Time-Variant and Quasi-separable Systems*
– supplementary reading –
.
QL Factorization in State Space

Klaus Diepold, Patrick Dewilde

Spring 2025

1 Inner-Outer Factorization

1.1 Task Description

We will now discuss the factorizations of a causal operator T into the product of a causal factor L and an inner factor Q. The factor L is itself causal and also acausally left-invertible transfer function (by definition a *left-outer function* and Q is a causal, co-isometric transfer function, i.e., a transfer function with orthonormal rows, called a *right-inner function*.

Such a decomposition is traditionally called an *inner-outer factorization* for reasons that have to do with pole and zero locations in LTI transfer functions (a topic that will not occupy us), or, equivalently, a factorization into a minimal phase factor and a lossless factor—a terminology mostly used in electrical engineering. This type of factorization has important applications such as Kalman filtering, least squares optimal control and efficient system inversion.

1.2 QL Factorization

The QL-factorization of a matrix T is a powerful tool for solving linear systems of equations or linear least squares problems as $u = T^{\dagger}y$. The QL-factorization of a given matrix T amounts to computing

$$T = Q \cdot L, \quad Q'Q = \mathbf{1}_n, \quad L\text{: lower triangular.}$$

Note that L is a lower-triangular matrix corresponding with a causal system (in contrast with conventional QR decomposition, where R is supposed to be an upper triangular matrix. We assume that T is a *tall*

*P. Dewilde, K. Diepold, A.-J. v.d. Veen. Time-Variant ans Quasi-separable Systems, Cambridge University Press, 2024 matrix (i.e. $T \in \mathbb{R}^{m\times n}, m \geq n$) and that it has full column rank, i.e. $\det T'T \neq 0$. Having determined the QL-factorization we can easily determine the inverse or the pseudo-inverse of T as

$$T^{\dagger} = L^{-1}Q', \quad L \text{ is square}, \ \det L \neq 0$$

Using the time-varying state-space methodology we are interested in calculating the QL-factorization of a given matrix T in terms of its state-space realizations directly.

The process for computing the QL factorization of T then goes as follows:

1. Determine a state space realization for T
2. Compute the QL factorization in state-space
3. Invert the factors in state-space
4. Determine (Moore-Penrose Pseudo) inverse

Note that for solving least squares problems it is not necessary to actually compute the product $L^{-1}Q'$, it is more economic to keep the factorized form an apply the matrices Q' and L^{-1} subsequently to the vector y and $Q'y$, respectively.

We start out with a time-varying state-space realization Σ for T, such that we can represent the coefficient matrix as

$$T = D + C(\mathbf{1}_n - ZA)^{-1}ZB, \quad \Sigma = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right].$$

Since T is the product of two matrices Q and L we aim at state-space realizations for them, that is we represent both matrices in terms of state-space models

$$Q = D^Q + C^Q(\mathbf{1}_n - ZA^Q)^{-1}ZB^Q, \quad \Sigma^Q = \left[\begin{array}{c|c} A^Q & B^Q \\ \hline C^Q & D^Q \end{array}\right].$$

$$L = D^L + C^L(\mathbf{1}_n - ZA)^{-1}ZB, \quad \Sigma^L = \left[\begin{array}{c|c} A & B \\ \hline C^L & D^L \end{array}\right].$$

We look for a recursive computational scheme to determine the components state space in a direct way.

Figure 1 shows the computational structure we aim for, that is, this structure represents the QL-factorization of the matrix T. To this end we need to determine the realizations for Q and for L.

