be split into an upper triangular and a lower triangular part, each of which can be separately modeled by a computational network. The computational model of the lower triangular part has a recursion which runs backwards:

$$\begin{aligned} x'_k &= x'_{k+1} A'_k + u_k B'_k \\ y_k &= x'_{k+1} C'_k + u_k D'_k \end{aligned} \quad .$$

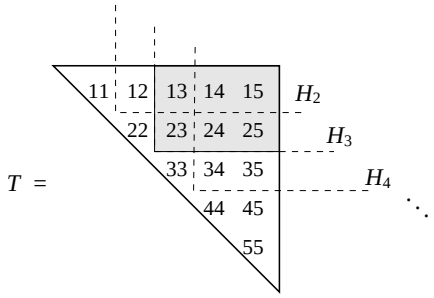

Figure 2. Hankel matrices are (mirrored) submatrices of T.

The model of the lower triangular part can be used to determine a model of a unitary upper matrix U which is such that U^*T is upper and has a sparse state structure. In this way, results derived for upper matrices, such as the above inversion formula, can be generalized to matrices of mixed type [9].

1.2. Realization algorithm

One might wonder for which class of matrices T there exists a sparse computational network (or state space realization) that realizes the same multiplication operator. For an upper triangular $(n \times n)$ matrix T, let the matrices H_i $(1 \le i \le n)$, which are submatrices of T, be

$$H_i = \begin{bmatrix} T_{i-1,i} & T_{i-1,i+1} & \cdots & T_{i-1,n} \\ T_{i-2,i} & T_{i-2,i+1} & & \vdots \\ \vdots & & \ddots & T_{2,n} \\ T_{1,i} & \cdots & T_{1,n-1} & T_{1,n} \end{bmatrix}$$

(see figure 2). We call the H_i (time-varying) Hankel matrices, as they will have a Hankel structure (constant along anti-diagonals) if T has a Toeplitz structure.* In terms of the Hankel matrices, the criterion by which matrices with a sparse state structure can be detected is given by the following theorem.

Theorem 1.1. *The number of states that are needed at stage k in a minimal computational network of an upper triangular matrix T is equal to the rank of its k-th Hankel matrix* H_k.

PROOF Suppose that $\{A_k, B_k, C_k, D_k\}$ is a realization for T as in equation (1.1). Then a typical Hankel matrix has the following structure:

$$H_2 = \begin{bmatrix} B_1C_2 & B_1A_2C_3 & B_1A_2A_3C_4 & \cdots \\ B_0A_1C_2 & B_0A_1A_2C_3 & & \\ B_{-1}A_0A_1C_2 & & \ddots & \\ \vdots & & & \end{bmatrix}$$

$$= \begin{bmatrix} B_1 \\ B_0A_1 \\ B_{-1}A_0A_1 \\ \vdots \end{bmatrix} \cdot [C_2 \quad A_2C_3 \quad A_2A_3C_4 \quad \cdots] = \mathcal{C}_2\mathcal{O}_2$$

*Warning: in the current context (arbitrary upper triangular matrices) the H_i do not have a Hankel structure and the predicate 'Hankel matrix' could lead to misinterpretations. Our terminology finds its motivation in system theory, where the H_i are related to an abstract operator H_T which is commonly called the Hankel operator. For time-invariant systems, H_T reduces to an operator with a matrix representation that has indeed a Hankel structure.

From the decomposition $H_k = \mathcal{C}_k\mathcal{O}_k$ it is directly inferred that if A_k is of size $(d_k \times d_{k+1})$, then rank(H_k) is at most equal to d_k. We have to show that there exists a realization $\{A_k, B_k, C_k, D_k\}$ for which $d_k = \text{rank}(H_k)$: if it does, then clearly this must be a minimal realization. To find such a minimal realization, take any minimal factorization $H_k = \mathcal{C}_k\mathcal{O}_k$ into full rank factors $\mathcal{C}_k$ and $\mathcal{O}_k$. We must show that there are matrices $\{A_k, B_k, C_k, D_k\}$ such that

$$\mathcal{C}_k = \begin{bmatrix} B_{k-1} \\ B_{k-2}A_{k-1} \\ \vdots \end{bmatrix} \qquad \mathcal{O}_k = [C_k \quad A_kC_{k+1} \quad A_kA_{k+1}C_{k+2} \quad \cdots] \,. \tag{1.3}$$

To this end, we use the fact that H_k satisfies a shift-invariance property: with $H_2^{\leftarrow}$ denoting H_2 without its first column, we have

$$H_2^{\leftarrow} = \begin{bmatrix} B_1 \\ B_0A_1 \\ B_{-1}A_0A_1 \\ \vdots \end{bmatrix} \cdot A_2 \cdot [C_3 \quad A_3C_4 \quad A_3A_4C_5 \quad \cdots] \,.$$

In general, $H_k^{\leftarrow} = \mathcal{C}_kA_k\mathcal{O}_{k+1}$, and in much the same way, $H_k^{\uparrow} = \mathcal{C}_{k-1}A_{k-1}\mathcal{O}_k$, where $H_k^{\uparrow}$ is H_k without its first row. The shift-invariance properties carry over to $\mathcal{C}_k$ and $\mathcal{O}_k$, *e.g.*, $\mathcal{O}_k^{\leftarrow} = A_k\mathcal{O}_{k+1}$, and we obtain that $A_k = \mathcal{O}_k^{\leftarrow}\mathcal{O}_{k+1}^{*}(\mathcal{O}_{k+1}\mathcal{O}_{k+1}^{*})^{-1}$, where '*' denotes complex conjugate transposition. The inverse exists because $\mathcal{O}_{k+1}$ is of full rank. C_k follows as the first column of the chosen $\mathcal{O}_k$, while B_k is the first row of $\mathcal{C}_{k+1}$. It remains to verify that $\mathcal{C}_k$ and $\mathcal{O}_k$ are indeed generated by this realization. This is straightforward by a recursive use of the shift-invariance properties. ∎

Let's verify theorem 1.1 with the example. The Hankel matrices are

$$H_1 = [\,\cdot\ \cdot\ \cdot\,] \,, \qquad H_2 = [1/2 \ \ 1/6 \ \ 1/24\,] \,,$$

$$H_3 = \begin{bmatrix} 1/3 & 1/12 \\ 1/6 & 1/24 \end{bmatrix} \,, \qquad H_4 = \begin{bmatrix} 1/4 \\ 1/12 \\ 1/24 \end{bmatrix} \,.$$

Since rank$(H_1) = 0$, no states x_1 are needed. One state is needed for x_2 and one for x_4, because rank(H_2) = rank$(H_4) = 1$. Finally, also only one state is needed for x_3, because rank$(H_3) = 1$. In fact, this is (for this example) the only non-trivial rank condition: if one of the entries in H_3 would have been different, then two states would have been needed. In general, rank$(H_i) \le \min(i-1, n-i-1)$, and for a general upper triangular matrix T without state structure, a computational model will indeed require at most $\min(i-1, n-i-1)$ states for x_i.

The construction in the proof of theorem 1.1 leads to a realization algorithm (algorithm 1). In this algorithm, $A(:, 1:p)$ denotes the first p columns of A, and $A(1:p, :)$ the first p rows. The key part of the algorithm is to obtain a basis $\mathcal{O}_k$ for the rowspace of each Hankel matrix H_k of T. The singular value decomposition (SVD) [10] is a robust tool for doing this. It is a decomposition of H_k into factors U_k, Σ_k, V_k, where U_k and V_k are unitary matrices whose columns contain the left and right singular vectors of H_k, and Σ_k is a diagonal matrix with positive entries (the singular values of H_k) on the diagonal. The integer d_k is set equal to the number of nonzero singular values of H_k, and $V_k^*(1:d_k, :)$ contains the corresponding singular vectors. The rows of $V^*(1:d_k, :)$ span the row space of H_k. The rest of the realization algorithm is straightforward in view of the shift-invariance property. Note that, based on the singular values of H_k, a reduced order model can be obtained by taking a smaller basis for $\mathcal{O}_k$, much as in the Principal Component identification method in system theory [11], which is also known as balanced

In: T (an upper triangular $n \times n$ matrix)
Out: $\{\mathbf{T}_k\}_1^n$ (a minimal realization, in output normal form)

$\mathcal{O}_{n+1} = [\,\cdot\,]$
for $k = n, \cdots, 1$
$\quad H_k =: U_k \Sigma_k V_k^*$
$\quad d_k = \mathrm{rank}(\Sigma_k)$
$\quad \mathcal{C}_k = (U_k \Sigma_k)(:, 1{:}d_k)$
$\quad \mathcal{O}_k = V_k^*(1{:}d_k, :)$
$\quad A_k = \mathcal{O}_k [\,0 \quad \mathcal{O}_{k+1}\,]^*$
$\quad C_k = \mathcal{O}_k(:, 1)$
$\quad B_k = \mathcal{C}_{k+1}(1, :)$
$\quad D_k = T_{k,k}$
end

Algorithm 1. Realization algorithm.

model reduction. Although widely used for time-invariant systems, this would result in a "heuristic" model reduction theory, as the modeling error norm is not known. The goal of the present paper is to obtain a precise theory. A final remark is that the above algorithm yields a realization in *output normal form*:

$$A_k A_k^* + C_k C_k^* = I$$

which is a consequence of the fact that an orthonormal basis for the row space of H_k has been used.

1.3. Hankel norm approximation

In the previous section, we have assumed that the given matrix T has indeed a computational model of an order that is low enough to favor the use of a minimal computational network over an ordinary matrix multiplication. However, if the rank of the Hankel matrices of T (*i.e.*, the system order) is not low, then it could make sense to approximate T by a new upper triangular matrix T_a that has a lower complexity, *i.e.*, whose Hankel matrices have low rank. It is of course dependent on the origin of T whether this indeed yields a useful approximation of the underlying (physical) problem that is described by the original matrix. For example, it could happen that the given matrix T is not of low complexity because numerical inaccuracies of the entries of T have increased the rank of the Hankel matrices of T, since the rank of a matrix is a very sensitive (ill-conditioned) parameter. But even if the given matrix T is known to be exact, an approximation by a reduced-order model could be appropriate, for example for design purposes in engineering, to capture the essential behavior of the model. With such a reduced-complexity model, the designer can more easily detect that certain features are not desired and can possibly predict the effects of certain changes in the design; an overly detailed model would rather mask these features.

Because the system order at each point is given by the rank of the Hankel matrix at that point, a possible approximation scheme is to replace each Hankel matrix by one that is of lower rank (this could be done using the SVD). The approximation error could then very well be defined in terms of the individual Hankel matrix approximations as the supremum over the individual approximation errors. The error criterion for which we will obtain a solution is called the Hankel norm. It is defined as the supremum over the operator norm (the spectral norm, or the matrix 2-norm) of each individual Hankel matrix:

$$\| T \|_H = \sup_k \| H_k \| = \sup_k \sup_{\|u\|_2 \le 1} \| u H_k \|_2 \tag{1.4}$$

This is a generalization of the Hankel norm for time-invariant systems. It is a reasonably strong norm: if T is a strictly upper triangular matrix and $||T||_H \leq 1$, then each row and column of T has vector norm smaller than 1. In terms of the Hankel norm, we will prove the following theorem in section 3.

Theorem 1.2. *Let T be a strictly upper triangular matrix and let* $\Gamma = \mathrm{diag}(\gamma_i)$ *be a diagonal Hermitian matrix which parametrizes the acceptable approximation tolerance* ($\gamma_i > 0$)*. Let* H_k *be the Hankel matrix of* $\Gamma^{-1}T$ *at stage k, and suppose that, for each k, none of the singular values of* H_k *are equal to 1. Then there exists a strictly upper triangular matrix* T_a *with system order at stage k at most equal to the number of singular values of* H_k *that are larger than 1, such that*

$$||\Gamma^{-1}(T - T_a)||_H \leq 1 .$$

In fact, there is a collection of such T_a. We will show the theorem by construction and obtain a computational model of a particular T_a as well. Because the Hankel matrices have many entries in common, it is not clear at once that this approximation scheme is feasible: replacing one Hankel matrix by a matrix of lower rank in a certain norm might make it impossible for the next Hankel matrix to be replaced by an optimal approximant (in that norm) such that the part that it has in common with the previous Hankel matrix is approximated by the same matrix. In other words: each individual local optimization might prevent a global optimum. The severity of this dilemma is mitigated by a proper choice of the error criterion: the fact that the above defined Hankel norm uses the operator norm of each Hankel matrix, rather than the stronger Frobenius norm, gives just enough freedom to obtain a nice solution to this dilemma. The solution can even be obtained in a non-iterative form.

Γ can be used to influence the local approximation error. For a uniform approximation, $\Gamma = \gamma I$, and hence $||T - T_a||_H \leq \gamma$: the approximant is γ-close to T in Hankel norm, which implies in particular that the approximation error in each row or column of T is less than γ. If one of the γ_i is made larger than γ, then the error at the i-th row of T can become larger also, which might result in an approximant T_a to take on less states. Hence Γ can be chosen to yield an approximant that is accurate at certain points but less tight at others, and whose complexity is minimal.

Hankel norm approximation theory originates as a special case of the solution to the Schur-Takagi interpolation problem in the context of complex function theory. The solution was formulated by Adamjan, Arov and Krein (AAK) [12], who studied properties of the SVD of infinite Hankel matrices (having a Hankel structure) and associated approximation problems of bounded analytical functions $f(z)$ by rational functions. In linear system theory, it is a well known result of Kronecker that the degree of a rational function is equal to the rank of the Hankel matrix constructed on the coefficients of its Taylor expansion [13]. The main problem with approximating a Hankel matrix using SVD, in the time-invariant context, is to ensure that the approximation has again a Hankel structure. When the function is regarded as the transfer function of a linear time-invariant system this number is the model order. It was remarked in Bultheel-Dewilde [14] and subsequently worked out by a number of authors (Glover [15], Kung-Lin [16], Genin-Kung [17]) that the procedure of AAK could be utilized to solve the problem of optimal model-order reduction of a dynamical time-invariant system, and that, although the Hankel matrix is of infinite size, computations can be made finite if a finite-order state model is already known [14]. It is possible to give a global expression of the approximant, based on a global state space based solution of a related Schur-Takagi interpolation problem; the necessary theory was extensively studied in the book [18]. The computations can also be done in a recursive fashion [19]. State space theory provided a bridge between analytical theory and matrix computations.

In a recent series of papers [20, 21, 22, 23, 24, 25] a theory was developed to derive models for upper triangular matrices as, now time-varying, linear systems. The classical interpolation problems of

Schur or Nevanlinna-Pick can be formulated and solved in a context where diagonals take the place of scalars. A comprehensive treatment can be found in [24], and we will adopt the notation of that paper. A supplementary realization theory of upper operators in a state space context appeared in [25] and provided the tools to solve the generalized Hankel-norm model reduction problem in combination with the interpolation theory. The general solution is published in [26], the present paper is a specialization to finite upper triangular matrices, and contains independent, finite dimensional proofs.

1.4. Numerical example

As an example of the use of theorem 1.2, we consider a matrix T and determine an approximant T_a. Let the matrix to be approximated be

$$T = \left[\begin{array}{cccc|cc} 0 & .800 & .200 & .050 & .013 & .003 \\ 0 & 0 & .600 & .240 & .096 & .038 \\ 0 & 0 & 0 & .500 & .250 & .125 \\ \hline 0 & 0 & 0 & 0 & .400 & .240 \\ 0 & 0 & 0 & 0 & 0 & .300 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

The position of the Hankel matrix H_4 is indicated. Taking $\Gamma = 0.1\,I$, the non-zero singular values of the Hankel operators of $\Gamma^{-1}T$ are

H_1	H_2	H_3	H_4	H_5	H_6
	8.26	6.85	6.31	5.53	4.06
		0.33	0.29	0.23	
			0.01		

Hence T has a state space realization which grows from zero states ($i = 1$) to a maximum of 3 states ($i = 4$), and then shrinks back to 0 states ($i > 6$). The number of Hankel singular values of $\Gamma^{-1}T$ that are larger than one is 1 ($i = 2, \cdots, 6$). At each point in the sequence, this is to correspond to the number of states of the approximant at that point. Using the techniques of this paper, we obtain

$$T_a = \begin{bmatrix} 0 & .790 & .183 & .066 & .030 & .016 \\ 0 & 0 & .594 & .215 & .098 & .052 \\ 0 & 0 & 0 & .499 & .227 & .121 \\ 0 & 0 & 0 & 0 & .402 & .214 \\ 0 & 0 & 0 & 0 & 0 & .287 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

with non-zero Hankel singular values (scaled by Γ)

H_1	H_2	H_3	H_4	H_5	H_6
	8.15	6.71	6.16	5.36	3.82

whose number indeed correspond to the number of Hankel singular values of $\Gamma^{-1}T$ that are larger than 1. Also, the modeling error is

$$T - T_a = \begin{bmatrix} 0 & .010 & .017 & -.016 & -.017 & -.013 \\ 0 & 0 & .006 & .025 & -.002 & -.014 \\ 0 & 0 & 0 & .001 & .023 & .004 \\ 0 & 0 & 0 & 0 & -.002 & .026 \\ 0 & 0 & 0 & 0 & 0 & .013 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

with Hankel norm of $\Gamma^{-1}(T - T_a)$ less than 1:

$$\|\Gamma^{-1}(T - T_a)\|_H = \sup\{0.334,\ 0.328,\ 0.338,\ 0.351,\ 0.347\} = 0.351$$

The realization algorithm (algorithm 1) yields as realization for T

$$\mathbf{T}_1 = \left[\begin{array}{c|c} \cdot & \cdot \\ \hline -.826 & 0 \end{array}\right] \qquad \mathbf{T}_2 = \left[\begin{array}{cc|c} .246 & -.041 & -.968 \\ \hline -.654 & -.00 & 0 \end{array}\right]$$

$$\mathbf{T}_3 = \left[\begin{array}{ccc|c} .397 & -.044 & .000 & -.917 \\ .910 & .140 & .040 & .388 \\ \hline -.573 & .00 & .00 & 0 \end{array}\right] \qquad \mathbf{T}_4 = \left[\begin{array}{cc|c} .487 & .037 & -.873 \\ .853 & -.237 & .465 \\ .189 & .971 & .147 \\ \hline -.466 & .00 & 0 \end{array}\right]$$

$$\mathbf{T}_5 = \left[\begin{array}{cc} -.515 & -.858 \\ .858 & -.515 \\ \hline .300 & 0 \end{array}\right] \qquad \mathbf{T}_6 = \left[\begin{array}{c|c} \cdot & 1 \\ \hline \cdot & 0 \end{array}\right]$$

A realization of the approximant is determined via algorithm 3 in section 3.5 as

$$\mathbf{T}_{a,1} = \left[\begin{array}{c|c} \cdot & \cdot \\ \hline -.993 & 0 \end{array}\right] \qquad \mathbf{T}_{a,2} = \left[\begin{array}{c|c} .293 & -.795 \\ \hline -.946 & 0 \end{array}\right]$$

$$\mathbf{T}_{a,3} = \left[\begin{array}{c|c} .410 & -.629 \\ \hline -.901 & 0 \end{array}\right] \qquad \mathbf{T}_{a,4} = \left[\begin{array}{c|c} .525 & -.554 \\ \hline -.837 & 0 \end{array}\right]$$

$$\mathbf{T}_{a,5} = \left[\begin{array}{c|c} -.651 & -.480 \\ \hline .729 & 0 \end{array}\right] \qquad \mathbf{T}_{a,6} = \left[\begin{array}{c|c} \cdot & .393 \\ \hline \cdot & 0 \end{array}\right]$$

The corresponding computational schemes are depicted in figure 3. It is seen that a small change in T can lead to a significant reduction in the complexity of the computations.

