Connections of Time-Varying Systems and Computational Linear Algebra

Alle-Jan van der Veen and Patrick Dewilde

Delft University of Technology
Department of Electrical Engineering
2628 CD Delft, The Netherlands

Linear algebra problems such as matrix-vector multiplication, inversion and factorizations may be studied from the point of view of time-varying systems and state realizations. This leads to new and efficient algorithms for solving certain large structured matrix problems. In this paper, we treat the matrix inversion problem in more detail.

1. STATE REALIZATION OF A MATRIX

In a number of signal processing applications, such as inverse filtering and spectrum estimation, the basic algorithmic kernel consists of QR-factorizations and matrix inversions of fairly large matrices. Usually, these matrices are not fully random but have some kind of structure, which is inherited from the underlying signal properties. For example, in stationary environments, the covariance matrices formed on the data have a Toeplitz structure (constant along diagonals). Efficient algorithms which exploit this structure are known in this case: the inverse can be computed via Levinson recursions or Gohberg/Semencul recursions [1], and the QR-factorization can be computed via a generalization of the Schur recursion [2]. The resulting algorithms have computational 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.

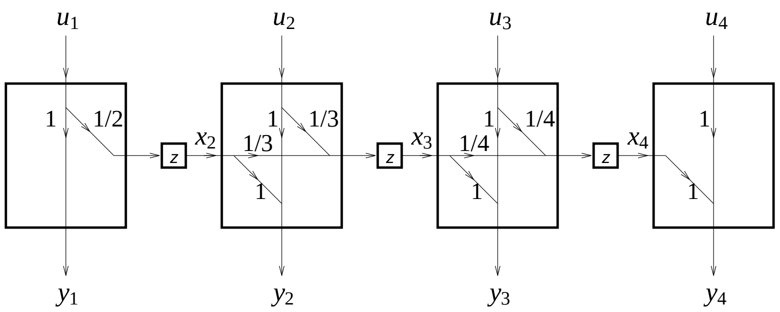
In this paper, we consider a different (complementary) kind of structure in a matrix which applies, for example, to non-stationary signal models. Let $T = [T_{ij}]_{i,j=1}^n$ be a matrix with entries T_{ij}. For added generality, we will allow T to be a block matrix so that the entries are matrices themselves: T_{ij} is an $M_i \times N_j$ matrix, where the dimensions M_i and N_j need not be constant over i and j, and can even be equal to zero at some points. When a (row) vector is viewed as a signal sequence on a finite discrete-time interval, then a vector-matrix multiplication corresponds to the application of a system to the signal. The i-th row of the matrix is the impulse response of the system due to an impulse at time i, and the system is causal if the matrix is block upper.

We will say that T has a *state realization* (computational network) if there exist matrices

$$\mathbf{T}_k = \begin{bmatrix} A_k & C_k \\ B_k & D_k \end{bmatrix}, \qquad \mathbf{T}'_k = \begin{bmatrix} A'_k & C'_k \\ B'_k & 0 \end{bmatrix}, \qquad k = 1, \cdots, n,$$

such that the entries of T are given by

$$T_{ij} = \begin{cases} D_i, & i = j, \\ B_i A_{i+1} \cdots A_{j-1} C_j, & i < j, \\ B'_i A'_{i-1} \cdots A'_{j+1} C'_j, & i > j. \end{cases}$$



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

Figure 1. Computational network corresponding to an upper triangular T.

The state matrices $\{A_k, B_k, C_k, D_k, A'_k, B'_k, C'_k\}_1^n$ must have compatible dimensions in order for the multiplications to make sense, but they need not be square. If $[T_{ij}]$ is described in this way, then a vector-matrix multiplication $z = uT$, where $u = [u_1, u_2, \cdots, u_n]$, $z = [z_1, z_2, \cdots, z_n]$, can be computed via the time-varying forward and backward state recursions

$$\begin{array}{ll} (\mathbf{T}) \left\{ \begin{array}{rcl} x_{k+1} & = & x_k A_k + u_k B_k \\ y_k & = & x_k C_k + u_k D_k \end{array} \right. & \qquad (\mathbf{T}') \left\{ \begin{array}{rcl} x'_{k-1} & = & x'_k A'_k + u_k B'_k \\ y'_k & = & x'_k C'_k \end{array} \right. \\ z_k = y_k + y'_k\,, & \qquad x_1 = [\,\cdot\,]\,, x'_n = [\,\cdot\,]\,. \end{array} \tag{1}$$