1.3 Determine a State Space Realization for T

To this end we could engage the realization procedure based on the Kronecker Theorem, i.e. by factoring the Hankel matrices into the product of observability and controllability and then read of the elements A_k, B_k, C_k and D_k of the state-space realization. However, this computational process may require us to compute the corresponding factorizations and the necessary inversions, which turn out to be computationally expensive. Alternatively, we can determine realizations directly by reading off the matrix entries from T and creating a simple computational state-space structure for T as a starting point.

$$T = \tilde{Q}_N \dots \tilde{Q}_2 \tilde{Q}_1 \tilde{L}_N \dots \tilde{L}_2 \tilde{L}_1$$

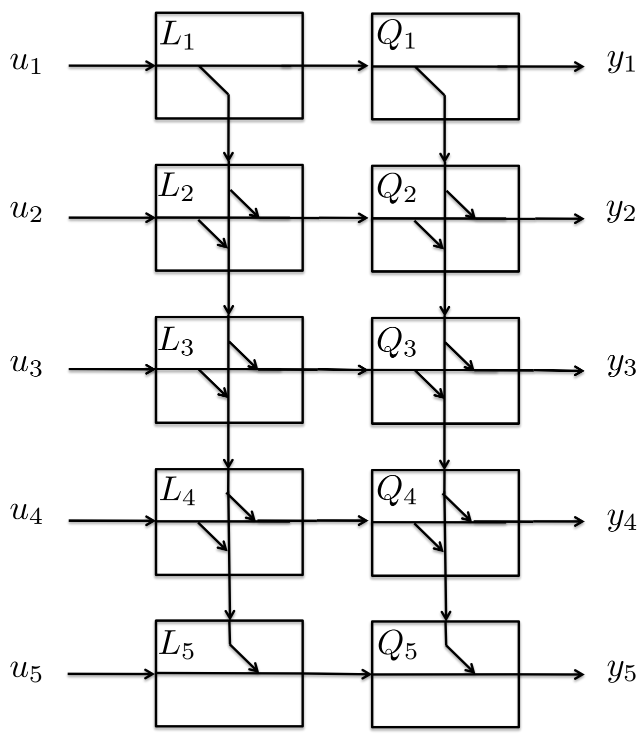

Figure 1: State-Space Realization for the QL-Factorization of T (taken from [5])

1.3.1 Example 1

Consider the simple example of a 4×2 matrix T shown in Figure 2. We can easily check that the computational structure shown in the figure realizes the computation for $Tu = y$. The state-space realization matrices for this structure can also be easily read off as

$$\left[\begin{array}{c|c} A_1 & B_1 \\ \hline C_1 & D_1 \end{array}\right] = \left[\begin{array}{c|c} \cdot & 1 \\ \hline \cdot & t'_1 \end{array}\right], \quad \left[\begin{array}{c|c} A_2 & B_2 \\ \hline C_2 & D_2 \end{array}\right] = \left[\begin{array}{c|c} 1 & \cdot \\ \hline t'_2 & \cdot \end{array}\right], \quad \left[\begin{array}{c|c} A_3 & B_3 \\ \hline C_3 & D_3 \end{array}\right] = \left[\begin{array}{c|c} 1 & \cdot \\ \hline t'_3 & \cdot \end{array}\right]$$

and

$$\left[\begin{array}{c|c} A_4 & B_4 \\ \hline C_4 & D_4 \end{array}\right] = \left[\begin{array}{c|c} \cdot & \cdot \\ \hline t'_4 & \cdot \end{array}\right].$$