2. NOTATION AND PRELIMINARIES

2.1. Spaces

An essential ingredient of our theory is the concept of non-uniform sequences: vectors whose entries are again vectors in some Euclidean space and which can have different dimensions for each entry. Thus let

$$\mathcal{B} = \mathcal{B}_1 \times \mathcal{B}_2 \times \cdots \times \mathcal{B}_n$$

where $\mathcal{B}_i = \mathbb{C}^{d_i}$, and d_i is the dimension of $\mathcal{B}_i$. Some dimensions might be zero, *e.g.*, $\mathcal{B} = \mathbb{C}^1 \times \varnothing \times \mathbb{C}^2$ is a valid space sequence, and $[0.5, , [2, 1]]$ is an element of $\mathcal{B}$, the 2-norm (vector norm) of which is $(0.25 + 4 + 1)^{1/2}$. A generalized matrix (a block matrix, which we will call a *tableau* to distinguish) is a linear map $\mathcal{M} \to \mathcal{N}$, where $\mathcal{M}, \mathcal{N}$ are space sequences as $\mathcal{B}$ above. For example, to $\mathcal{M} = \mathbb{C}^2 \times \varnothing \times \mathbb{C}^1$, $\mathcal{N} = \mathbb{C} \times \mathbb{C} \times \mathbb{C}$ correspond tableaus of the form

$$\begin{array}{c} \\ \mathbb{C}^2 \\ \varnothing \\ \mathbb{C} \end{array} \begin{array}{c} \begin{array}{ccc} \mathbb{C} & \mathbb{C} & \mathbb{C} \end{array} \\ \left[\begin{array}{ccc} \boxed{*} & * & * \\ * & * & * \\ \cdot & \underline{\cdot} & \cdot \\ * & * & \underline{*} \end{array}\right] \end{array}$$

where the $(1, 1)$ entry is identified by a square, the main diagonal is distinguished by an underscore, '$*$' stands for any scalar, and '$\cdot$' stands for an entry with an empty dimension. The above tableau is isomorphic to a 3×3 ordinary matrix. We denote by $\mathcal{X}(\mathcal{M}, \mathcal{N})$ the space of linear maps $\mathcal{M} \to \mathcal{N}$, by $\mathcal{U}(\mathcal{M}, \mathcal{N})$ the space of upper tableaus in $\mathcal{X}(\mathcal{M}, \mathcal{N})$, that is $\mathcal{U} = \{F \in \mathcal{X} : F_{i,j} = 0, i > j\}$, by $\mathcal{L}(\mathcal{M}, \mathcal{N})$

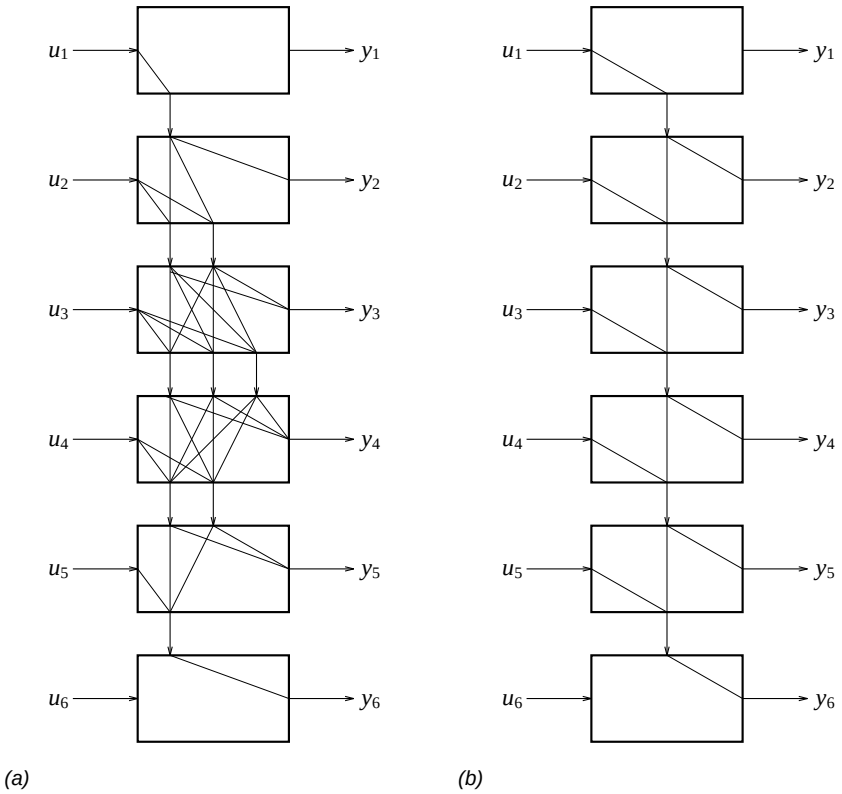

Figure 3. Computational scheme (*a*) of T and (*b*) of T_a.

the space of lower tableaus in $\mathcal{X}(\mathcal{M},\mathcal{N})$, and by $\mathcal{D}(\mathcal{M},\mathcal{N})$ the space of diagonals. Note that if $F \in \mathcal{U}$ is invertible, its inverse is not necessarily in $\mathcal{U}$ (unlike with ordinary matrices), as is demonstrated for example by

$$F = \begin{array}{c} \\ \mathbb{C}^2 \\ \mathbb{C} \\ \varnothing \end{array} \begin{array}{c} \begin{array}{ccc} \mathbb{C} & \mathbb{C} & \mathbb{C} \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 0 & \underline{1/2} & 1 \\ \cdot & \cdot & \underline{\cdot} \end{bmatrix} \end{array} \qquad F^{-1} = \begin{array}{c} \\ \mathbb{C} \\ \mathbb{C} \\ \mathbb{C} \end{array} \begin{array}{c} \begin{array}{ccc} \mathbb{C}^2 & \mathbb{C} & \varnothing \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & \cdot \\ -1/2 & 1 & \underline{0} & \cdot \\ 1/4 & -1/2 & 1 & \underline{\cdot} \end{bmatrix} \end{array}$$

When viewed as matrices, F^{-1} is of course just the matrix inverse of F.

A rightward shifted space sequence is denoted by $\mathcal{B}^{(k)}$, as in

$$\mathcal{B}^{(1)} = \varnothing \times \mathcal{B}_1 \times \mathcal{B}_2 \times \cdots \times \mathcal{B}_n \,. \tag{2.1}$$

The shift operator Z shifts a sequence to the right and is a map $\mathcal{B} \to \mathcal{B}^{(1)}$, with tableau

$$Z = \begin{array}{c} \\ \mathcal{B}_1 \\ \mathcal{B}_2 \\ \vdots \\ \mathcal{B}_n \end{array} \begin{array}{c} \begin{array}{ccccc} \varnothing & \mathcal{B}_1 & \mathcal{B}_2 & \cdots & \mathcal{B}_n \end{array} \\ \begin{bmatrix} \cdot & I & & & \\ \cdot & \underline{0} & I & & \\ & & \ddots & \ddots & \\ \cdot & & & \underline{0} & I \end{bmatrix} \end{array}$$

It is unitary: $ZZ^* = I_{\mathcal{B}}$, $Z^*Z = I_{\mathcal{B}^{(1)}}$. We denote by $Z^{[k]}$ the product of k shifts. It is a map $\mathcal{B} \to \mathcal{B}^{(k)}$. Let

$T \in \mathcal{U}(\mathcal{M}, \mathcal{N})$ be an $n \times n$ tableau. We can decompose T into a sum of shifted diagonals:

$$T = \sum_{k=0}^{n-1} Z^{[k]} T_{[k]} ,$$

where $T_{[k]} \in \mathcal{D}(\mathcal{M}^{(k)}, \mathcal{N})$ is the k-th diagonal above the main (0-th) diagonal. Given a diagonal $A \in \mathcal{D}$, we can write $A = \mathrm{diag}(A_i)$, where the A_i are the diagonal entries of A. Its k-th shift into the South-East direction is defined by $A^{(k)} = (Z^{[k]})^* A Z^{[k]}$, so that $A_i^{(k)} = A_{i-k}$.

We define $\mathbf{P}$ as the projection of $\mathcal{X}$ onto $\mathcal{U}$, $\mathbf{P}_{Z\mathcal{U}}$ the projection onto strictly upper matrices, and $\mathbf{P}_0$ as the projection of $\mathcal{X}$ onto $\mathcal{D}$. With regard to matrix norms, $|| T ||$ is the operator norm (matrix 2-norm), $|| T ||_F$ is the Frobenius norm, and $|| T ||_H$ is the Hankel norm, defined respectively by

$$\begin{aligned} || T || &= \sup_{|| u ||_2 \le 1} || uT ||_2 \\ || T ||_F &= \left\{ \sum || T_{i,j} ||^2 \right\}^{1/2} \\ || T ||_H &= \sup_{U \in \mathcal{L}Z^{-1},\, || U ||_F \le 1} || \mathbf{P}(UT) ||_F . \end{aligned}$$

Note that the above definition of the Hankel norm is equivalent to the definition in (1.4). We remark that this norm is only a norm on the space $Z\mathcal{U}$, while on $\mathcal{X}$ it is a semi-norm. We will also employ a new norm, which we call the *diagonal 2-norm.* Let T_i be the i-th row of a tableau $T \in \mathcal{X}$, then

$$\begin{aligned} &D \in \mathcal{D} : \ || D ||_{\mathcal{D}2} = \sup_i || D_i ||, \\ &T \in \mathcal{X} : \ || T ||_{\mathcal{D}2}^2 = || \mathbf{P}_0(TT^*) ||_{\mathcal{D}2} = \sup_i || T_i T_i^* || . \end{aligned}$$

For diagonals, it is equal to the operator norm, but for more general matrices, it is the supremum over the vector 2-norms of each row of T.

Proposition 2.1. *The Hankel norm satisfies the following ordering:*

$$T \in \mathcal{X} : \qquad || T ||_H \le || T || \tag{2.2}$$

$$T \in Z\mathcal{U} : \qquad || T ||_{\mathcal{D}2} \le || T ||_H . \tag{2.3}$$

PROOF The first norm inequality is proven by

$$\begin{aligned} || T ||_H &= \sup_{U \in \mathcal{L}Z^{-1}, || U ||_F \le 1} && || \mathbf{P}(UT) ||_F \\ &\le \sup_{U \in \mathcal{L}Z^{-1}, || U ||_F \le 1} && || UT ||_F \\ &\le \sup_{U \in \mathcal{X}, || U ||_F \le 1} && || UT ||_F = || T || . \end{aligned}$$

For the second norm inequality, we first prove $|| T ||_{\mathcal{D}2}^2 \le \sup_{D \in \mathcal{D}, || D ||_F \le 1} || DTT^* D^* ||_F$. Indeed,

$$\begin{aligned} || T ||_{\mathcal{D}2}^2 &= || \mathbf{P}_0(TT^*) ||_{\mathcal{D}2}^2 \\ &= \sup_{D \in \mathcal{D}, || D ||_{\mathcal{D}2} \le 1} && || D\mathbf{P}_0(TT^*)D^* ||_{\mathcal{D}2} \\ &= \sup_{D \in \mathcal{D}, || D ||_F \le 1} && || D\mathbf{P}_0(TT^*)D^* ||_F \\ &\le \sup_{D \in \mathcal{D}, || D ||_F \le 1} && || DTT^* D^* ||_F . \end{aligned}$$

Then (2.3) is proven, with use of the fact that $T \in Z\mathcal{U}$:

$$\begin{aligned} || T ||_{\mathcal{D}2}^2 &\le \sup_{D \in \mathcal{D}, || D ||_F \le 1} && || DTT^* D^* ||_F \\ &= \sup_{D \in \mathcal{D}, || D ||_F \le 1} && || DZ^* TT^* ZD^* ||_F \\ &= \sup_{D \in \mathcal{D}, || D ||_F \le 1} && || \mathbf{P}(DZ^* T) \left[\mathbf{P}(DZ^* T) \right]^* ||_F \\ &\le \sup_{U \in \mathcal{L}Z^{-1}, || U ||_F \le 1} && || \mathbf{P}(UT) [\mathbf{P}(UT)]^* ||_F \\ &= || T ||_H^2 . \end{aligned}$$

■

We see that the Hankel norm is not as strong as the operator norm, but is stronger than the row-wise uniform least square norm.

2.2. Realizations

For a given $T \in \mathcal{U}(\mathcal{M}, \mathcal{N})$, a computational model is defined by the sequence of matrices $\{A_k, B_k, C_k, D_k\}$ in the form given by equation (1.1). Let the state $x_k \in \mathcal{B}_k$. We can assemble the matrices $\{A_k\}$, $\{B_k\}$ etc. into diagonals, by defining

$$\begin{array}{rclcrcl} A & \in & \mathcal{D}(\mathcal{B}, \mathcal{B}^{(-1)}) = \mathrm{diag}(A_k)\,, & \qquad & C & \in & \mathcal{D}(\mathcal{B}, \mathcal{N}) = \mathrm{diag}(C_k)\,, \\ B & \in & \mathcal{D}(\mathcal{M}, \mathcal{B}^{(-1)}) = \mathrm{diag}(B_k)\,, & & D & \in & \mathcal{D}(\mathcal{M}, \mathcal{N}) = \mathrm{diag}(D_k)\,, \end{array} \tag{2.4}$$

which together constitute a realization $\mathbf{T}$ of T,

$$y = uT \quad \Leftrightarrow \quad \left\{ \begin{array}{rcl} xZ^{-1} & = & xA + uB \\ y & = & xC + uD \end{array} \right. \qquad \mathbf{T} = \left[\begin{array}{cc} A & C \\ B & D \end{array} \right] \tag{2.5}$$

This description is equivalent to (1.1), but often more convenient to handle because the time-index has been suppressed. Substitution leads to

$$T = D + BZ(I - AZ)^{-1}C\,,$$

where $(I - AZ)^{-1}$ satisfies the expansion

$$\begin{array}{rcl} (I - AZ)^{-1} & = & I + AZ + AZAZ + \cdots \\ & = & I + AZ + AA^{(-1)}Z^{[2]} + AA^{(-1)}A^{(-2)}Z^{[3]} + \cdots\,. \end{array}$$

As we will assume throughout the paper that the realization starts and ends with empty state spaces, this summation is in fact finite: $AA^{(-1)} \cdots A^{(-n)} = [\,\cdot\,]$, where n is the size of T. Hence $(I - AZ)^{-1}$ always exists and the expression for T is meaningful.