Here, $[\,\cdot\,]$ denotes a $0{\times}0$ matrix, x_k is the forward state, and x'_k is the backward state, and these variables can be viewed as the intermediate quantities in the computation. The computation of the vector-matrix product using the state equations is more efficient than a direct multiplication if, for all k, the dimensions of x_k and x'_k are relatively small compared to the matrix size. If this dimension is, on average, equal to d, then a vector-matrix multiplication has complexity $\mathcal{O}(d^2 n)$ (this can be reduced further to $\mathcal{O}(dn)$ by considering special types of realizations), and a matrix inversion has complexity $\mathcal{O}(d^2 n)$ rather than $\mathcal{O}(n^3)$, as we show in section 2.

A simple example of a computational network is depicted in figure 1, in which a state realization is shown of the vector-matrix multiplication $y = uT$, where T is an upper triangular matrix, so that only the forward state recursions are necessary in the computation.

A first question to answer is, given a matrix T, when does there exist a computational network with a low number of states that realizes this matrix. To this end, define the submatrices

$$H_k = \begin{bmatrix} T_{k-1,k} & T_{k-1,k+1} & \cdots & T_{k-1,n} \\ T_{k-2,k} & T_{k-2,k+1} & & \vdots \\ \vdots & & \ddots & T_{2,n} \\ T_{1,k} & \cdots & T_{1,n-1} & T_{1,n} \end{bmatrix}, \quad H'_k = \begin{bmatrix} T_{k,k-1} & T_{k,k-2} & \cdots & T_{k,1} \\ T_{k+1,k-1} & T_{k+1,k-1} & & \vdots \\ \vdots & & \ddots & T_{n-1,1} \\ T_{n,k-1} & \cdots & T_{n,2} & T_{n,1} \end{bmatrix}. \tag{2}$$

The H_k can be called time-varying Hankel matrices, as they would have a Hankel structure in the time-invariant context. In terms of the H_k, we have the following result.

Theorem 1 ([3]) *The minimal dimension of x_k and x'_k of any state realization of T is equal to the rank of H_k and H'_k, respectively.*

Based on factorizations of the H_k, it is possible to derive minimal realizations of a given matrix [3, 4]. It is also possible to compute optimal approximate realizations of lower system order of a given matrix whose Hankel matrices do not possess low rank [5, 4].

2. MATRIX INVERSION

As an example of how time-varying state realizations can be used in matrix computations, we consider the problem of matrix inversion. If the matrix is block upper and has an inverse which is again upper, then it is straightforward to derive a state realization of the inverse of the matrix, given a state realization of the matrix itself.

Theorem 2 ([3]) *Let T be a block upper triangular matrix, whose entries T_{ii} on the main diagonal are square and invertible. Then $S = T^{-1}$ is again block upper triangular. If T has a state realization $\{A_k, B_k, C_k, D_k\}_1^n$, then a realization of S is given by*

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

The next step is to show how a more general matrix T which is not block upper can be mapped by a unitary matrix U^* to block-upper. $((\cdot)^*$ stands for complex conjugate transpose.) We first consider the special case where T is lower triangular. This case is related to the inner-coprime factorization in [5].

Proposition 3 ([5]) *Let T be a block lower matrix, with state realization $\{A_k', B_k', C_k', D_k'\}_1^n$ normalized such that $(A_k')^* A_k' + (B_k')^* B_k' = I$. Then T has a factorization $T = U\Delta$, where U is a unitary block lower matrix and Δ is a block upper matrix. Realizations of U and Δ are given by*

$$\mathbf{U}_k' = \begin{bmatrix} A_k' & C_{U,k}' \\ B_k' & D_{U,k}' \end{bmatrix} , \qquad \mathbf{\Delta}_k = \begin{bmatrix} (A_k')^* & (A_k')^* C_k' + (B_k')^* D_k' \\ (C_{U,k}')^* & (D_{U,k}')^* D_k' + (C_{U,k}')^* C_k' \end{bmatrix}$$

where $\mathbf{U}_k'$ is a square unitary matrix and $C_{U,k}'$ and $D_{U,k}'$ are determined by completing $\left[\begin{smallmatrix} A_k' \\ B_k' \end{smallmatrix}\right]$ to a square unitary matrix, for each k in turn.