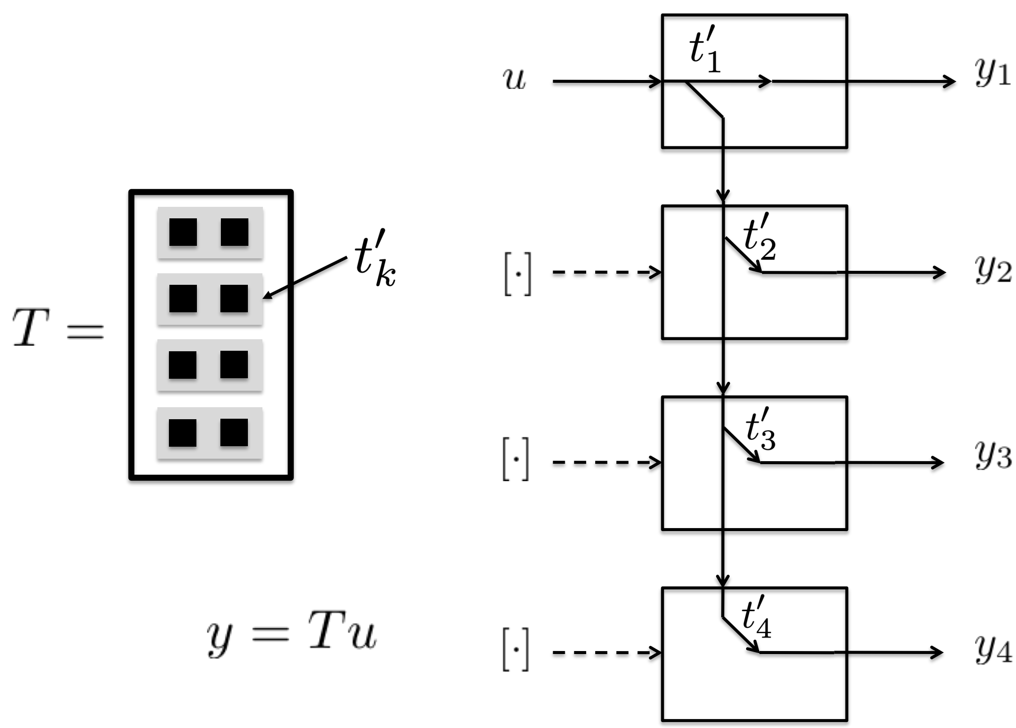

Figure 2: State-Space Realization for the matrix T (taken from [5])

1.3.2 Example 2

Yet another slightly more complicated example is shown in Figure 3. For this example we partition the input vector as $\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$ to give us

$$y = T \begin{bmatrix} u_1 \\ u_2 \end{bmatrix},$$

where we have the corresponding partitioning of the matrix T given as

$$T = \left[\begin{array}{cc} T^{(1)} & T^{(2)} \end{array}\right].$$

For each of these two matrices we assume to have state-space realizations

$$T^{(1)} \Leftrightarrow \left[\begin{array}{c|c} A_k^{(1)} & B_k^{(1)} \\ \hline C_k^{(1)} & D_k^{(1)} \end{array}\right] \quad T^{(2)} \Leftrightarrow \left[\begin{array}{c|c} A_k^{(2)} & B_k^{(2)} \\ \hline C_k^{(2)} & D_k^{(2)} \end{array}\right],$$

which can be combined into one realization matrix as

$$T_k = \left[\begin{array}{cc|cc} A_k^{(1)} & 0 & B_k^{(1)} & 0 \\ 0 & A_k^{(2)} & 0 & B_k^{(2)} \\ \hline C_k^{(1)} & C_k^{(2)} & D_k^{(1)} & D_k^{(2)} \end{array}\right].$$

Using this generic partitioning of the realization matrix along with the realizations determined in the previous section we can easily read of a time-varying realization to consist of the matrices

$$T_1 = \left[\begin{array}{cc|cc} \cdot & 0 & 1 & 0 \\ 0 & \cdot & 0 & \cdot \\ \hline \cdot & \cdot & t_1' & \cdot \end{array}\right], \quad T_2 = \left[\begin{array}{cc|cc} 1 & 0 & \cdot & 0 \\ 0 & \cdot & 0 & \cdot \\ \hline t_2' & \cdot & \cdot & \cdot \end{array}\right],$$

$$T_3 = \left[\begin{array}{cc|cc} 1 & 0 & \cdot & 0 \\ 0 & \cdot & 0 & 1 \\ \hline t_3' & \cdot & \cdot & t_1' \end{array}\right], \quad T_4 = \left[\begin{array}{cc|cc} \cdot & 0 & \cdot & 0 \\ 0 & 1 & 0 & \cdot \\ \hline t_4' & t_2' & \cdot & \cdot \end{array}\right],$$