Connected to a state realization, we can distinguish global controllability and observability operators defined as

$$\mathcal{C} := \left[\begin{array}{l} B^{(+1)} \\ B^{(+2)}A^{(+1)} \\ B^{(+3)}A^{(+2)}A^{(+1)} \\ \vdots \end{array} \right] \qquad \mathcal{O} := \left[C \quad AC^{(-1)} \quad AA^{(-1)}C^{(-2)} \quad \cdots \right]\,. \tag{2.6}$$

$\mathcal{C}_k$ and $\mathcal{O}_k$ as in equation (1.3) are obtained as the k-th (block) column and row of $\mathcal{C}$ and $\mathcal{O}$, respectively. Recall that $\mathcal{C}_k$ and $\mathcal{O}_k$ are closely related to the Hankel operator: its k-th "snapshot" H_k has the decomposition

$$H_k \;=\; \mathcal{C}_k\mathcal{O}_k\,.$$

We say that the realization is controllable when the controllability operator $\mathcal{C}$ is such that the diagonal matrix $M := \mathcal{C}^*\mathcal{C}$, is invertible, *i.e.*, each $M_k = \mathcal{C}_k^*\mathcal{C}_k$ is invertible. Likewise, the realization is observable if $Q := \mathcal{O}\mathcal{O}^*$ is invertible. In the present context, it is always possible to choose the realization to be both controllable and observable, in which case the realization is also minimal, in the sense that the dimensions of the state space at each point k in the sequence is minimal. For such realizations, the rows of $\mathcal{O}_k$ form a (minimal) basis for the row space of H_k, and the columns of $\mathcal{C}_k$ form a basis for its column space. $\mathcal{C}$ and $\mathcal{O}$ can be thought of as a collection of these bases into a single object.

Another notion that we will need is that of "state transformations". If $\{A, B, C, D\}$ is a realization of a system with transfer matrix T, then an equivalent realization is found by applying a state transformation $\hat{x} = xR$ on the state sequence x of the system, where R is an invertible diagonal matrix. The realization matrix $\mathbf{T}$ is then transformed to

$$\mathbf{T}' = \left[\begin{array}{cc} R & \\ & I \end{array} \right] \left[\begin{array}{cc} A & C \\ B & D \end{array} \right] \left[\begin{array}{cc} (R^{(-1)})^{-1} & \\ & I \end{array} \right]\,.$$

(Note the diagonal shift in $(R^{(-1)})^{-1}$). State transformations are often used to bring a realization into some desirable form. This then leads to equations of the famous Lyapunov or Lyapunov-Stein type. For example, the Lyapunov equation

$$M^{(-1)} = A^* M A + B^* B\,, \qquad M \in \mathcal{D}(\mathcal{B}, \mathcal{B}) \tag{2.7}$$

arises in the transformation of a controllable realization to input normal form: one for which $A^*A+B^*B = I$. If the original realization is controllable, then an invertible state transformator R can be found such that $A_1 = RA(R^{(-1)})^{-1}$, $B_1 = B(R^{(-1)})^{-1}$ and

$$A_1^* A_1 + B_1^* B_1 = I\,.$$

Substitution leads to equation (2.7), with $M = R^*R$, and hence it suffices to solve this equation for M and to verify that M is invertible, in which case a factor R is invertible too. Since equation (2.7) only involves diagonals, it can be solved recursively: $M_{k+1} = A_k^* M_k A_k + B_k^* B_k$, where the initial value is $M_1 = [\,\cdot\,]$. Finally, if $\mathcal{C}$ is the controllability operator of the given realization, then $M = \mathcal{C}^*\mathcal{C}$ is the solution of (2.7), which shows that M is invertible if the realization is controllable. Likewise, if the realization is observable ($\mathcal{O}$ is such that $Q = \mathcal{O}\mathcal{O}^*$ is invertible), then Q is the unique solution of the Lyapunov equation

$$Q = AQ^{(-1)}A^* + CC^*$$

and with the factoring of $Q = RR^*$ this yields an invertible state transformation R such that $A_1 = R^{-1}AR^{(-1)}$, $B_1 = BR^{(-1)}$, $C_1 = R^{-1}C$, and

$$A_1 A_1^* + C_1 C_1^* = I\,.$$

The resulting $\{A_1, B_1, C_1, D\}$ then form an output normal realization for the matrix. In section 3.3 we will assume that the matrix to be approximated is indeed specified by a realization in output normal form, which is automatically the case if the realization algorithm (1) has been used.

2.3. *J*-unitary matrices

If a matrix is at the same time unitary and upper (with respect to its block structure), we will call it inner. In this paper we will make extensive use of matrices Θ that are block upper and J-unitary. To introduce these matrices properly, we must define a splitting of the sequence of input spaces into two sequences $\mathcal{M}_1$ and $\mathcal{N}_1$, a splitting of the sequence of output spaces into two sequences $\mathcal{M}_2$ and $\mathcal{N}_2$, and signature sequences J_1 and J_2:

$$\Theta = \begin{bmatrix} \Theta_{11} & \Theta_{12} \\ \Theta_{21} & \Theta_{22} \end{bmatrix}, \qquad J_1 = \begin{bmatrix} I_{\mathcal{M}_1} & \\ & -I_{\mathcal{N}_1} \end{bmatrix}, \qquad J_2 = \begin{bmatrix} I_{\mathcal{M}_2} & \\ & -I_{\mathcal{N}_2} \end{bmatrix}. \tag{2.8}$$

Θ decomposes in four blocks, mapping $\mathcal{M}_1 \times \mathcal{N}_1$ to $\mathcal{M}_2 \times \mathcal{N}_2$. If each of these maps are upper, we say that Θ is block-upper. Θ will be called J-unitary relative to this splitting in blocks, when

$$\Theta^* J_1 \Theta = J_2 \quad \text{and} \quad \Theta J_2 \Theta^* = J_1\,. \tag{2.9}$$

A J-unitary matrix Θ can be constructed using a computational model $\boldsymbol{\Theta}$ that is J-unitary in the following sense. Let $\mathcal{B}$ be the state sequence space of a realization $\boldsymbol{\Theta}$, and let $\mathcal{B} = \mathcal{B}_+ \times \mathcal{B}_-$ be a decomposition of $\mathcal{B}$. Define the signature matrix

$$J_{\mathcal{B}} = \begin{bmatrix} I_{\mathcal{B}_+} & \\ & -I_{\mathcal{B}_-} \end{bmatrix}$$

(we call $J_{\mathcal{B}}$ the state signature sequence). A realization $\boldsymbol{\Theta}$ is called J-unitary (with respect to $\{J_{\mathcal{B}}, J_1, J_2\}$) if it satisfies

$$\boldsymbol{\Theta}^* \begin{bmatrix} J_{\mathcal{B}} & \\ & J_1 \end{bmatrix} \boldsymbol{\Theta} = \begin{bmatrix} J_{\mathcal{B}}^{(-1)} & \\ & J_2 \end{bmatrix}, \qquad \boldsymbol{\Theta} \begin{bmatrix} J_{\mathcal{B}}^{(-1)} & \\ & J_2 \end{bmatrix} \boldsymbol{\Theta}^* = \begin{bmatrix} J_{\mathcal{B}} & \\ & J_1 \end{bmatrix}. \tag{2.10}$$

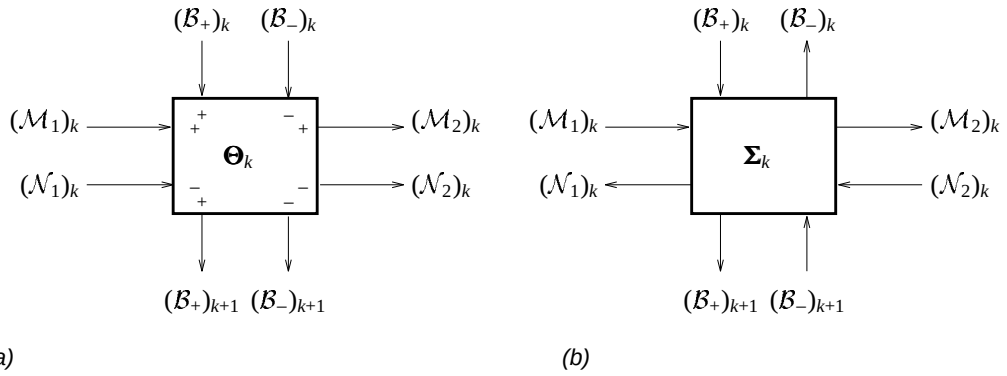

Figure 4. (*a*) The spaces connected with a realization for a *J*-unitary block-upper matrix Θ which transfers $\mathcal{M}_1 \times \mathcal{N}_1$ to $\mathcal{M}_2 \times \mathcal{N}_2$. The realization matrix is marked as $\boldsymbol{\Theta}$. (*b*) The corresponding scattering—or unitary—situation.

Figure 4(*a*) gives a sketch of the situation for the model $\boldsymbol{\Theta}$ associated with Θ.

Proposition 2.2. *If $\boldsymbol{\Theta}$ is a J-unitary realization in the sense of equation (2.10), then the corresponding transfer matrix Θ will be J-unitary in the sense of equation (2.9).*

PROOF This is readily verified by taking as realization for Θ an $\{\alpha, \beta, \gamma, \delta\}$ which satisfies (2.10), and evaluating $J_2 - \Theta^* J_1 \Theta$:

$$\begin{aligned} J_2 - \Theta^* J_1 \Theta &= J_2 - \delta^* J_1 \delta + \gamma^* Z^* (I - \alpha^* Z^*)^{-1} \alpha^* J_{\mathcal{B}} \gamma + \gamma^* J_{\mathcal{B}} \alpha (I - Z\alpha)^{-1} Z\gamma + \\ &\quad - \gamma^* Z^* (I - \alpha^* Z^*)^{-1} \left\{ J_{\mathcal{B}}^{(-1)} - \alpha^* J_{\mathcal{B}} \alpha \right\} (I - Z\alpha)^{-1} Z\gamma \\ &= \gamma^* J_{\mathcal{B}} \gamma + \gamma^* (I - Z^* \alpha^*)^{-1} \left\{ Z^* \alpha^* J_{\mathcal{B}} + J_{\mathcal{B}} \alpha Z - J_{\mathcal{B}} - Z^* \alpha^* J_{\mathcal{B}} \alpha Z \right\} (I - \alpha Z)^{-1} \gamma, \end{aligned}$$

since $\beta^* J_1 \delta = -\alpha^* J_{\mathcal{B}} \gamma$, $\beta^* J_1 \beta = J_{\mathcal{B}}^{(-1)} - \alpha^* J_{\mathcal{B}} \alpha$ and $J_2 - \delta^* J_1 \delta = \gamma^* J_{\mathcal{B}} \gamma$, and hence

$$\begin{aligned} J_2 - \Theta^* J_1 \Theta &= \gamma^* (I - Z^* \alpha^*)^{-1} \left\{ (I - Z^* \alpha^*) J_{\mathcal{B}} (I - \alpha Z) + \right. \\ &\quad \left. + Z^* \alpha^* J_{\mathcal{B}} + J_{\mathcal{B}} \alpha Z - J_{\mathcal{B}} - Z^* \alpha^* J_{\mathcal{B}} \alpha Z \right\} (I - \alpha Z)^{-1} \gamma \\ &= 0 . \end{aligned}$$

The second equality of (2.9) follows by an analogous procedure as above. ∎

A *J*-unitary upper matrix has the following special property.

Proposition 2.3. *If $\{\alpha, \beta, \gamma, \delta\}$ is an observable realization for a J-unitary block-upper matrix Θ, then*

$$Z^* (I - \alpha^* Z^*)^{-1} \beta^* J_1 \Theta \in [\mathcal{U} \quad \mathcal{U}] \tag{2.11}$$

that is, $Z^*(I - \alpha^* Z^*)^{-1} \beta^* J_1$, which is a strictly lower matrix, is mapped by Θ to a block upper matrix.

PROOF Evaluation of the first part of equation (2.9) reveals that

$$\begin{aligned} &Z^* (I - \alpha^* Z^*)^{-1} \beta^* J_1 \Theta \\ &\quad = Z^* (I - \alpha^* Z^*)^{-1} \beta^* J_1 \left(\delta + \beta Z (I - \alpha Z)^{-1} \gamma \right) \\ &\quad = Z^* (I - \alpha^* Z^*)^{-1} \left\{ -\alpha^* J_{\mathcal{B}} + (J_{\mathcal{B}}^{(-1)} - \alpha^* J_{\mathcal{B}} \alpha) Z (I - \alpha Z)^{-1} \right\} \gamma \\ &\quad = (Z - \alpha^*)^{-1} \left\{ -\alpha^* J_{\mathcal{B}} (I - \alpha Z) + J_{\mathcal{B}}^{(-1)} Z - \alpha^* J_{\mathcal{B}} Z \right\} (I - \alpha Z)^{-1} \gamma \\ &\quad = J_{\mathcal{B}} (I - \alpha Z)^{-1} \gamma \quad \in [\mathcal{U} \quad \mathcal{U}] . \end{aligned} \tag{2.12}$$

∎

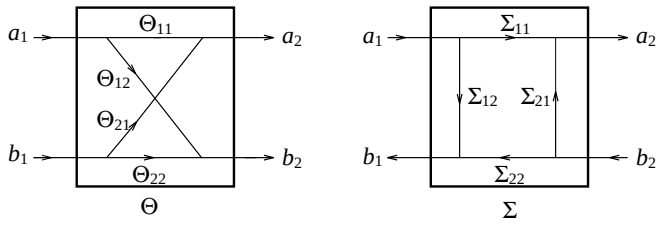

Figure 5. Relation between a J-unitary matrix Θ and the corresponding unitary matrix Σ.

Proposition 2.3 can be interpreted as a general "interpolation principle" which will be treated in detail in sections 3.1–3.3.

Another property that follows from the J-unitarity of Θ is that Θ_{22} is invertible. Associated to Θ is a matrix Σ,

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

which is such that

$$[a_1 \quad b_2]\Sigma = [a_2 \quad b_1] \qquad \Leftrightarrow \qquad [a_1 \quad b_1]\Theta = [a_2 \quad b_2],$$

(see figure 5), that is,

$$\Sigma = \begin{bmatrix} I & -\Theta_{12} \\ 0 & I \end{bmatrix} \begin{bmatrix} \Theta_{11} & 0 \\ 0 & \Theta_{22}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ \Theta_{21} & I \end{bmatrix} = \begin{bmatrix} \Theta_{11} - \Theta_{12}\Theta_{22}^{-1}\Theta_{21} & -\Theta_{12}\Theta_{22}^{-1} \\ \Theta_{22}^{-1}\Theta_{21} & \Theta_{22}^{-1} \end{bmatrix} \tag{2.13}$$

It is straightforward to prove that from the J-unitarity of Θ it follows that Σ is unitary. Σ is known as a *scattering matrix*, while Θ is called a *chain scattering matrix*. Σ and Θ constitute the same linear relations between the quantities a_1, a_2, b_1, b_2. However, the signal flows of the "incident" and "reflected" waves of Σ coincide with the direction of the energy going into and out of the system: $a_1 a_1^* + b_2 b_2^* = a_2 a_2^* + b_1 b_1^*$, whereas for Θ the relation $a_1 a_1^* - b_1 b_1^* = a_2 a_2^* - b_2 b_2^*$ reflects conservation of energy between port 1 and port 2.

Let $\boldsymbol{\Theta}$ be a J-unitary realization. Since each of the $\boldsymbol{\Theta}_k$ is a J-unitary matrix, there is a unitary matrix $\boldsymbol{\Sigma}_k$ associated to each $\boldsymbol{\Theta}_k$ in the same way as Σ followed from Θ, but now according to the rule

$$\begin{aligned} & [x_+ \quad x_- \quad a_1 \quad b_1]\,\boldsymbol{\Theta} \;\; = \;\; [x_+^{(-1)} \quad x_-^{(-1)} \quad a_2 \quad b_2] \\ \Leftrightarrow \quad & [x_+ \quad x_-^{(-1)} \quad a_1 \quad b_2]\,\boldsymbol{\Sigma} \;\; = \;\; [x_+^{(-1)} \quad x_- \quad a_2 \quad b_1] \end{aligned} \tag{2.14}$$

(that is, inputs of $\boldsymbol{\Sigma}$ have positive signature). Again, the directions of the arrows corresponding to negative signatures in $\boldsymbol{\Theta}$ is reversed (see figure 4(*b*)). An explicit formula for $\boldsymbol{\Sigma}$ in terms of $\boldsymbol{\Theta}$ is given below. Although $\boldsymbol{\Sigma}$ constitutes the same linear relations between the state variables as $\boldsymbol{\Theta}$, and hence elimination of x_+ and x_- will lead to the scattering matrix Σ associated to Θ, it should be noted that $\boldsymbol{\Sigma}$ is not a realization of Σ, since the state flow is not uni-directional: the next state of $\boldsymbol{\Sigma}$ is specified in terms of its current state only in an implicit way. $\boldsymbol{\Sigma}$ will be called a state representation of Σ, rather than a realization.

$\boldsymbol{\Sigma}$ is computed from Θ in the following way. Partition the state x of $\boldsymbol{\Theta}$ according to the signature $J_{\mathcal{B}}$ into $x = [x_+ \;\; x_-]$, and partition $\boldsymbol{\Theta}$ likewise:

$$\boldsymbol{\Theta} = \begin{array}{c} \\ x_+ \\ x_- \\ a_1 \\ b_1 \end{array} \begin{array}{c} \begin{array}{cccc} x_+^{(-1)} & x_-^{(-1)} & a_2 & b_2 \end{array} \\ \left[\begin{array}{cc|cc} \alpha_{11} & \alpha_{12} & \gamma_{11} & \gamma_{12} \\ \alpha_{21} & \alpha_{22} & \gamma_{21} & \gamma_{22} \\ \hline \beta_{11} & \beta_{12} & \delta_{11} & \delta_{12} \\ \beta_{21} & \beta_{22} & \delta_{21} & \delta_{22} \end{array}\right] \end{array}, \tag{2.15}$$

then the corresponding $\mathbf{\Sigma}$, defined by the relation (2.14) has a partitioning

$$\mathbf{\Sigma} = \begin{array}{c} \\ x_+ \\ x_-^{(-1)} \\ a_1 \\ b_2 \end{array} \begin{array}{c} \begin{array}{cc} x_+^{(-1)} & x_- \end{array} \\ \left[\begin{array}{cc|cc} F_{11} & F_{12} & H_{11} & H_{12} \\ F_{21} & F_{22} & H_{21} & H_{22} \\ \hline G_{11} & G_{12} & K_{11} & K_{12} \\ G_{21} & G_{22} & K_{21} & K_{22} \end{array}\right] \end{array} \begin{array}{c} \\ \\ \\ a_2 \\ b_1 \end{array} . \tag{2.16}$$

First, we prove the existence of $\mathbf{\Sigma}$ by remarking that, because of the J-unitarity of $\mathbf{\Theta}$, the submatrix

$$\begin{bmatrix} \alpha_{22} & \gamma_{22} \\ \beta_{22} & \delta_{22} \end{bmatrix}$$

is at each point k square and invertible. The entries in $\mathbf{\Sigma}$ can be determined from those of $\mathbf{\Theta}$ as

$$\begin{aligned}
\begin{bmatrix} F_{11} & H_{11} \\ G_{11} & K_{11} \end{bmatrix} &= \begin{bmatrix} \alpha_{11} & \gamma_{11} \\ \beta_{11} & \delta_{11} \end{bmatrix} - \begin{bmatrix} \alpha_{12} & \gamma_{12} \\ \beta_{12} & \delta_{12} \end{bmatrix} \begin{bmatrix} \alpha_{22} & \gamma_{22} \\ \beta_{22} & \delta_{22} \end{bmatrix}^{-1} \begin{bmatrix} \alpha_{21} & \gamma_{21} \\ \beta_{21} & \delta_{21} \end{bmatrix} \\
\begin{bmatrix} F_{12} & H_{12} \\ G_{12} & K_{12} \end{bmatrix} &= - \begin{bmatrix} \alpha_{12} & \gamma_{12} \\ \beta_{12} & \delta_{12} \end{bmatrix} \begin{bmatrix} \alpha_{22} & \gamma_{22} \\ \beta_{22} & \delta_{22} \end{bmatrix}^{-1} \\
\begin{bmatrix} F_{21} & H_{21} \\ G_{21} & K_{21} \end{bmatrix} &= \begin{bmatrix} \alpha_{22} & \gamma_{22} \\ \beta_{22} & \delta_{22} \end{bmatrix}^{-1} \begin{bmatrix} \alpha_{21} & \gamma_{21} \\ \beta_{21} & \delta_{21} \end{bmatrix} \\
\begin{bmatrix} F_{22} & H_{22} \\ G_{22} & K_{22} \end{bmatrix} &= \begin{bmatrix} \alpha_{22} & \gamma_{22} \\ \beta_{22} & \delta_{22} \end{bmatrix}^{-1}
\end{aligned} \tag{2.17}$$

(cf. equation (2.13)). Note that each matrix $\mathbf{\Sigma}_k$ only depends on the entries of $\mathbf{\Theta}_k$ so that it can be computed independently from the other stages.

3. CONSTRUCTION OF A HANKEL-NORM APPROXIMANT

3.1. Summary of the procedure

In this section, we solve the Hankel norm model reduction problem for a strictly upper matrix described by a "higher order model" with an observable realization $\{A, B, C, 0\}$. Let input and output spaces $\mathcal{M}$ and $\mathcal{N}$ be as in equation (2.4), and let Γ be a diagonal and hermitian matrix belonging to $\mathcal{D}(\mathcal{M}, \mathcal{M})$. We use Γ as a measure for the local accuracy of the reduced order model; it will parametrize the solutions. We look for a matrix $T' \in \mathcal{X}(\mathcal{M}, \mathcal{N})$ such that (*i*) the scaled difference with T is smaller than 1 in operator norm:

$$\| \Gamma^{-1}(T - T') \| \leq 1 , \tag{3.1}$$

and such that (*ii*) the approximant

$$T_a := \mathbf{P}_{Z\mathcal{U}}(T') , \tag{3.2}$$

i.e., the strictly upper part of T', has a state dimension sequence of low order — as low as possible for a given Γ. Using the norm inequality (2.2), we immediately obtain that T_a satisfies

$$\| \Gamma^{-1}(T - T_a) \|_H \leq \| \Gamma^{-1}(T - T') \| \leq 1 ,$$

i.e., T_a is a Hankel-norm approximant of T when T' is an operator-norm approximant. The second norm inequality (2.3) gives in addition

$$\| \Gamma^{-1}(T - T_a) \|_{\mathcal{D}2} \leq \| \Gamma^{-1}(T - T_a) \|_H .$$

The interpretation of this second inequality is that the change in each row of $\Gamma^{-1}T$ is (in 2-norm) smaller than the error in Hankel norm, and at least smaller than 1. A comparable result holds for the columns of $\Gamma^{-1}T$. Consequently, the matrix entries of a Hankel norm approximant T_a are close to those of T.

The construction of a matrix T' satisfying (3.1) consists of the following three steps. We start by computing a factorization of T in the form

$$T = \Delta^* U \tag{3.3}$$

where Δ and U are upper matrices which have state space dimensions of the same size as that of T, and U is inner. We will call such a factorization an *external* factorization. We show in section 3.2 that this factorization is easy to determine if the realization (2.4) for T is chosen to be in output normal form, *i.e.*, such that $AA^* + CC^* = I$. The construction of a proper T' continues by the determination of a matrix Θ that is J-unitary as in (2.8) and block-upper, such that

$$\begin{bmatrix} U^* & -T^*\Gamma^{-1} \end{bmatrix} \Theta = \begin{bmatrix} A' & -B' \end{bmatrix} \tag{3.4}$$

consists of two upper matrices A' and B'. As an aside, we remark that this expression can equivalently be written as

$$U^* \begin{bmatrix} I & -\Delta\Gamma^{-1} \end{bmatrix} \Theta = \begin{bmatrix} A' & -B' \end{bmatrix} . \tag{3.5}$$

which is the 'standard' formulation of an interpolation problem (see [24]): for time-invariant systems, the equation expresses that $\begin{bmatrix} I & -\Delta\Gamma^{-1} \end{bmatrix} \Theta$ has zeros at poles of U^*, which ensures that the approximant is equal to the original system at certain points in the z-plane.

We will show that a solution to this interpolation problem exists if certain conditions on a Lyapunov equation associated to $\Gamma^{-1}T$ are satisfied (this can always be the case for judiciously chosen Γ). The state dimension of Θ will again be the same as that of T. Because Θ is J-unitary, we have that $\Theta_{22}^*\Theta_{22} = I + \Theta_{12}^*\Theta_{12}$. Hence Θ_{22}^{-1} will exist (but will not necessarily be upper) and $\Sigma_{12} = -\Theta_{12}\Theta_{22}^{-1}$ will be contractive. From (3.4) we have $B' = -U^*\Theta_{12} + T^*\Gamma^{-1}\Theta_{22}$. In terms of the definition of Θ and B', the approximating matrix T' is subsequently defined as

$$T' = \Gamma\Theta_{22}^{-*}B'^* . \tag{3.6}$$

Then the resulting approximation error is $\Gamma^{-1}(T - T') = -\Sigma_{12}^* U$. Because Σ_{12} is contractive and U unitary, we infer that $\| \Gamma^{-1}(T - T') \| \leq 1$, so that T' is indeed an operator-norm approximant with an admissible modeling error. Taking T_a equal to the upper triangular part of T', the definitions (3.4), (3.6) and (3.2) result in a Hankel norm approximant T_a. We will also show that, from (3.6) and the fact that B' is upper triangular, it can be inferred that the state dimension of T_a will, at each point in time, be at most equal to that of the upper part of Θ_{22}^{-*}. (With more effort, one shows that the state dimensions are precisely equal to each other [26].) In view of the target theorem 1.2, it remains (1) to construct U, (2) to construct Θ satisfying (3.4), taking care that the upper part of Θ_{22}^{-*} has state dimensions as low as possible, and (3) to verify the complexity of the Hankel norm approximant in connection with the Hankel singular values of $\Gamma^{-1}T$. These are the subjects of the following sections. Subsequently, formulas describing a realization of T_a are derived (theorem 3.7).

3.2. External factorization of T

The aim of this section is to prove the following proposition.

Proposition 3.1. *If a matrix T is upper, $T \in \mathcal{U}(\mathcal{M}, \mathcal{N})$, then there exists a space sequence $\mathcal{M}_U$ and an inner matrix $U \in \mathcal{U}(\mathcal{M}_U, \mathcal{N})$ such that $\Delta = UT^*$ is upper, and T has a factorization*

$$T = \Delta^* U .$$

PROOF To obtain U, we start from a model $\{A, B, C, D\}$ of T which is in output normal form, $A_kA_k^* + C_kC_k^* = I$ for all k. It is obtained, for example, by the realization algorithm (1). For each point k, determine matrices $B_{U,k}$ and $D_{U,k}$ via the orthogonal complement of the rows of $[A_k \;\; C_k]$, so that $\mathbf{U}_k$,

$$\mathbf{U}_k \;=\; \begin{array}{c} \\ \mathcal{B}_k \\ \mathcal{M}_{U,k} \end{array} \begin{array}{c} \begin{array}{cc} \mathcal{B}_{k+1} & \mathcal{N}_k \end{array} \\ \begin{bmatrix} A_k & C_k \\ B_{U,k} & D_{U,k} \end{bmatrix} \end{array},$$

is a square and unitary matrix. Take $\mathbf{U}$ to be a computational model for U. Then U is inner, because its realization is unitary (proposition 2.2). It remains to verify that $\Delta = UT^*$ is upper. This follows by direct computation of Δ, in which we make use of the relations $AA^* + CC^* = I$, $B_UA^* + D_UC^* = 0$:

$$\begin{aligned} \Delta = UT^* \;&=\; \left[D_U + B_UZ(I-AZ)^{-1}C\right]\left[D^* + C^*(I-Z^*A^*)^{-1}Z^*B^*\right] \\ &=\; \left[D_U + B_UZ(I-AZ)^{-1}C\right]D^* \;+\; \underline{D_UC^*}(I-Z^*A^*)^{-1}Z^*B^* \;+ \\ &\quad + B_UZ(I-AZ)^{-1}\underline{CC^*}(I-Z^*A^*)^{-1}Z^*B^* \\ &=\; \left[D_U + B_UZ(I-AZ)^{-1}C\right]D^* \;-\; B_UA^*(I-Z^*A^*)^{-1}Z^*B^* \;+ \\ &\quad + B_UZ(I-AZ)^{-1}\,(I-AA^*)\,(I-Z^*A^*)^{-1}Z^*B^* . \end{aligned}$$

Now, we make use of the relation

$$Z(I-AZ)^{-1}(I-AA^*)(I-Z^*A^*)^{-1}Z^* = (I-ZA)^{-1} \;+\; A^*(I-Z^*A^*)^{-1}Z^* ,$$

which is easily verified by pre- and postmultiplying with $(I - ZA)$ and $(Z - A^*)$, respectively. Plugging this relation into the expression for Δ, it is seen that the lower triangular parts of the expression cancel, and we obtain

$$\begin{aligned} \Delta \;&=\; \left[D_U + B_UZ(I-AZ)^{-1}C\right]D^* + B_U(I-ZA)^{-1}B^* \\ &=\; D_UD^* + B_UB^* + B_UZ(I-AZ)^{-1}(AB^* + CD^*) . \end{aligned}$$

which is, indeed, upper. ∎

Because the A_k are not necessarily square matrices, the dimension of the state space may vary in time. A consequence of this will be that the number of inputs of U will vary in time for an inner U having minimal state dimension. The varying number of inputs of U will of course be matched by a varying number of outputs of Δ^*. Figure 6 illustrates this point.

3.3. Determination of Θ

In this section we will show how, under satisfaction of a condition of Lyapunov type, equation (3.4) can be satisfied with a J-unitary transfer matrix Θ. Let T be a strictly upper matrix with model $\{A, B, C, 0\}$ in output normal form, and let $\{A, B_U, C, D_U\}$ be the unitary realization for the inner factor $U \in \mathcal{U}(\mathcal{M}_U, \mathcal{N})$ of T. Denote by $\mathcal{B}$ the state sequence space of $\mathbf{T}$. We submit that Θ satisfying (3.4) has a realization $\boldsymbol{\Theta}$ of the form

$$\begin{aligned} \boldsymbol{\Theta} \;&=\; \left[\begin{array}{c|c} X & \\ \hline & I \end{array}\right] \left[\begin{array}{c|cc} A & C_1 & C_2 \\ \hline B_U & D_{11} & D_{12} \\ \Gamma^{-1}B & D_{21} & D_{22} \end{array}\right] \left[\begin{array}{c|c} (X^{(-1)})^{-1} & \\ \hline & I \end{array}\right] \\ &=:\; \left[\begin{array}{c|cc} \alpha & \gamma_1 & \gamma_2 \\ \hline \beta_1 & \delta_{11} & \delta_{12} \\ \beta_2 & \delta_{21} & \delta_{22} \end{array}\right] = \begin{bmatrix} \alpha & \gamma \\ \beta & \delta \end{bmatrix} \end{aligned} \qquad (3.7)$$

which is a square matrix at each point k, and where the X and C_i, D_{ij} are yet to be determined. Note that the state sequence space $\mathcal{B}$ is the same for $\boldsymbol{\Theta}$ and T. X is an invertible diagonal state transformation matrix which is such that $\boldsymbol{\Theta}$ is J-unitary as in (2.10), where the state signature matrix $J_{\mathcal{B}}$ is also to be determined. The following theorem summarizes what we will prove in this section.

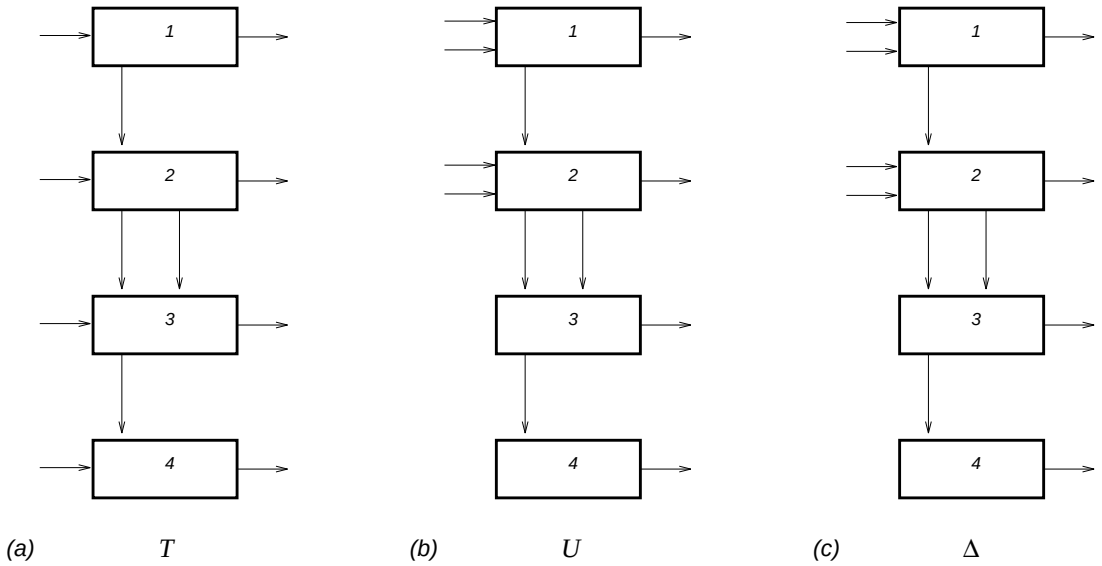

Figure 6. (*a*) The computational scheme for an example *T*, (*b*) the computational structure of the corresponding inner factor *U* and (*c*) of Δ.

Theorem 3.2. *Let $T \in \mathcal{U}(\mathcal{M}, \mathcal{N})$ be a strictly upper matrix, with $\{A, B, C, 0\}$ a model of T in output normal form, and let $\Gamma \in \mathcal{D}(\mathcal{M}, \mathcal{M})$ be an invertible Hermitian diagonal matrix. Let U be the inner factor of an external factorization of T, with unitary model $\{A, B_U, C, D_U\}$. If the solution M of the Lyapunov equation*

$$A^*MA + B^*\Gamma^{-2}B = M^{(-1)} \tag{3.8}$$

is such that $\Lambda = I - M$ is invertible, then there exists a J-unitary block upper matrix Θ such that

$$[U^* \quad -T^*\Gamma^{-1}]\,\Theta \tag{3.9}$$

is block upper. The corresponding J-unitary realization $\boldsymbol{\Theta}$ is of the form (3.7), with state transformation X and state signature matrix $J_{\mathcal{B}}$ given by the factorization $\Lambda = X^*J_{\mathcal{B}}X$.