$$T_5 = \left[\begin{array}{cc|cc} \cdot & 0 & \cdot & 0 \\ 0 & 1 & 0 & \cdot \\ \hline \cdot & t_3' & \cdot & \cdot \end{array}\right], \quad T_6 = \left[\begin{array}{cc|cc} \cdot & 0 & \cdot & 0 \\ 0 & \cdot & 0 & \cdot \\ \hline \cdot & t_4' & \cdot & \cdot \end{array}\right].$$

1.3.3 Factored Representation

We represent the matrix T as the product

$$T = \tilde{T}_N \cdot \tilde{T}_{N-1} \cdot \tilde{T}_{N-2} \cdots \tilde{T}_1,$$

where the factors $\tilde{T}_k$ provide for an embedding of the k-th realization matrix in the form

$$\tilde{T}_k = \left[\begin{array}{c|ccccc} A_k & \dots & \dots & \dots & B_k & \\ \hline \vdots & 1 & & & \vdots & \\ \vdots & & \ddots & & \vdots & \\ \vdots & & & 1 & \vdots & \\ C_k & \dots & \dots & \dots & D_k & \\ & & & & & 1 \end{array}\right].$$

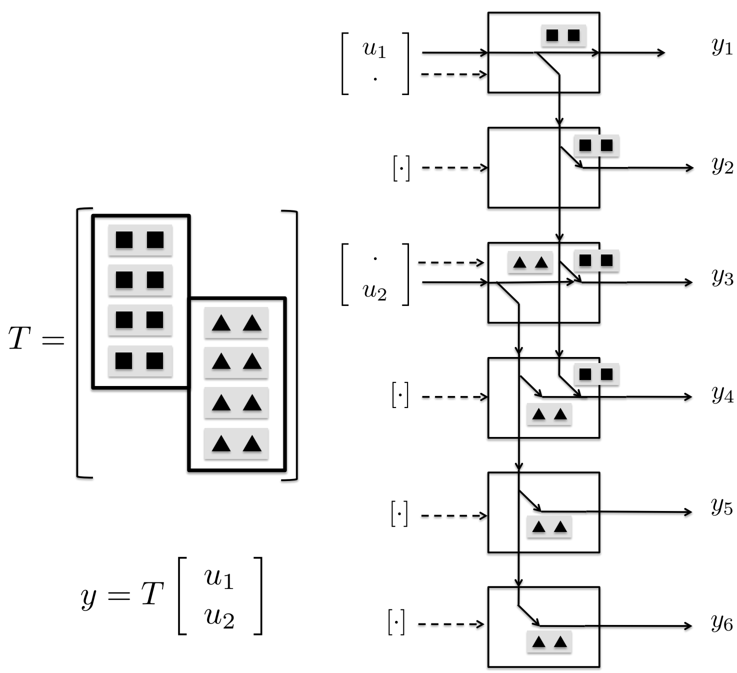

Figure 3: Direct State-Space Realization for the matrix T (taken from [5])

Example We can check this factorized representation for T looking at a small example for $N = 4$. For the lower 4×4 semi-separable matrix we have

$$T = \begin{bmatrix} D_1 & & & \\ C_2B_1 & D_2 & & \\ C_3A_2B_1 & C_3B_2 & D_3 & \\ C_4A_3A_2B_1 & C_4A_3B_2 & C_4B_3 & D_4 \end{bmatrix}.$$

We check the factored form by directly calculating

$$\begin{aligned}
T &= \tilde{T}_4 \cdot \tilde{T}_3 \cdot \tilde{T}_2 \cdot \tilde{T}_1 \\
&= \left[\begin{array}{c|cccc} A_4 & . & . & . & B_4 \\ \hline . & 1 & & & \\ . & & 1 & & \\ . & & & 1 & \\ C_4 & & & & D_4 \end{array}\right] \cdot \left[\begin{array}{c|cccc} A_3 & . & . & B_3 & . \\ \hline . & 1 & & & \\ . & & 1 & & \\ C_3 & & & D_3 & \\ . & & & & 1 \end{array}\right] \cdot \left[\begin{array}{c|cccc} A_2 & . & B_2 & . & . \\ \hline . & 1 & & & \\ C_2 & & D_2 & & \\ . & & & 1 & \\ . & & & & 1 \end{array}\right] \cdot \left[\begin{array}{c|cccc} A_1 & B_1 & . & . & . \\ \hline C_1 & D_1 & & & \\ . & & 1 & & \\ . & & & 1 & \\ . & & & & 1 \end{array}\right] \\
&= \left[\begin{array}{c|cccc} A_4A_3 & . & . & A_4B_3 & B_4 \\ \hline . & 1 & & & \\ . & & 1 & & \\ C_3 & & & D_3 & \\ C_4A_3 & & & C_4B_3 & D_4 \end{array}\right] \cdot \left[\begin{array}{c|cccc} A_2 & . & B_2 & . & . \\ \hline . & 1 & & & \\ C_2 & & D_2 & & \\ . & & & 1 & \\ . & & & & 1 \end{array}\right] \cdot \left[\begin{array}{c|cccc} A_1 & B_1 & . & . & . \\ \hline C_1 & D_1 & & & \\ . & & 1 & & \\ . & & & 1 & \\ . & & & & 1 \end{array}\right] \\
&= \left[\begin{array}{c|cccc} A_4A_3A_2 & . & A_4A_3B_2 & A_4B_3 & B_4 \\ \hline . & 1 & & & \\ C_2 & & D_2 & & \\ C_3A_2 & & C_3B_2 & D_3 & \\ C_4A_3A_2 & & C_4A_3B_2 & C_4B_3 & D_4 \end{array}\right] \cdot \left[\begin{array}{c|cccc} A_1 & B_1 & . & . & . \\ \hline C_1 & D_1 & & & \\ . & & 1 & & \\ . & & & 1 & \\ . & & & & 1 \end{array}\right] \\
&= \left[\begin{array}{c|cccc} A_4A_3A_2A_1 & A_4A_3A_2B_1 & A_4A_3B_2 & A_4B_3 & B_4 \\ \hline C_1 & D_1 & & & \\ C_2A_1 & C_2B_1 & D_2 & & \\ C_3A_2A_1 & C_3A_2B_1 & C_3B_2 & D_3 & \\ C_4A_3A_2A_1 & C_4A_3A_2B_1 & C_4A_3B_2 & C_4B_3 & D_4 \end{array}\right] \\
&= \left[\begin{array}{c|cccc} \cdot & \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & D_1 & & & \\ \cdot & C_2B_1 & D_2 & & \\ \cdot & C_3A_2B_1 & C_3B_2 & D_3 & \\ \cdot & C_4A_3A_2B_1 & C_4A_3B_2 & C_4B_3 & D_4 \end{array}\right],
\end{aligned}$$

where we have used that $A_1 = [\cdot]$, $C_1 = [\cdot]$, $A_4 = [\cdot]$ and $B_4 = [\cdot]$, producing a zero-dimensional first block-column and first block-row.