PROOF We first construct Θ by determining a realization $\boldsymbol{\Theta}$ that has the structure of equation (3.7), and then show that it satisfies (3.9). The first step in solving for the unknowns in (3.7) is to determine X such that

$$\begin{bmatrix} \alpha \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} XA(X^{(-1)})^{-1} \\ B_U(X^{(-1)})^{-1} \\ \Gamma^{-1}B(X^{(-1)})^{-1} \end{bmatrix} \tag{3.10}$$

is J-isometric in the sense of equation (2.10), *i.e.*, such that for some signature matrix $J_{\mathcal{B}}$,

$$\begin{aligned}(X^{(-1)})^{-*}A^*X^*\,J_{\mathcal{B}}\,XA(X^{(-1)})^{-1} \;+\; (X^{(-1)})^{-*}B_U^*B_U(X^{(-1)})^{-1} \;+\\ -\,(X^{(-1)})^{-*}B^*\Gamma^{-2}B(X^{(-1)})^{-1} \;=\; J_{\mathcal{B}}^{(-1)}\,.\end{aligned}$$

Writing $\Lambda = X^*J_{\mathcal{B}}X$, this produces

$$A^*\Lambda A + B_U^*B_U - B^*\Gamma^{-2}B = \Lambda^{(-1)}\,, \tag{3.11}$$

which determines Λ recursively, and hence also both the factor X and the state signature $J_{\mathcal{B}}$. For X to be invertible, it is sufficient to require Λ to be invertible. Equation (3.11) may be rewritten in terms of the original data by using $B_U^*B_U = I - A^*A$, which yields

$$A^*MA + B^*\Gamma^{-2}B = M^{(-1)}\,, \qquad M = I - \Lambda\,.$$

M is the solution of one of the Lyapunov equations associated to $\Gamma^{-1}T$ (viz. equation (2.7)). We proceed with the construction of a realization $\boldsymbol{\Theta}$ of the form (3.7) which satisfies (2.10) for

$$J_1 = \begin{bmatrix} I_{\mathcal{M}_U} & \\ & -I_{\mathcal{M}} \end{bmatrix}, \qquad J_2 := \begin{bmatrix} I_{\mathcal{M}_2} & \\ & -I_{\mathcal{N}_2} \end{bmatrix}$$

where J_2 is still to be determined (and with it the output space sequences $\mathcal{M}_2$ and $\mathcal{N}_2$). Since signature of matrices is conserved under congruence relations as (2.10), we must have that the signatures of the matrices

$$\begin{bmatrix} J_{\mathcal{B}} & \\ & J_1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} J_{\mathcal{B}}^{(-1)} & \\ & J_2 \end{bmatrix}$$

are equal. Let s-dim denote the sequence of dimensions of a non-uniform space (a sequence of integers), and let $\#_+(J)$ denote the sequence whose k-th entry is the number of positive entries in the signature matrix J at point k (and likewise for the number of negative entries $\#_-(J)$), then

$$\begin{aligned} \text{s-dim } \mathcal{M}_2 &= \#_+(J_{\mathcal{B}}) - \#_+(J_{\mathcal{B}}^{(-1)}) + \text{s-dim } \mathcal{M}_U \\ \text{s-dim } \mathcal{N}_2 &= \#_-(J_{\mathcal{B}}) - \#_-(J_{\mathcal{B}}^{(-1)}) + \text{s-dim } \mathcal{M}\,. \end{aligned}$$

The positivity of these dimensions is readily derived from equation (3.11) by Sylvester's inequality.

To obtain $\boldsymbol{\Theta}$, it remains to complete the matrix (3.10) to form the matrix $\boldsymbol{\Theta}$ in (3.7) so that the whole matrix is now J-unitary according to (2.10). This matrix completion can be achieved at the local level: it is for each stage k an independent problem of matrix algebra (see algorithm 2). It is not hard to see that the completion is always possible.

To conclude the proof, we have to show that $[U^* \quad -T^*\Gamma^{-1}]\,\Theta$ is block upper. We have

$$[U^* \quad -T^*\Gamma^{-1}] \;=\; [D_U^* \quad 0] + C^*Z^*(I - A^*Z^*)^{-1}[B_U^* \quad -B^*\Gamma^{-1}] \tag{3.12}$$

and it will be enough to show that

$$Z^*(I - A^*Z^*)^{-1}[B_U^* \quad -B^*\Gamma^{-1}]\,\Theta \tag{3.13}$$

is block upper. With entries as in equation (3.7), and using the state equivalence transformation defined by X, this is equivalent to showing that

$$X^*Z^*(I - \alpha^*Z^*)^{-1}[\beta_1^* \quad \beta_2^*]\,J_1\,\Theta$$

is block-upper. That this is indeed the case follows directly from proposition 2.3—see equation (2.11).

■

For later use, we evaluate $[U^* \quad -T^*\Gamma^{-1}]\,\Theta$. Equation (2.12) gives

$$\begin{aligned} C^*Z^*(I - A^*Z^*)^{-1}[B_U^* \quad -B^*\Gamma^{-1}]\Theta &= C^*X^*Z^*(I - \alpha^*Z^*)^{-1}\beta^*J_1\Theta \\ &= C^*X^*J_{\mathcal{B}}(I - \alpha Z)^{-1}\gamma \\ &= C^*\Lambda(I - AZ)^{-1}[C_1 \quad C_2]\,. \end{aligned}$$

Consequently,

$$\begin{aligned} [U^* \quad -T^*\Gamma^{-1}]\Theta &= [D_U^* \quad 0]\left\{\delta + [B_U^* \quad B^*\Gamma^{-1}]^*Z(I - AZ)^{-1}[C_1 \quad C_2]\right\} + C^*\Lambda(I - AZ)^{-1}[C_1 \quad C_2] \\ &= \{[D_U^* \quad 0]\delta + C^*\Lambda[C_1 \quad C_2]\} + C^*(\Lambda - I)AZ(I - AZ)^{-1}[C_1 \quad C_2] \end{aligned}$$

(in which we used $C^*A + D_U^*B_U = 0$). Since this expression is equal to $[A' \quad -B']$, we obtain a computational model for B' as

$$B' = \left\{-D_U^*D_{12} + C^*(I - M)C_2\right\} + \left\{C^*MA\right\}Z(I - AZ)^{-1}C_2\,. \tag{3.14}$$

In:	$\mathbf{T}$	(model in output normal form for a strictly upper matrix T)
	Γ	(approximation parameters)
Out:	$\boldsymbol{\Theta}$	(realization for Θ satisfying (3.4))

$M_1 = [\,\cdot\,]$
$X_1 = [\,\cdot\,]$
$J_{\mathcal{B}_1} = [\,\cdot\,]$

for $k = 1, \cdots, n$

$$
\left[\begin{array}{l}
M_{k+1} = A_k^* M_k A_k + B_k^* \Gamma_k^{-2} B_k \\
X_{k+1}^* J_{\mathcal{B}_{k+1}} X_{k+1} := I - M_{k+1} \\
[B_{U,k} \quad D_{U,k}] = [A_k \quad C_k]^{\perp} \\
\left[\begin{array}{c} \alpha \\ \beta \end{array}\right] = \left[\begin{array}{c} X_k A_k \\ \hline B_{U,k} \\ \Gamma_k^{-1} B_k \end{array}\right] X_{k+1}^{-1} \\
\left[\begin{array}{c} c \\ d \end{array}\right] = \left[\begin{array}{c} J_{\mathcal{B}_k} \alpha \\ J_1 \beta \end{array}\right]^{\perp} \\
r^* J_2 r := [c^* \quad d^*] \left[\begin{array}{cc} J_{\mathcal{B}_k} & \\ & J_1 \end{array}\right] \left[\begin{array}{c} c \\ d \end{array}\right] \\
\left[\begin{array}{c} \gamma \\ \delta \end{array}\right] = \left[\begin{array}{c} c \\ d \end{array}\right] r^{-1} \\
\boldsymbol{\Theta}_k = \left[\begin{array}{cc} \alpha & \gamma \\ \beta & \delta \end{array}\right]
\end{array}\right.
$$

end

Algorithm 2. The interpolation algorithm.

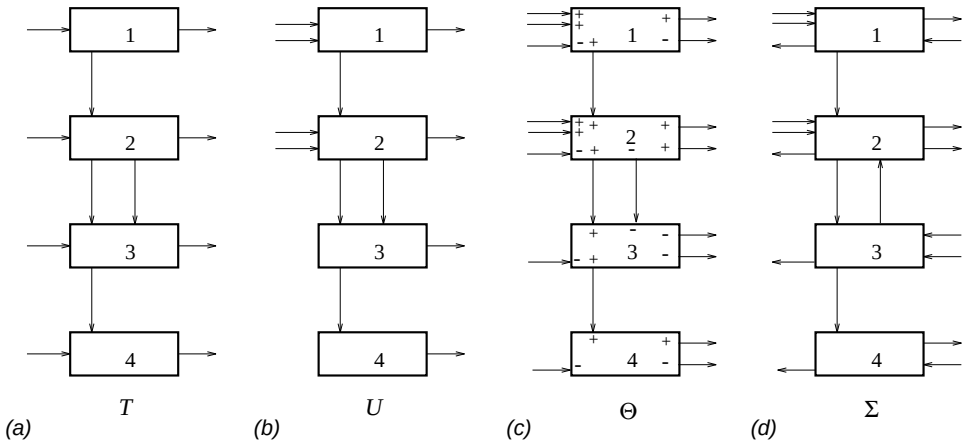

Figure 7. (*a*) State space realization scheme for T and (*b*) for U. (*c*) State space realization scheme for a possible Θ, where it is assumed that one singular value of the Hankel operator of $\Gamma^{-1}T$ at time 1 is larger than 1, and (*d*) for the corresponding scattering operator Σ.

Algorithm 2 summarizes the construction in theorem 3.2 and can be used to compute Θ satisfying equation (3.4). The inner factor U of T is computed *en passant*.

The key to construct the interpolating Θ in (3.4) is hence the solution of the Lyapunov equation (3.8). It can be computed recursively by taking the k-th entry of each diagonal in the equation, yielding

$$M_{k+1} = A_k^* M_k A_k + B_k^* \Gamma_k^{-2} B_k$$

The initial point of this recursion is $M_1 = [\,\cdot\,]$, if the state dimension sequence of the realization of T starts with zero states. We conclude this section by establishing the link between this Lyapunov equation and the Hankel matrix connected with $\Gamma^{-1}T$. This will provide the connection of the Hankel singular values of $\Gamma^{-1}T$ and the state complexity of the Hankel norm approximant, discussed in the next subsection.

Theorem 3.3. *Let $T \in \mathcal{U}(\mathcal{M},\mathcal{N})$ have a model $\{A,B,C,0\}$ in output normal form, and let Γ be an invertible diagonal Hermitian matrix. Let H_k be the Hankel matrix of $\Gamma^{-1}T$ at stage k, and suppose that, for each k, none of the singular values of H_k are equal to 1. Let N_k be the number of singular values of H_k that are larger than 1.*

Then the solution M_k of the Lyapunov recursion

$$M_{k+1} = A_k^* M_k A_k + B_k^* \Gamma_k^{-2} B_k\,, \qquad M_1 = [\,\cdot\,]\,, \tag{3.15}$$

is such that $\Lambda_k = I - M_k$ is invertible and has signature $J_{\mathcal{B}_k}$ having precisely N_k negative entries.

PROOF According to section 2.2, the Hankel matrix H_k of $\Gamma^{-1}T$ at stage k satisfies the decomposition $H_k = \mathcal{C}_k\mathcal{O}_k$, where $\mathcal{C}_k$ and $\mathcal{O}_k$ are given as in (1.3), save for a scaling of B_k by Γ_k^{-1}. Hence

$$H_k H_k^* = \mathcal{C}_k \mathcal{O}_k \mathcal{O}_k^* \mathcal{C}_k^*\,.$$

In the present context we have started from an output normal form: $Q = \mathcal{O}\mathcal{O}^* = I$. The non-zero eigenvalues of $H_k H_k^* = \mathcal{C}_k\mathcal{C}_k^*$ will be the same as those of $\mathcal{C}_k^*\mathcal{C}_k$, and in section 2.2 it was shown that $M_k = \mathcal{C}_k^*\mathcal{C}_k$ is precisely the solution of the Lyapunov recursion (3.15). In particular, the number of singular values of H_k that are larger than 1 is equal to the number of eigenvalues of M_k that are larger than 1. Writing $\Lambda_k = I - M_k$, this is in turn equal to the number of negative eigenvalues of Λ_k. ∎

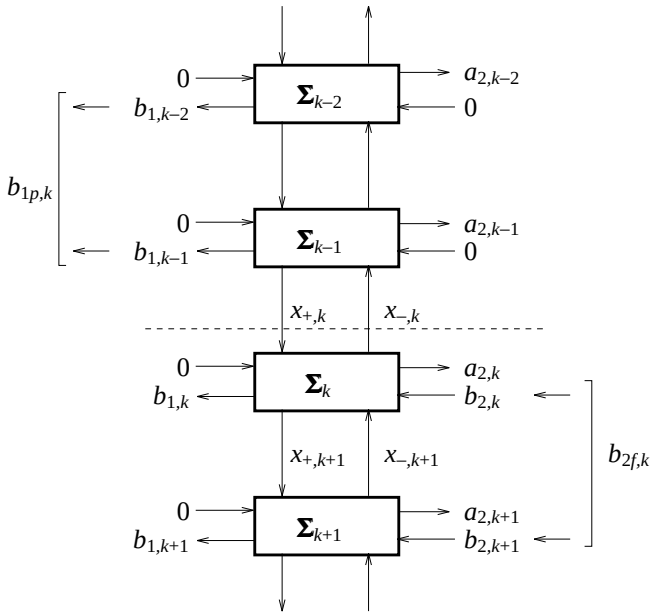

Figure 8. Dataflow scheme for Σ, which shows that $x_{-,k}$ is a state in the transfer $b_{2f,k} \to b_{1p,k}$.

Figure 7 shows a simple instance of the application of the theory developed in this section, especially with regard to the dimensions of the input, output and state sequence spaces related to the Θ-matrix. The signal flow of the state realization matrices $\mathbf{\Theta}_k$ runs strictly from top to bottom and from left to right. Corresponding to Θ is the scattering operator Σ, whose state representation $\mathbf{\Sigma}_k$ is for each k computed from $\mathbf{\Theta}_k$ using equation (2.16). The arrows in the scattering situation (where the signal flow coincides with 'positive energy flow') run in the reverse direction for inputs and outputs of $\mathbf{\Theta}_k$ that have a negative signature. In the figure, we assumed that one singular value of the Hankel operator of $\Gamma^{-1}T$ at time 1 is larger than 1, which results in one state variable with negative signature, and hence there is one upward arrow in the diagram for Σ. Because of the upward arrow, Σ is not an upper matrix (it is not a causal transfer operator), and $\mathbf{\Sigma}$ only specifies Σ implicitly: figure 7(*d*) contains a loop between stage 1 and 2 which renders the network uncomputable. As is shown in the next section, upward arrows generate the states of the Hankel-norm approximant, and the number of upward arrows is equal to the number of states of the approximant.

3.4. State dimension of T_a

At this point we have covered the first part of theorem 1.2: we have constructed a J-unitary Θ and from it a matrix T_a which is a Hankel-norm approximant of T. It remains to verify the complexity assertion, which stated that the dimension of the state space of T_a is at most equal to N_k at point k: the number of singular values of the k-th Hankel matrix of $\Gamma^{-1}T$ that are larger than one, or (by theorem 3.3) the number of negative entries in the state signature $J_{\mathcal{B}}$ of Θ at point k. Not surprisingly from the definition of T_a, an important role will be played by Θ_{22}^{-1}, which is the 22-entry of the scattering matrix Σ associated to Θ by equation (2.13). The representation $\mathbf{\Sigma}$ specifies, be it in an implicit form, the relations between the input and output quantities of the non-causal operator Σ. The existence of Σ implies, e.g., that all intermediate state quantities $x_{+,k}$, $x_{-,k}$ are well-defined, given inputs a_1 and b_2. In particular, $\Sigma_{22} = \Theta_{22}^{-1}$ is obtained by imposing $a_1 = 0$ and looking at the transfer $b_2 \mapsto a_2$. Finding a realization for the strictly upper part of Θ_{22}^{-*} will consist in "unwinding" the loops in the representation $\mathbf{\Sigma}$ of Σ and deducing the realization for it. The fact that Σ can be resolved and that a realization for Θ_{22}^{-*} can be deduced will be the topic of the next section.

In this section, we prove the following proposition, which provides with theorem 3.2 and theorem 3.3 a proof of the Hankel norm approximation theorem (theorem 1.2).

Proposition 3.4. *If the conditions of theorem 3.2 are satisfied, then the state dimension of the approximant T_a is (at most) equal to the state dimension of the strictly upper part of Θ_{22}^{-*} at each point. This dimension is in turn (at most) equal to the number of negative entries in the state signature $J_{\mathcal{B}}$ of Θ at point k, or the number of singular values of the Hankel matrix of $\Gamma^{-1}T$ at point k that are larger than 1.*

PROOF T_a is determined by the definitions (3.2), (3.6):

$$\begin{aligned} T' &= \Gamma\Theta_{22}^{-*}B'^{*} \\ T_a &= \Gamma\mathbf{P}_{\mathcal{ZU}}(T') = \Gamma\mathbf{P}_{\mathcal{ZU}}(\Theta_{22}^{-*}B'^{*}) \\ &= \Gamma\mathbf{P}_{\mathcal{ZU}}(\mathbf{P}_{\mathcal{ZU}}(\Theta_{22}^{-*})B'^{*}). \end{aligned} \tag{3.16}$$

Since B' is upper, and we are only interested in the strictly upper part of T', only the strictly upper part of Θ_{22}^{-*} will play a role, or equivalently, the strictly lower part of Θ_{22}^{-1}. Moreover, again because B' is upper, multiplication of Θ_{22}^{-*} by B'^{*} does not increase the rank of the Hankel matrices of $\mathbf{P}_{\mathcal{ZU}}(\Theta_{22}^{-*})$ because the product involves only linear combinations of the columns of each separate Hankel matrix of $\mathbf{P}_{\mathcal{ZU}}(\Theta_{22}^{-*})$. Hence the state dimension of T_a is (at most) equal to the state dimension of the strictly upper part of Θ_{22}^{-*}.