1.4 Compute the QL Factorization in State Space

1.4.1 Setting up the Recursion

This represents the completion of step 2 in our operational process, where we represent the matrix T as the product

$$T = \tilde{Q}_N \cdots \tilde{Q}_2 \tilde{Q}_1 \tilde{L}_N \cdots \tilde{L}_2 \tilde{L}_1,$$

where we use the notation

$$\tilde{Q}_k = \left[\begin{array}{c|cccc} A_k^Q & \dots & \dots & \dots & B_k^Q \\ \hline \vdots & 1 & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & 1 & \vdots \\ C_k^Q & \dots & \dots & \dots & D_k^Q \\ & & & & & 1 \end{array}\right], \quad \tilde{L}_k = \left[\begin{array}{c|cccc} A_k^L & \dots & \dots & \dots & B_k^L \\ \hline \vdots & 1 & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & 1 & \vdots \\ C_k^L & \dots & \dots & \dots & D_k^L \\ & & & & & 1 \end{array}\right].$$

The state space realizations for the factors of a QL factorization can be computed by a recursive algorithm to determine

$$\left[\begin{array}{c|c} Y_{k+1}A_k & Y_{k+1}B_k \\ \hline C_k & D_k \end{array}\right] = \left[\begin{array}{c|c} A_k^Q & B_k^Q \\ \hline C_k^Q & D_k^Q \end{array}\right] \left[\begin{array}{c|c} Y_k & 0 \\ \hline C_k^L & D_k^L \end{array}\right]. \tag{1}$$

The matrices Y_k and A_k have the same number of columns, D_k^L and D_k have also the same number of columns, while D_k^L has full row rank. The matrices Y_k or D_k^L may be zero-dimensional ($[\cdot]$) and hence the matrix entry 0 may also vanish. The recursion starts out with $Y_{k+1} = [\cdot]$ and continues for $k = N, N-1, \dots, 1$.

Recall from the Lossless Bounded Real Lemma, that the inner function Q has a unitary (orthogonal) state space realization, which amounts to

$$Q'Q = 1 \quad \Rightarrow \Sigma = \left[\begin{array}{c|c} A_k^Q & B_k^Q \\ \hline C_k^Q & D_k^Q \end{array}\right] \quad \Rightarrow \Sigma'\Sigma = 1.$$

For practical purposes we rewrite Equation 1 by bringing the factor Q to the left side to arrive at

$$\left[\begin{array}{c|c} A_k^Q & B_k^Q \\ \hline C_k^Q & D_k^Q \end{array}\right]' \cdot \left[\begin{array}{c|c} Y_{k+1}A_k & Y_{k+1}B_k \\ \hline C_k & D_k \end{array}\right] = \left[\begin{array}{c|c} Y_k & 0 \\ \hline C_k^L & D_k^L \end{array}\right].$$

As a result of performing this recursive computation scheme for all values of k we arrive at a realization matrix for the lower matrix L

$$\Sigma_k^L = \left[\begin{array}{c|c} A_k & B_k \\ \hline C_k^L & D_k^L \end{array}\right].$$

Note that this amounts to applying an appropriately chosen sequence of Givens rotations from the left with the purpose to eliminate the 12-block and to generate the lower triangular shape on the right-hand side of the equation. This is very similar to the conventional algorithm for computing the QR factorization (as shown in [2]) except that we create a lower triangular matrix instead of an upper triangular. This requires a slight change in the elimination sequence.

It is interesting to observe that computing the QR decomposition of a matrix T in state space ends up being a QR decomposition, using the same computational tools, inheriting all the positive the numerical properties, while being more efficient.

1.4.2 Details of the Recursive Computation

Working out the first steps of the recursive algorithm produces the intermediate matrices

$$
\tilde{Q}_N' T = \left[\begin{array}{c|c} A_N^Q & B_N^Q \\ \hline & 1 \\ C_N^Q & D_N^Q \end{array}\right]' \cdot \left[\begin{array}{c|c} Y_{N+1}A_N & Y_{N+1}B_N \\ \hline & 1 \\ C_N & D_N \end{array}\right] \cdot \tilde{T}_{N-1} \cdot \cdots \cdot \tilde{T}_1 =
$$

$$
= \left[\begin{array}{c|cccc} Y_N & & & & 0 \\ \hline & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ C_N^L & & & & D_N^L \end{array}\right] \cdot \left[\begin{array}{c|cccc} A_{N-1} & & & B_{N-1} & \\ \hline & 1 & & & \\ & & \ddots & & \\ C_{N-1} & & & D_{N-1} & \\ & & & & 1 \end{array}\right] \cdot \tilde{T}_{N-2} \cdot \cdots \cdot \tilde{T}_1
$$

$$
= \left[\begin{array}{c|ccc} Y_N A_{N-1} & & Y_N B_{N-1} & \\ \hline & 1 & & \\ & & \ddots & \\ C_{N-1} & & D_{N-1}^L & \\ C_N^L A_{N-1} & & C_N^L B_{N-1} & D_N^L \end{array}\right] \cdot \tilde{T}_{N-2} \cdot \cdots \cdot \tilde{T}_1,
$$

where we made us of Equation 1. Pre-multiplication with the next factor $\tilde{Q}_{N-1}'$ produces the intermediate result

$$
\tilde{Q}_{N-1}' \tilde{Q}_N' T = \left[\begin{array}{c|ccc} Y_{N-1} & & 0 & \\ \hline & 1 & & \\ & & \ddots & \\ C_{N-1}^L & & D_{N-1}^L & \\ C_N^L A_{N-1} & & C_N^L B_{N-1} & D_N^L \end{array}\right] \cdot \left[\begin{array}{c|ccc} A_{N-2} & B_{N-2} & & \\ \hline & 1 & & \\ C_{N-2} & D_{N-1} & & \\ & & 1 & \\ & & & 1 \end{array}\right] \cdot \tilde{T}_{N-3} \cdot \cdots \cdot \tilde{T}_1
$$

$$
= \left[\begin{array}{c|cccc} Y_{N-1} A_{N-2} & & Y_{N-1} B_{N-2} & & \\ \hline & 1 & & & \\ C_{N-2} & & D_{N-2}^L & & \\ C_{N-1}^L & & C_{N-1}^L B_{N-2} & D_{N-1}^L & \\ C_N^L A_{N-1} & & C_N^L A_{N-1} B_{N-2} & C_N^L B_{N-1} & D_N^L \end{array}\right] \cdot \tilde{T}_{N-3} \cdot \cdots \cdot \tilde{T}_1
$$

Continuing the recursive computation will eventually produce

$$
\tilde{Q}_1' \ldots \tilde{Q}_N' T = \left[\begin{array}{c|cccc} Y_1 & & & & \\ \hline C_1^T & D_1^L & & & \\ C_2^L A_1 & C_2^L B_1 & D_2^L & & \\ \vdots & \vdots & \vdots & \ddots & \\ C_N^L A_{N-1} \ldots A_1 & C_N^L A_{N-1} \ldots A_2 B_1 & \ldots & \ldots & D_N^L \end{array}\right],
$$

which we identify as the lower factor L, since $Y_1 = [\cdot]$ and $A_1 = [\cdot]$.

Each individual factor $\tilde{Q}_i$ can be thought of as either a Jacobi or a Householder transformation used for the subsequent elimination of the appropriate matrix entries.

1.5 Invert the Factors in State Space

In the next step we take the individual stages of the structure and determine the inverse realization by local inversion. Once we have computed the state-space realizations for Q and L we can easily determine the state-space realizations of the inverse systems via

$$\Gamma_k^S = \left[\begin{array}{c|c} A_k - B_k(D_k^L)^{-1}C_k^L & B_k(D_k^L)^{-1} \\ \hline -(D_k^L)^{-1}C_k^L & (D_k^L)^{-1} \end{array}\right] = \left[\begin{array}{c|c} A_k^S & B_k^S \\ \hline C_k^S & D_k^S \end{array}\right],$$

where we have $S = L^{-1}$ and

$$\Gamma_k^Q = (\Sigma_k^Q)' = \left[\begin{array}{c|c} (A_k^Q)^T & (C_k^Q)^T \\ \hline (B_k^Q)^T & (D_k^Q)^T \end{array}\right],$$

where the realizations Γ_k^Q represent an anti-causal system. Anti-causal realization corresponds to a backward recursion

$$\begin{aligned} x_k &= (A_k^Q)' x_{k+1} + (C_k^Q)' u_k \\ y_k &= (B_k^Q)' x_{k+1} + (D_k^Q)' u_k \\ & \quad k = N, N-1, \ldots 1 \end{aligned}$$