To determine the latter dimension, consider figure 8. We position ourselves at point k and split inputs a_1, b_2 and outputs a_2, b_1 of Θ into a strict past and a future segment, with respect to point k. This is written, *e.g.*, as $b_1 = [b_{1p,k}\ \ b_{1f,k}]$, where $b_{1p,k}$ contains the first $k-1$ entries of the sequence b_1. $\Theta_{22}^{-1} = \Sigma_{22}$ is the transfer from port b_2 to port b_1 with the boundary condition $a_1 \equiv 0$, and the strictly lower part of Θ_{22}^{-1} is determined by the collection of transfers $b_{2f,k} \rightarrow b_{1p,k}$ with $a_1 = 0$ and $b_{2p,k} = 0$, for all k in turn. Note that each of these maps defines a local Hankel operator (more precisely, a conjugate Hankel operator, as it describes the effect of an input in the future to the past part of the corresponding output). In addition, a response $b_{1p,k}$ to an input for which $a_{1p,k} = 0$ and $b_{2p,k} = 0$ satisfies an energy relation which is inherited from the unitarity of $\mathbf{\Sigma}$:

$$x_{-,k}x_{-,k}^{*} = b_{1p,k}b_{1p,k}^{*} + a_{2p,k}a_{2p,k}^{*} + x_{+,k}x_{+,k}^{*}. \tag{3.17}$$

Hence the map $x_{-,k} \mapsto [b_{1p,k},\ a_{2p,k},\ x_{+,k}]$ is well-defined (univocal) since if there would exist another image $[b'_{1p,k},\ a'_{2p,k},\ x'_{+,k}]$ for $x_{-,k}$, the quadratic norm of the difference would yield, with (3.17),

$$\|b_{1p,k} - b'_{1p,k}\|^2 + \|a_{2p,k} - a'_{2p,k}\|^2 + \|x_{+,k} - x'_{+,k}\|^2 = 0,$$

which leads to $b_{1p,k} = b'_{1p,k}$, etc. Consequently, the map $x_{-,k} \mapsto b_{1p,k}$ is univocal as well. The Hankel map $H'_k : b_{2f,k} \rightarrow b_{1p,k}$ can be factored into a controllability times an observability map, *i.e.*, (*i*) the transfer of $b_{2f,k}$ to $x_{-,k}$, followed by (*ii*) the transfer of $x_{-,k}$ to $b_{1p,k}$. Hence the state dimension of the strictly upper part of Θ_{22}^{-*} is equal to (at most) the dimension of $x_{-,k}$. Theorems 3.2 and 3.3 claimed that this dimension is in turn equal to the number of negative entries in the signature $J_{\mathcal{B}}$ at point k, or the number of singular values of the Hankel matrix of $\Gamma^{-1}T$ at point k that are larger than 1. Combining this with the previous result, it follows that the state dimension of T_a is (at most) equal to this number. ∎

The following corollary follows from equation (3.17) and is needed in the next section.

Corollary 3.5. *Under the conditions of theorem 3.2, and if $a_{1p,k} = 0$ and $b_{2p,k} = 0$, the map $S_k : x_{-,k} \mapsto x_{+,k}$ is well-defined and a contraction.*

At this point, we have proven the basic form of the Hankel-norm model reduction theorem for time-varying systems (theorem 1.2). With more effort, it is possible to prove that, in proposition 3.4, equality

holds throughout, implying that the approximant T_a has precisely the number of states as specified by the number of Hankel singular values that are larger than 1 [26, 27]. It is also possible to derive an expression (a chain fraction description in terms of Θ) which describes all possible Hankel norm approximants of minimal complexity, given the error tolerance parameter Γ [26, 27].

3.5. Computational model for T_a

A computational model of T_a can be computed directly from the models of T and Θ, via models of B' and Θ_{22}^{-1}. A model for B' has already been obtained in equation (3.14). The model for the strictly upper part of Θ_{22}^{-*} is however more difficult to obtain, and follows from the scattering representation $\mathbf{\Sigma}$ associated to $\mathbf{\Theta}$.

Lemma 3.6. *In the context and under the conditions of theorem 3.2, let* $\mathbf{\Sigma} = \{F, G, H, K\}$ *be the model representation of the unitary scattering matrix associated with* $\mathbf{\Theta} = \{\alpha, \beta, \gamma, \delta\}$*, relating the signal sequences* $[x_+ \quad x_- \quad a_1 \quad b_1]$ *and* $[x_+^{(-1)} \quad x_-^{(-1)} \quad a_2 \quad b_2]$ *as in (2.14). Partition* $\mathbf{\Sigma}$ *and* $\mathbf{\Theta}$ *as in equations (2.16) and (2.15), and let*

$$\begin{array}{lll} S = \mathrm{diag}[S_k] \in \mathcal{D}: & x_{+,k} = x_{-,k} S_k & (a_{1p,k} = 0,\ b_{2p,k} = 0) \\ R = \mathrm{diag}[R_k] \in \mathcal{D}: & x_{-,k} = x_{+,k} R_k & (a_{1f,k} = 0,\ b_{2f,k} = 0) \end{array}$$

Then S and R are well defined, contractive and determined by the recursions

$$\begin{array}{rcl} S^{(-1)} & = & F_{21} + F_{22}(I - SF_{12})^{-1} S F_{11} \\ R & = & F_{12} + F_{11}(I - R^{(-1)} F_{21})^{-1} R^{(-1)} F_{22} \end{array} \tag{3.18}$$

A computational model $\{A_a, B_a, C_r\}$ *of the strictly upper part of* Θ_{22}^{-*}*, i.e.,* $\mathbf{P}_{ZU}(\Theta_{22}^{-*}) = B_a Z (I - A_a Z)^{-1} C_r$*, is given in terms of S, R by*

$$\begin{array}{rcl} A_a^* & = & F_{22}(I - SF_{12})^{-1} \\ B_a^* & = & H_{22} + F_{22}(I - SF_{12})^{-1} S H_{12} \\ C_r^* & = & \left[G_{22} + G_{21}(I - R^{(-1)} F_{21})^{-1} R^{(-1)} F_{22}\right] (I - SR)^{-1} \end{array} \tag{3.19}$$

PROOF The existence and contractivity of S has been derived in corollary 3.5, the comparable result on R is proven in the same way. For clarity, we will not suppress the index k in this proof, so that we are in the context of figure 8. Writing out the relevant part of the relations (2.16), with $a_1 = 0$, we have

$$\left\{ \begin{array}{rcl} x_{+,k+1} & = & x_{+,k} F_{11,k} + x_{-,k+1} F_{21,k} + b_{2,k} G_{21,k} \\ x_{-,k} & = & x_{+,k} F_{12,k} + x_{-,k+1} F_{22,k} + b_{2,k} G_{22,k} \\ b_{1,k} & = & x_{+,k} H_{12,k} + x_{-,k+1} H_{22,k} \end{array} \right. \tag{3.20}$$

With the additional constraint $b_{2p,k+1} = 0$, S_{k+1} satisfies

$$\left\{ \begin{array}{rcl} x_{+,k+1} = x_{-,k+1} S_{k+1} & = & x_{-,k} S_k F_{11,k} + x_{-,k+1} F_{21,k} \\ x_{-,k} & = & x_{-,k} S_k F_{12,k} + x_{-,k+1} F_{22,k} \end{array} \right.$$

Next, $F_{12,k}$ is strictly contractive, because $\mathbf{\Sigma}_{12,k} := \left[\begin{smallmatrix} F_{12,k} & H_{12,k} \\ G_{12,k} & K_{12,k} \end{smallmatrix}\right]$ satisfies

$$I - \mathbf{\Sigma}_{12,k}^* \mathbf{\Sigma}_{12,k} = \mathbf{\Sigma}_{22,k}^* \mathbf{\Sigma}_{22,k} = \left[\begin{array}{cc} \alpha_{22,k} & \gamma_{22,k} \\ \beta_{22,k} & \delta_{22,k} \end{array} \right]^{-*} \left[\begin{array}{cc} \alpha_{22,k} & \gamma_{22,k} \\ \beta_{22,k} & \delta_{22,k} \end{array} \right]^{-1}$$

which is strictly positive definite by the J-unitarity of $\mathbf{\Theta}_k$, so that $\mathbf{\Sigma}_{12,k}$ itself is strictly contractive. $F_{12,k}$, as an entry of it, inherits the property, and hence we can solve for $x_{-,k}$:

$$\left\{ \begin{array}{rcl} x_{-,k} & = & x_{-,k+1} F_{22,k} (I - S_k F_{12,k})^{-1} \\ x_{-,k+1} S_{k+1} & = & x_{-,k+1} \left\{F_{22,k}(I - S_k F_{12,k})^{-1} S_k F_{11,k} + F_{21,k}\right\} \end{array} \right. \tag{3.21}$$

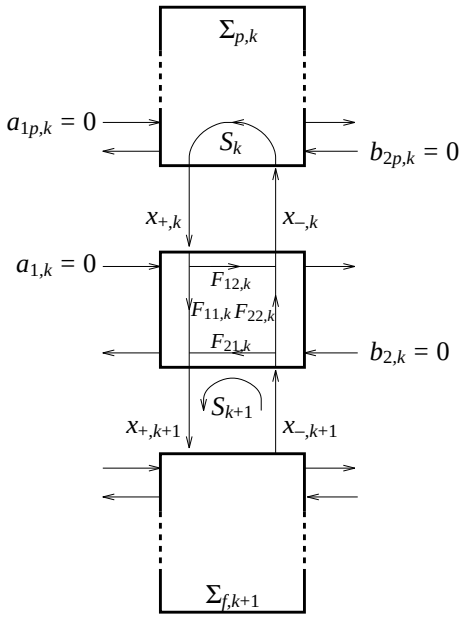

Figure 9. Recursion for *S*.

Consequently, *S* satisfies the indicated recursive relations (see also figure 9). The recursion for *R* is determined likewise.

Let $\{A_a, B_a, C_r\}$ be a state realization for $\mathbf{P}_{ZU}(\Theta_{22}^{-*})$, *i.e.*, $\mathbf{P}_{Z^*\mathcal{L}}(\Theta_{22}^{-1}) = C_r^*(I - Z^*A_a^*)^{-1}Z^*B_a$, which corresponds to the anti-causal computational model

$$\left\{ \begin{array}{lcl} x_{-,k} &=& x_{-,k+1}A_{a,k}^* \;+\; b_{2,k}C_{r,k}^* \\ b_{1,k} &=& x_{-,k+1}B_{a,k}^* \end{array} \right. .$$

The unknowns A_a, B_a and C_r can be expressed in terms of *F*, *G*, *H* by substitution in equations (3.20), using *S* and *R* as intermediate quantities. Doing so with $b_2 = 0$, the first equation in (3.21) yields the expression for A_a in (3.19) and B_a can be determined in terms of *S* from the last equation in (3.20).

Finally, $C_{r,k}^*$ is obtained pointwise as the transfer $b_{2,k} \to x_{-,k}$ for $a_1 = 0$ and $b_{2,i} = 0$ ($i \neq k$). Using (3.6), (3.18) and (3.20), this yields C_r as in (3.19). ∎

We are now in a position to determine a computational model for T_a.

Theorem 3.7. *Let T, Γ, U and Θ be as in theorem 3.2, so that* $[U^* \;\; -T^*\Gamma^{-1}]\Theta = [A' \;\; -B']$. *Let* $\{A, B, C, 0\}$ *be an output normal strictly stable state realization for T, let M be defined by the recursion in (3.8), and let* $\{A, B_U, C, D_U\}$ *be a realization for U. Suppose that* $\mathbf{\Theta}$ *is partitioned as in (3.7), and* $\mathbf{\Sigma}$ *corresponding to* $\mathbf{\Theta}$ *as in (2.16). Define S, R,* $C_r \in \mathcal{D}$ *by the relations*

$$\begin{array}{lcl} S^{(-1)} &=& F_{21} + F_{22}(I - SF_{12})^{-1}SF_{11} \\ R &=& F_{12} + F_{11}(I - R^{(-1)}F_{21})^{-1}R^{(-1)}F_{22} \\ C_r^* &=& \left[G_{22} + G_{21}(I - R^{(-1)}F_{21})^{-1}R^{(-1)}F_{22}\right](I - SR)^{-1} . \end{array}$$

Then T_a *has a computational model* $\{A_a, \Gamma B_a, C_a, 0\}$ *given by*

$$\begin{array}{lcl} A_a^* &=& F_{22}(I - SF_{12})^{-1} \\ B_a^* &=& H_{22} + F_{22}(I - SF_{12})^{-1}SH_{12} \\ C_a &=& C_r\left[-D_{12}^*D_U + C_2^*(I - M)C\right] + A_aY^{(-1)}A^*MC \end{array}$$

where $Y \in \mathcal{D}$ *is given by the solution of the recursion* $Y = A_aY^{(-1)}A^* + C_rC_2^*$.

PROOF The computational model for T_a will be obtained, using definition (3.16), by multiplying a model for B' by the model $\{A_a, B_a, C_r\}$ for $\mathbf{P}_{\mathcal{U}}(\Theta_{22}^{-*})$ as obtained in lemma 3.6. A model for B' has already been obtained in equation (3.14). With $D' := -D_U^* D_{12} + C^*(I - M)C_2$, T_a is given by the strictly upper part of

$$\begin{aligned}\Gamma \mathbf{P}_{\mathcal{U}}(\Theta_{22}^{-*}) B'^* &= \Gamma \left\{ B_a Z(I - A_a Z)^{-1} C_r \right\} \cdot \left\{ C_2^*(I - Z^* A^*)^{-1} Z^* A^* MC + D'^* \right\} \\ &= \Gamma B_a Z(I - A_a Z)^{-1} C_r D'^* + \Gamma B_a \left\{ Z(I - A_a Z)^{-1} C_r C_2^* (I - Z^* A^*)^{-1} \right\} Z^* A^* MC .\end{aligned}$$

The computation of the strictly upper part of this expression requires a partial fraction decomposition of the expression $Z(I - A_a Z)^{-1} C_r C_2^* (I - Z^* A^*)^{-1}$. We seek diagonal matrices X and Y such that

$$Z(I - A_a Z)^{-1} C_r C_2^* (I - Z^* A^*)^{-1} = Z(I - A_a Z)^{-1} Y + X(I - Z^* A^*)^{-1} .$$

Pre- and postmultiplying with $(Z^* - A_a)$ and $(I - Z^* A^*)$, respectively, we obtain the equations

$$\left\{ \begin{array}{lcl} C_r C_2^* &=& Y - A_a X \\ 0 &=& -Y^{(-1)} A^* + X \end{array} \right. \quad \Leftrightarrow \quad \left\{ \begin{array}{lcl} X &=& Y^{(-1)} A \\ Y &=& A_a Y^{(-1)} A^* + C_r C_2^* \end{array} \right.$$

The recursive equation for Y that we have thus obtained always has a solution, since for $n \times n$ matrices T with a zero number of states at point $n + 1$, we can start with $Y_{n+1} = [\,\cdot\,]$ and work backwards to Y_1. Via $Z(I - A_a Z)^{-1} Y Z^* = Y^{(-1)} + Z(I - A_a Z)^{-1} A_a Y^{(-1)}$ we obtain

$$T_a = \Gamma B_a Z(I - A_a Z)^{-1} \left\{ C_r D'^* + A_a Y^{(-1)} A^* MC \right\} ,$$

that is, $C_a = C_r \left\{ -D_{12}^* D_U + C_2^*(I - M)C \right\} + A_a Y^{(-1)} A^* MC$. ∎

A check on the dimensions of A_a reveals that the state realization for T_a has indeed a state space dimension given by $N = \#_-(J_{\mathcal{B}})$: at each point it is equal to the number of local Hankel singular values of T which are larger than 1. The realization is given in terms of four recursions: two for M and S that run forward in time, the other two for R and Y that run backward in time and depend on S. Algorithm 3 shows the computations derived from theorem 3.7. It computes a model $\{A_a, B_a, C_a, 0\}$ for T_a in terms of a model $\{A, B, C, 0\}$ for T.

4. COMPUTATION OF Θ BY A GENERALIZED SCHUR ALGORITHM

4.1. Introduction

The global state space procedure of section 3 yields, for a given $T \in \mathcal{U}$, an inner factor U and an interpolating Θ. It can be specialized to the case where T is a general upper triangular matrix without an a priori known state structure. The resulting procedure to obtain Θ leads to a generalized Schur recursion, which we derive for an example T.