1.6 Determine (Moore-Penrose Pseudo) Inverse

Concatenating the inverse realizations produces the structure shown in Figure 4, which is a state space realization for the inverse matrix T^{-1} given in terms of the factors Q^T and $\boldsymbol{L}^{-1}$. More accurately, the (Moore-Penrose Pseudo) Inverse is given in terms of the product

$$T^\dagger = \tilde{S}_N \cdots \tilde{S}_2 \tilde{S}_1 \tilde{Q}_1' \tilde{Q}_2' \cdots \tilde{Q}_N'.$$

Note in Figure 4 how the anti-causality of the realization for Q' creates an upward flow of the state signals.

Literatur

[1] G. Strang. *Computational Science and Engineering.* Wellesley-Cambridge Press, 2007.

[2] G. Golub, Ch. van Loan. *Matrix Computations.* John Hopkins, 1992.

[3] P. Dewilde, A.-J. van der Veen. *Time-Varying Systems and Computations.* Kluwer Academic Publishers, 1998.

[4] P. Dewilde, K. Diepold, A.-J. van der Veen. *Time-Variant and Quasi-separable Systems.* Cambridge University Press, 2024.

[5] L. Tong, A.-J. van der Veen, P. Dewilde, Y. Sung. *Blind Decorrelating RAKE Receivers for Long-Code WCDMA.* IEEE Trans. Signal Processing, Vol.51, No.6, pp. 1642-1655, June 2003.

$$T^{\dagger} = \tilde{S}_N \ldots \tilde{S}_2 \tilde{S}_1 \tilde{Q}_1^T \tilde{Q}_2^T \ldots \tilde{Q}_N^T$$

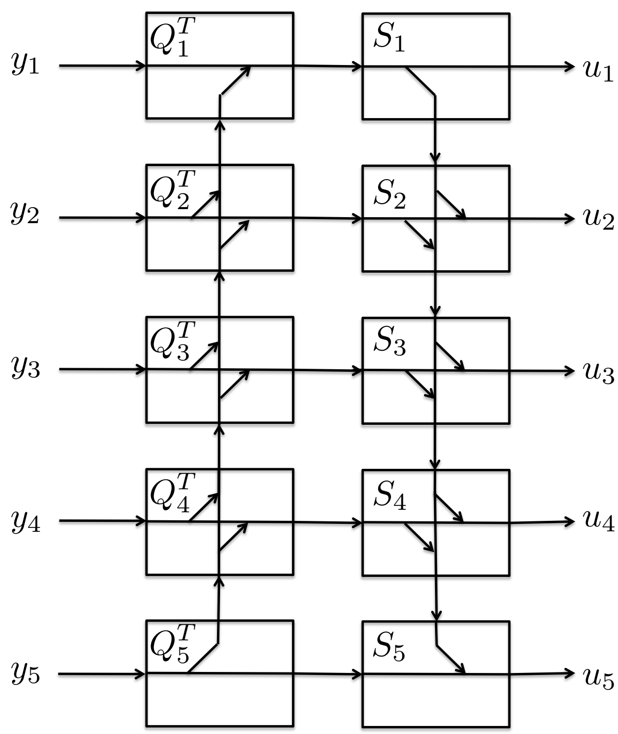

Figure 4: State-Space Realization for the (Moore-Penrose Pseudo) inverse of T (taken from [5])