Consider a 4×4 strictly upper triangular matrix T,

$$T = \begin{bmatrix} \boxed{0} & t_{12} & t_{13} & t_{14} \\ & \underline{0} & t_{23} & t_{24} \\ & & \underline{0} & t_{34} \\ & & & \underline{0} \end{bmatrix} ,$$

where the $(1, 1)$-entry is indicated by a square and the main diagonal by underscores. For convenience of notation, and without loss of generality, we may take $\Gamma = I$, and thus seek for T_a (a 4×4 matrix) such that $\| T - T_a \| \leq 1$. A trivial (but non-minimal) state realization for T that has $AA^* + CC^* = I$ is obtained by selecting $\{[0, 0, 1], [0, 1, 0], [1, 0, 0]\}$ as a basis for the row space of the second Hankel

In:	$\mathbf{T} = \{A, B, C, D\}$	(model in output normal form for a strictly upper matrix T)
	Γ	(approximation parameters)
Out:	$\mathbf{T}_a = \{A_a, \Gamma B_a, C_a, 0\}$	(model for Hankel norm approximant T_a)

do algorithm 2: gives M_k, $\mathbf{\Theta}_k$, $J_{\mathcal{B}_k}$, $C_{2,k}$, $D_{12,k}$, $D_{U,k}$ ($k = 1, \cdots, n$)

$S_1 = [\,\cdot\,]$

for $k = 1, \cdots, n$

$\quad$ Compute $\mathbf{\Sigma}_k$ from $\mathbf{\Theta}_k$ using (2.17)

$\quad$ $S_{k+1} = F_{21,k} + F_{22,k}(I - S_k F_{12,k})^{-1} S_k F_{11,k}$

end

$R_{n+1} = [\,\cdot\,]$

$Y_{n+1} = [\,\cdot\,]$

for $k = n, \cdots, 1$

$$
\begin{aligned}
R_k &= F_{12,k} + F_{11,k}(I - R_{k+1}F_{21,k})^{-1} R_{k+1} F_{22,k} \\
C^*_{r,k} &= \left\{ G_{22,k} + G_{21,k}(I - R_{k+1}F_{21,k})^{-1} R_{k+1} F_{22,k} \right\} (I - S_k R_k)^{-1} \\
A_{a,k} &= \left\{ F_{22,k}(I - S_k F_{12,k})^{-1} \right\}^* \\
B_{a,k} &= \left\{ H_{22,k} + F_{22,k}(I - S_k F_{12,k})^{-1} S_k H_{12,k} \right\}^* \\
Y_k &= A_{a,k} Y_{k+1} A^*_k + C_{r,k} C^*_{2,k} \\
C_{a,k} &= C_{r,k} \left\{ -D^*_{12,k} D_{U,k} + C^*_{2,k}(I - M_k) C_k \right\} + A_{a,k} Y_{k+1} A^*_k M_k C_k
\end{aligned}
$$

end

Algorithm 3. The approximation algorithm.

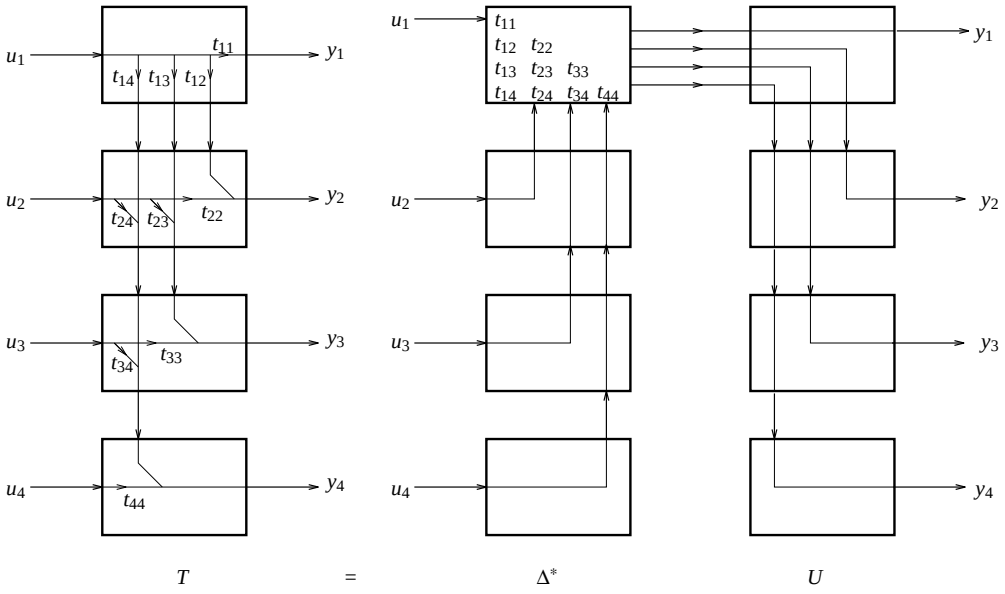

Figure 10. Trivial external factorization of *T*.

matrix $H_2 = [t_{12}, t_{13}, t_{14}]$, and likewise we select trivial bases for H_3 and H_4. Omitting the details, the realizations for *T* and an inner factor *U* that result from this choice turn out to be

$$\mathbf{T}_1 = \left[\begin{array}{cccc} \cdot & \cdot & \cdot & \cdot \\ \hline t_{14} & t_{13} & t_{12} & 0 \end{array}\right] \qquad \mathbf{U}_1 = \left[\begin{array}{cccc} \cdot & \cdot & \cdot & \cdot \\ \hline 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{array}\right]$$

$$\mathbf{T}_2 = \left[\begin{array}{ccc} 1 & & \\ & 1 & \\ & & 1 \\ \hline t_{24} & t_{23} & 0 \end{array}\right] \qquad \mathbf{U}_2 = \left[\begin{array}{ccc} 1 & & \\ & 1 & \\ & & 1 \\ \hline \cdot & \cdot & \cdot \end{array}\right]$$

$$\mathbf{T}_3 = \left[\begin{array}{cc} 1 & \\ & 1 \\ \hline t_{34} & 0 \end{array}\right] \qquad \mathbf{U}_3 = \left[\begin{array}{cc} 1 & \\ & 1 \\ \hline \cdot & \cdot \end{array}\right]$$

$$\mathbf{T}_4 = \left[\begin{array}{cc} \cdot & 1 \\ \hline \cdot & 0 \end{array}\right] \qquad \mathbf{U}_4 = \left[\begin{array}{cc} \cdot & 1 \\ \hline \cdot & \cdot \end{array}\right]$$

('$\cdot$' stands for an entry with zero dimensions). The corresponding matrices *U* and $\Delta = UT^*$ are

$$U = \begin{bmatrix} \boxed{1} & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \qquad \Delta = \begin{bmatrix} \boxed{0} & & & \\ t_{12}^* & 0 & & \\ t_{13}^* & t_{23}^* & 0 & \\ t_{14}^* & t_{24}^* & t_{34}^* & 0 \end{bmatrix}$$

with input space sequences $\mathbb{C}^4 \times \varnothing \times \varnothing \times \varnothing$, and output space sequence $\mathbb{C}^1 \times \mathbb{C}^1 \times \mathbb{C}^1 \times \mathbb{C}^1$. All inputs of *U* and Δ are concentrated at point 1, and hence the causality requirement is always satisfied: $U \in \mathcal{U}$ and $\Delta \in \mathcal{U}$. The structure of Δ and *U* is clarified by figure 10.

The global realization procedure would continue by computing a sequence *M*

$$M_{k+1} = A_k^* M_k A + B_k^* B_k\,, \qquad M_1 = [\,\cdot\,]$$

and use this to derive Θ as in section 3.3. Note that it is not necessary to have a *minimal* realization for T (or U). The extra states will correspond to eigenvalues of M that are zero, and hence are of no influence on the negative signature of $\Lambda = I - M$ (independently of Γ). Hence our non-minimal choice of the realization for T will not influence the complexity of the resulting approximant T_a. For a recursive derivation of an interpolating matrix Θ, however, we proceed as follows. The (trivial) state realizations $\mathbf{T}$ and $\mathbf{U}$ are not needed, but the resulting U is used. The interpolation problem is to determine a J-unitary and causal Θ (whose signature will be determined by the construction) such that

$$[U^* \quad -T^*]\,\Theta \in [\mathcal{U} \quad \mathcal{U}].$$

Assume that $\Theta \in \mathcal{U}(\mathcal{M}_\Theta, \mathcal{N}_\Theta)$. The signature matrix $J_1 = J_{\mathcal{M}_\Theta}$ is known from the outset and is according to the decomposition $[U^* \quad -T^*]$. Although the signature J_2 is not yet known at this point, the number of outputs of Θ (*i.e.,* the space sequence $\mathcal{N}_\Theta$) is already determined by the condition that each $\mathbf{\Theta}_k$ is a square matrix. With the above (trivial) realizations of T and U, it turns out that Θ has a constant number of two outputs at each point in time. The signature of each output (+1 or –1) is determined in the process of constructing Θ, which will be done in two steps: $\Theta = \Theta'\Pi$. Here, Θ' is such that $[U^* \quad -T^*]\,\Theta' \in [\mathcal{U} \quad \mathcal{U}]$, where the dimension sequences of each $\mathcal{U}$ are constant and equal to 1 at each point:

$$\begin{array}{c} + \; + \; + \; + \quad - \quad - \quad - \quad - \\ \left[\begin{array}{cccc|cccc} \boxed{1} & & & & \boxed{-t_{11}^*} & & & \\ & 1 & & & -t_{12}^* & -t_{22}^* & & \\ & & 1 & & -t_{12}^* & -t_{23}^* & -t_{33}^* & \\ & & & 1 & -t_{14}^* & -t_{24}^* & -t_{34}^* & -t_{44}^* \end{array}\right] \end{array} \Theta' = \begin{array}{c} + \; + \quad - \; - \quad + \; + \quad - \; - \\ \left[\begin{array}{cccc|cccc} \boxed{*} & * & * & * & \boxed{*} & * & * & * \\ & \underline{*} & * & * & & \underline{*} & * & * \\ & & * & * & & & * & * \\ & & & * & & & & * \end{array}\right] \end{array}$$

The first upper triangular matrix corresponds to the first output of each section of Θ', and the second to the second output. At this point, the signature of each column at the right hand side can be positive of negative: the output signature matrix of Θ' is J_2', which is an *unsorted* signature matrix such that $\Theta' J_2' \Theta'^* = J_1$ (the signature of the right hand side in the equation above is just an example). See also figure 11. The second step is to sort the columns according to their signature, by introducing a permutation matrix $\Pi \in \mathcal{D}$, such that $J_2 = \Pi^* J_2' \Pi$ is a conventional (sorted) signature matrix. The permutation does not change the fact that $[U^* \quad -T^*]\Theta \in [\mathcal{U} \quad \mathcal{U}]$, but the output dimension sequences of each $\mathcal{U}$ will now be different, and in general not be constant any more. For the above example signature, $[A' \quad -B']$ will have the form

$$\begin{array}{c} + \; + \; + \; + \quad - \quad - \quad - \quad - \\ \left[\begin{array}{cccc|cccc} \boxed{1} & & & & \boxed{-t_{11}^*} & & & \\ & 1 & & & -t_{12}^* & -t_{22}^* & & \\ & & 1 & & -t_{12}^* & -t_{23}^* & -t_{33}^* & \\ & & & 1 & -t_{14}^* & -t_{24}^* & -t_{34}^* & -t_{44}^* \end{array}\right] \end{array} \Theta = \begin{array}{c} + \; + \; + \; + \qquad\qquad - \; - \; - \; - \\ \left[\begin{array}{cccc|cccc} \boxed{*} & * & * & * & \boxed{\cdot} & \cdot & * & * & * & * \\ & \underline{*} & \underline{*} & \cdot\;\cdot & & \underline{\cdot} & * & * & * & * \\ & & & \vdots\;\cdot & & & \underline{*} & \underline{*} & * & * \\ & & & \vdots & & & & & \underline{*} & \underline{*} \end{array}\right] \end{array}$$

$$= [A' \quad -B']$$

where A' has as output space sequence $\mathbb{C}^2 \times \mathbb{C}^2 \times \varnothing \times \varnothing$, and B' has as output space sequence $\varnothing \times \varnothing \times \mathbb{C}^2 \times \mathbb{C}^2$. We will now consider these operations in more detail.

4.2. Computational structure

Θ' can be determined recursively in $n-1$ steps: $\Theta' = \Theta_{(1)}\Theta_{(2)} \cdots \Theta_{(n-1)}$, in the following way. The columns of Θ' act on the columns of U^* and $-T^*$. Its operations on U^* are always causal because all columns of U^* correspond to the first point of the recursion ($k = 1$). However, for Θ to be causal,

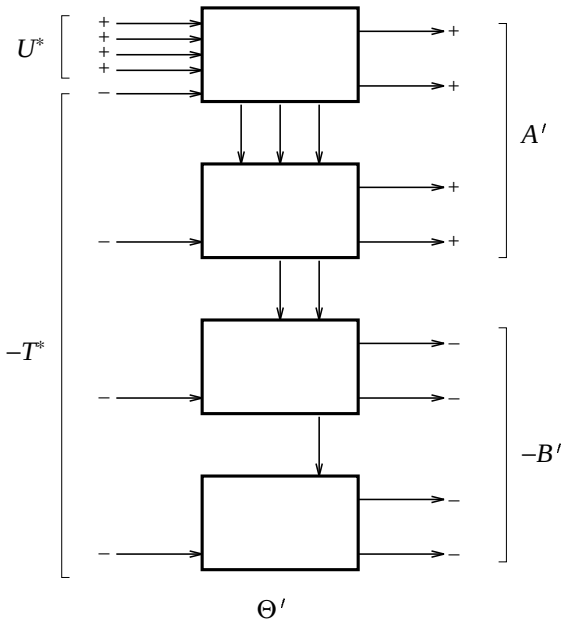

Figure 11. Computational structure of Θ', with example signature at the outputs.

the k-th column of Θ can act only on the first k columns of T^*. Taking this into consideration, we are led to a recursive algorithm of the form $[A_{(k)} \;\; B_{(k)}]\Theta_{(k)} = [A_{(k+1)} \;\; B_{(k+1)}]$, initialized by $A_{(1)} = U^*$, $B_{(1)} = -T^*$, and where $\Theta_{(k)}$ makes the last $(n-k)$ entries of the k-th column of $B_{(k)}$ equal to 0, using columns $n, n-1, \cdots, k+1$ of $A_{(k)}$. (The columns are used in reverse ordering to keep $A_{(k)}$ in the required shape.)

The operations to do each of these steps are elementary unitary (Jacobi) or J-unitary rotations that act on two columns at a time and make a selected entry of the second column equal to zero. The precise nature of the rotations depends on its signature and is in turn dependent on the data — this will be detailed later. We first verify that this recursion leads to a solution of the interpolation problem.

$k = 1$: Using 3 elementary rotations, the entries t^*_{14}, t^*_{13}, t^*_{12} are subsequently zeroed. This results in

$$[A_{(2)} \;\; B_{(2)}] = \left[\begin{array}{cccc|cccc} 1 & * & * & * & \boxed{*} & & & \\ 0 & * & * & * & 0 & -t_{22^*} & & \\ 0 & 0 & * & * & 0 & -t^*_{23} & -t^*_{33} & \\ 0 & 0 & 0 & * & 0 & -t^*_{24} & -t^*_{34} & -t^*_{44} \end{array}\right]$$

$k = 2$:

$$[A_{(3)} \;\; B_{(3)}] = \left[\begin{array}{cccc|cccc} 1 & * & * & * & \boxed{*} & * & & \\ 0 & * & * & * & 0 & * & & \\ 0 & 0 & * & * & 0 & 0 & -t^*_{33} & \\ 0 & 0 & 0 & * & 0 & 0 & -t^*_{34} & -t^*_{44} \end{array}\right]$$

$k = 3$:

$$[A_{(4)} \;\; B_{(4)}] = \left[\begin{array}{cccc|cccc} 1 & * & * & * & \boxed{*} & * & * & 0 \\ 0 & * & * & * & 0 & * & * & 0 \\ 0 & 0 & * & * & 0 & 0 & * & 0 \\ 0 & 0 & 0 & * & 0 & 0 & 0 & -t^*_{44} \end{array}\right]$$

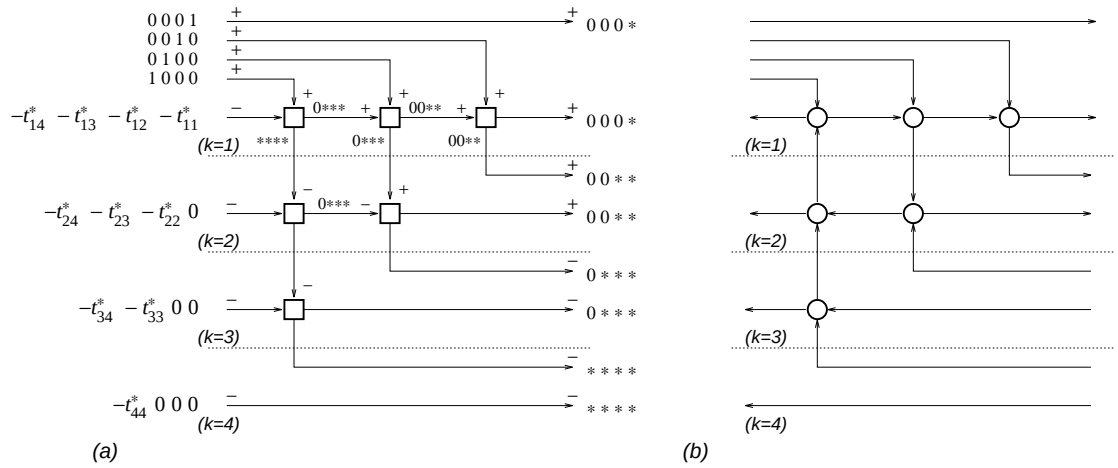

Figure 12. Computational structure of a recursive solution to the interpolating problem. (*a*) Θ', with elementary rotations of mixed type (both circular and hyperbolic); (*b*) the corresponding Σ, with circular elementary rotations. The type of sections in (*a*) and the signal flow in (*b*) depend on the data of the interpolation problem.

The resulting matrices are upper triangular. The signal flow corresponding to this computational scheme is outlined in figure 12(*a*). Note that the computations have introduced an implicit notion of state, formed by the arrows that cross a dotted line between two stages, so that a (non-minimal) realization of Θ can be inferred from the elementary operations.

$[A' \;\; -B']$ will be equal to a column permutation of $[A_{(4)} \;\; B_{(4)}]$, such that A' has all columns with positive signature, whereas B' has all columns with a negative signature. The determination of the signature of $[A_{(4)} \;\; B_{(4)}]$ is discussed in the next subsection.

4.3. Elementary rotations: keeping track of signatures

We will now consider the elementary operations in the above recursions. An elementary rotation θ such that $\theta^* j_1 \theta = j_2$ (j_1 and j_2 are 2×2 signature matrices) is defined by

$$[\, u \quad t \,]\, \theta = [\, * \quad 0 \,],$$

where u, t are scalars, and where '$*$' stands for some resulting scalar. Initially, one would consider θ of a traditional J-unitary form:

$$\theta_1 = \begin{bmatrix} 1 & -s \\ -s^* & 1 \end{bmatrix} \frac{1}{c^*}, \qquad cc^* + ss^* = 1,\ c \neq 0$$

which satisfies

$$\theta_1^* \begin{bmatrix} 1 & \\ & -1 \end{bmatrix} \theta_1 = \begin{bmatrix} 1 & \\ & -1 \end{bmatrix}.$$

However, since $|s| < 1$, a rotation of this form is appropriate only if $|u| > |t|$. In the recursive algorithm, this will be the case only if $H_T^* H_T < I$ which corresponds to a 'definite' interpolation problem and leads to an approximant $T_a = 0$. Our situation is more general. If $|u| < |t|$, we require a rotational section of the form

$$\theta_2 = \begin{bmatrix} -s & 1 \\ 1 & -s^* \end{bmatrix} \frac{1}{c^*},$$

resulting in $[\, u \;\; t \,]\, \theta_2 = [\, * \;\; 0 \,]$. θ_2 has signature pairs determined by

$$\theta_2^* \begin{bmatrix} 1 & \\ & -1 \end{bmatrix} \theta_2 = \begin{bmatrix} -1 & \\ & 1 \end{bmatrix}.$$

This shows that the signature of the 'energy' of the output vector of such a section is reversed: if $[a_1 \quad b_1]\theta_2 = [a_2 \quad b_2]$, then $a_1 a_1^* - b_1 b_1^* = -a_2 a_2^* + b_2 b_2^*$. Because this signature can be reversed at each elementary step, we will have to keep track of it to ensure that the resulting global Θ-matrix is J-unitary with respect to a certain signature. Thus assign to each column in $[U^* \quad -T^*]$ a signature (+1 or −1), which is updated after each elementary operation, in accordance to the type of rotation. Initially, the signature of the columns of U^* is chosen +1, and those of $-T^*$ are chosen −1. Because $\Theta' = \Theta_{(1)}\Theta_{(2)} \cdots \Theta_{(n-1)}$, where $\Theta_{(i)}$ is an embedding of the i-th elementary rotation $\theta_{(i)}$ into one of full size, it is seen that keeping track of the signature at each intermediate step ensures that

$$\Theta^* \begin{bmatrix} I & \\ & -I \end{bmatrix} \Theta = J_2' .$$

Here, J_2' is the unsorted signature matrix given by the signatures of the columns of the final resulting upper triangular matrices. The types of signatures that can occur, and the appropriate elementary rotations to use, are listed below. These form the processors in figure 12(*a*).

$$1.\quad \begin{matrix} + & - \\ u & t \end{matrix} \begin{bmatrix} 1 & -s \\ -s^* & 1 \end{bmatrix} \tfrac{1}{c^*} = \begin{matrix} + & - \\ * & 0 \end{matrix}, \qquad \text{if } |u| > |t|$$

$$2.\quad \begin{matrix} + & - \\ u & t \end{matrix} \begin{bmatrix} -s & 1 \\ 1 & -s^* \end{bmatrix} \tfrac{1}{c^*} = \begin{matrix} - & + \\ * & 0 \end{matrix}, \qquad \text{if } |u| < |t|$$

$$3.\quad \begin{matrix} - & + \\ u & t \end{matrix} \begin{bmatrix} -s & 1 \\ 1 & -s^* \end{bmatrix} \tfrac{1}{c^*} = \begin{matrix} + & - \\ * & 0 \end{matrix}, \qquad \text{if } |u| > |t|$$

$$4.\quad \begin{matrix} - & + \\ u & t \end{matrix} \begin{bmatrix} 1 & -s \\ -s^* & 1 \end{bmatrix} \tfrac{1}{c^*} = \begin{matrix} - & + \\ * & 0 \end{matrix}, \qquad \text{if } |u| < |t|$$

$$5.\quad \begin{matrix} + & + \\ u & t \end{matrix} \begin{bmatrix} c & s \\ -s^* & c^* \end{bmatrix} = \begin{matrix} + & + \\ * & 0 \end{matrix}$$

$$6.\quad \begin{matrix} - & - \\ u & t \end{matrix} \begin{bmatrix} c & s \\ -s^* & c^* \end{bmatrix} = \begin{matrix} - & - \\ * & 0 \end{matrix}$$

It can be shown (but we omit the details for brevity) that, for the hyperbolic rotations, the case $|u| = |t|$ can never occur in the algorithm. This is because, at the k-th step, the algorithm essentially acts on $[(H_U)_{k+1}^* \quad (H_T)_{k+1}^*]\underline{\Theta}$, where $\underline{\Theta}$ is some J-unitary matrix consisting of a subset of the rotations performed in the previous steps. The signature of this intermediate result is nonsingular, because $(H_U)_{k+1}^*(H_U)_{k+1} - (H_T)_{k+1}^*(H_T)_{k+1}$ is initially nonsingular by the imposed conditions on the singular values of H_T, and the signature is invariant under J-unitary transformations. At the same time, the first block matrix of $[(H_U)_{k+1}^* \quad (H_T)_{k+1}^*]\underline{\Theta}$ is upper triangular, whereas the second block matrix has all but the last column equal to zero. At this stage, the algorithm is zeroing this last column using columns of the first block matrix. Because of the form of these block matrices, the occurrence of $|u| = |t|$ in a hyperbolic rotation during this zeroing operation implies that the corresponding signature, and hence the initial signature, contains a zero element, leading to a contradiction. Hence $|u| = |t|$ cannot occur for the hyperbolic rotations.

We can associate, as usual, with each J-unitary rotation a corresponding unitary rotation, which is obtained by rewriting the corresponding equations such that the '+' quantities appear on the left hand side and the '−' quantities on the right hand side. The last two sections are already circular rotation matrices. By replacing each of the sections of Θ by the corresponding unitary section, a unitary Σ matrix that corresponds to Θ is obtained. A signal flow scheme of a possible Σ in our 4×4 example is depicted in figure 12(*b*). The matching of signatures at each elementary rotation in the algorithm effects in figure

12(*b*) that the signal flow is well-defined: an arrow leaving some section will not bounce into a signal flow arrow that leaves a neighboring section.

Finally, a solution to the interpolation problem $[U^* \quad -T^*]\Theta = [A' \quad -B']$ is obtained by *sorting* the columns of the resulting upper triangular matrices obtained by the above procedure according to their signature, such that all positive signs correspond to A' and all negative signs to B'. The columns of Θ are sorted likewise. The solution that is obtained this way is reminiscent of the state space solution of the previous section, and in fact can be derived from it by factoring Θ into elementary operations as above. Again, the network of Σ is not computable since it contains loops.

When T is a banded matrix, or has a staircase structure, then operations corresponding to entries off the band can be omitted. The recursion and the resulting computational network is a further generalization (to include indefinite interpolation) of the generalized Schur algorithm introduced in [21]. However, the formalism by which the matrices are set up to initiate the algorithm is new.

4.4. Computation of the approximant

With Θ and B' available, there are various ways to obtain the Hankel norm approximant T_a. The basic relations are given in terms of T' (the upper triangular part of which is equal to T_a) and the operator Σ associated to Θ:

$$\begin{aligned} T'^* &= T^* + U^*\Sigma_{12} \\ T'^* &= B'\Theta_{22}^{-1}, \qquad \Theta_{22}^{-1} = \Sigma_{22}\,. \end{aligned}$$

Ideally, one would want to use the computational network of Σ to derive either $U^*\Sigma_{12}$ or $B'\Theta_{22}^{-1}$. However, the network that has been constructed in the previous step of the algorithm is not *computable*: it contains delay-free loops, and hence it cannot be used directly. A straightforward alternative is to extract Θ_{22} from the network of Θ (by applying an input of the form $[0 \quad I]$), and subsequently use any technique to invert this matrix and apply it to B'. A second alternative is to compute a (non-causal) state realization for Σ from its network. This is a local operation: it can be done independently for each stage. From this realization, one can derive a realization for the upper triangular part of Θ_{22}^{-*}, by using the recursions given in section 3.5.

The first solution can be made more or less ‘in style’ with the way Θ has been constructed, to the level that only elementary, unitary operations are used. However, the overall solution is a bit crude: after extracting the matrix Θ_{22}, the computational network of Θ is discarded, although it reveals the structure of Θ_{22} and Θ_{22}^{-1}, and the algorithm continues with a matrix inversion technique that is not very specific to its current application. The state space technique, on the other hand, uses half of the computational network structure of Θ (the ‘vertical’ segmentation into stages), but does not use the structure within a stage. The algorithm operates on (state space) matrices, rather than at the elementary level, and is in this respect ‘out of style’ with the recursive computation of Θ. It is as yet unclear whether an algorithm can be devised that acts directly on the computational network of Θ with elementary operations.

5. Envoy

The theory presented in this paper gives a closed form solution to the generic problem of approximating a matrix which represents a linear transformation by a matrix of lower computational complexity. The measure of complexity that is used here is ‘state dimension of the computation’. The theory is based on a combination and generalization of three classical paradigms: (1) system theory and realization theory in the vein of Kronecker and Ho-Kalman, (2) interpolation theory in the sense of Schur-Takagi and Adamjan-Arov-Krein, (3) scattering theory as it was introduced in the network theory context by Youla and Belevitch. It is a remarkable fact that such diverse theories come together to produce a complete body of answers.

On the other hand, it is conceivable that alternative approximation schemes are possible. The generalized AAK scheme is based on interpolation of the error in selected "points" (here to be interpreted as diagonals of a matrix)— see equation (3.5). The scheme controls the error via interpolation. It is possible to construct a direct interpolation method, see *e.g.*, [21]. Such a theory will also yield strong approximants but will be dependent on the choice of interpolation points, and hence will not produce a global low-complexity minimum as the algorithm proposed here does. However, the method is easier and gives good results in practice. Other, heuristic methods based on setting entries to zero, *e.g.*, in factors of an LU-decomposition, may work well in practice, and are of course even simpler. It is, however, doubtful that they can produce systematic results.

The results presented can be extended in several directions. The method works well only on triangular matrices. A full matrix can be decomposed in an upper and a lower part, each of which can be approximated separately. A scheme for doing matrix inversions using such a decomposition has been published [9]. In another direction, one may consider the singular case, *i.e.*, when some of the local singular values of the Hankel operator are equal to one. Preliminary results are available but have not been published yet. There is also a connection with the theory of alpha-stationary systems as developed by Kailath and his coworkers [5, 6, 7, 8], but the question of introducing structure in the approximation scheme, or approximating under structural constraints has not been studied yet to our knowledge.

References

[1] I. Schur, "Uber Potenzreihen, die im Innern des Einheitskreises beschränkt sind, I," *J. Reine Angew. Math.*, vol. 147, pp. 205–232, 1917. Eng. Transl. *Operator Theory: Adv. Appl.*, vol. 18, pp. 31-59, Birkhäuser Verlag, 1986.

[2] N. Levinson, "The Wiener RMS error criterion in filter design and prediction," *J. Math. Phys.*, vol. 25, pp. 261–278, 1947.

[3] I. Gohberg and A. Semencul, "On the inversion of finite Toeplitz matrices and their continuous analogs," *Mat. Issled.*, vol. 2, pp. 201–233, 1972.

[4] J. Chun, T. Kailath, and H. Lev-Ari, "Fast parallel algorithms for *QR* and triangular factorizations," *SIAM J. Sci. Stat. Comp.*, vol. 8, no. 6, pp. 899–913, 1987.

[5] T. Kailath, S.Y. Kung, and M. Morf, "Displacement ranks of matrices and linear equations," *J. Math. Anal. Appl.*, vol. 68, no. 2, pp. 395–407, 1979.

[6] H. Lev-Ari and T. Kailath, "Triangular factorizations of structured Hermitian matrices," in *Operator Theory: Advances and Applications*, vol. 18, pp. 301–324, Birkhäuser Verlag, 1986.

[7] J. Chun, *Fast Array Algorithms for Structured Matrices*. PhD thesis, Stanford Univ., Stanford, CA, 1989.

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

[9] A.J. van der Veen and P.M. Dewilde, "Large matrix inversion using state space techniques," in *Proc. 1993 IEEE Workshop on VLSI Signal Processing*, (Veldhoven, The Netherlands), Oct. 1993.

[10] G. Golub and C.F. Van Loan, *Matrix Computations*. The Johns Hopkins University Press, 1984.

[11] S.Y. Kung, "A new identification and model reduction algorithm via singular value decomposition," in *Twelfth Asilomar Conf. on Circuits, Systems and Comp.*, (Asilomar, CA), pp. 705–714, Nov. 1978.

[12] V.M. Adamjan, D.Z. Arov, and M.G. Krein, "Analytic properties of Schmidt pairs for a Hankel operator and the generalized Schur-Takagi problem," *Math. USSR Sbornik*, vol. 15, no. 1, pp. 31–73, 1971. (transl. of *Iz. Akad. Nauk Armjan. SSR Ser. Mat. 6 (1971)).*

[13] L. Kronecker, "Algebraische Reduction der schaaren bilinearer Formen," *S.B. Akad. Berlin*, pp. 663–776, 1890.

[14] A. Bultheel and P.M. Dewilde, "On the Adamjan-Arov-Krein approximation, identification, and balanced realization," in *Proc. 1980 Eur. Conf. on Circ. Th. and Design*, vol. 2, pp. 186–191, 1980.

[15] K. Glover, "All optimal Hankel norm approximations of linear multi-variable systems and their L^∞-error bounds," *Int. J. Control*, vol. 39, no. 6, pp. 1115–1193, 1984.

[16] S.Y. Kung and D.W. Lin, "Optimal Hankel norm model reductions: Multi-variable systems," *IEEE Trans. Automat. Control*, vol. 26, pp. 832–852, Aug. 1981.

[17] Y.V. Genin and S.Y. Kung, "A two-variable approach to the model reduction problem with Hankel norm criterion," *IEEE Trans. Circuits Syst.*, vol. 28, no. 9, pp. 912–924, 1981.

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

[19] D.J.N. Limebeer and M. Green, "Parametric interpolation, H_∞-control and model reduction," *Int. J. Control*, vol. 52, no. 2, pp. 293–318, 1990.

[20] P. Dewilde and E. Deprettere, "Approximative inversion of positive matrices with applications to modeling," in *NATO ASI Series, Vol. F34 on Modeling, Robustness and Sensitivity Reduction in Control Systems*, Berlin: Springer Verlag, 1987.

[21] P. Dewilde and E. Deprettere, "The generalized Schur algorithm: Approximation and hierarchy," in *Operator Theory: Advances and Applications*, vol. 29, pp. 97–116, Birkhäuser Verlag, 1988.

[22] D. Alpay and P. Dewilde, "Time-varying signal approximation and estimation," in *Signal Processing, Scattering and Operator Theory, and Numerical Methods* (M.A. Kaashoek, J.H. van Schuppen, and A.C.M. Ran, eds.), vol. III of *Proc. Int. Symp. MTNS-89*, pp. 1–22, Birkhäuser Verlag, 1990.

[23] D. Alpay, P. Dewilde, and H. Dym, "Lossless Inverse Scattering and reproducing kernels for upper triangular operators," in *Extension and Interpolation of Linear Operators and Matrix Functions* (I. Gohberg, ed.), vol. 47 of *Operator Theory, Advances and Applications*, pp. 61–135, Birkhäuser Verlag, 1990.

[24] P. Dewilde and H. Dym, "Interpolation for upper triangular operators," in *Time-Variant Systems and Interpolation* (I. Gohberg, ed.), vol. 56 of *Operator Theory: Advances and Applications*, pp. 153–260, Birkhäuser Verlag, 1992.

[25] A.J. van der Veen and P.M. Dewilde, "Time-varying system theory for computational networks," in *Algorithms and Parallel VLSI Architectures, II* (P. Quinton and Y. Robert, eds.), pp. 103–127, Elsevier, 1991.

[26] P.M. Dewilde and A.J. van der Veen, "On the Hankel-norm approximation of upper-triangular operators and matrices," *Integral Equations and Operator Theory*, vol. 17, no. 1, pp. 1–45, 1993.

[27] A.J. van der Veen, *Time-Varying System Theory and Computational Modeling: Realization, Approximation, and Factorization*. PhD thesis, Delft University of Technology, Delft, The Netherlands, June 1993.

List of Figures

List of Tables