The realization for Δ is not necessarily minimal, which is seen, for example, if T is taken to be unitary itself. Because A_k' and B_k' need not have constant dimensions, the number of columns added to obtain $\mathbf{U}_k'$ is not necessarily constant in time, so that the number of outputs of U can be time-varying. In particular, U can be a block matrix whose entries are matrices, even if T itself has scalar entries.

The more general case is a corollary of the above proposition.

Theorem 4 *Let T be a block matrix with realizations $\mathbf{T}' = \{A_k', B_k', C_k', 0\}_1^n$, $\mathbf{T} = \{A_k, B_k, C_k, D_k\}_1^n$, with normalization $(A_k')^* A_k' + (B_k')^* B_k' = I$. Then $T = U\Delta$, with U a block lower unitary matrix, Δ a block upper matrix, having realizations*

$$\mathbf{U}_k' = \begin{bmatrix} A_k' & C_{U,k}' \\ B_k' & D_{U,k}' \end{bmatrix} , \qquad \mathbf{\Delta}_k = \left[\begin{array}{cc|c} (A_k')^* & (B_k')^* B_k & (A_k')^* C_k' + (B_k')^* D_k \\ 0 & A_k & C_k \\ \hline (C_{U,k}')^* & (D_{U,k}')^* B_k & (C_{U,k}')^* C_k' + (D_{U,k}')^* D_k \end{array}\right]$$

where $C_{U,k}'$ and $D_{U,k}'$ are determined by completing $\left[\begin{smallmatrix} A_k' \\ B_k' \end{smallmatrix}\right]$ to a square unitary matrix.

Using the above theorem, a given matrix T is mapped to a block-upper matrix Δ. At this point, it should be remarked that the inverse of a block-upper matrix is not necessarily upper itself. A simple example where T is block upper and T^{-1} is block lower is given by the pair of matrices

$$T = \begin{bmatrix} \underline{1} & 0 & 0 \\ \underline{1/2} & 2 & 0 \\ 0 & \underline{1/4} & 1 \\ \cdot & \cdot & \underline{\cdot} \end{bmatrix} \qquad \begin{array}{lcl} [(M_T)_k] & = & [2\ \ 1\ \ 0] \\ [(N_T)_k] & = & [1\ \ 1\ \ 1] \end{array}$$

$$T^{-1} = \begin{bmatrix} \underline{1} & \underline{0} & 0 & \cdot \\ -1/4 & 1/2 & \underline{0} & \cdot \\ 1/16 & -1/8 & 1 & \underline{\cdot} \end{bmatrix} \qquad \begin{array}{lcl} [(M_{T^{-1}})_k] & = & [1\ \ 1\ \ 1] \\ [(N_{T^{-1}})_k] & = & [2\ \ 1\ \ 0] \end{array}$$

The main diagonal of each matrix consists of the underlined entries. The entries '$\cdot$' are block-matrix entries with one of the dimensions equal to zero. When viewed as matrices without considering their block structure, T^{-1} is of course just the matrix inverse of T. Mixed cases (the inverse has a lower and an upper part) can also occur, and these inverses are not trivially computed, as they require a 'dichotomy': a splitting of spaces into a part that determines the upper part and a part that gives the lower part.

Hence, the final case to consider in order to connect theorem 4 with theorem 2 is a block upper matrix whose inverse is not block upper. Before theorem 2 can be applied, the matrix must be factored into the product of an isometric matrix and an invertible matrix whose inverse is upper again. This QR-factorization is known, in system theory, as the inner-outer factorization. The factorization can be computed in state space terms, as discussed in the following theorem.

Theorem 5 ([6, 4]) *Let T be a block upper matrix. Then T has a factorization $T = VT_0$, where V is upper and an isometry ($V^*V = I$), and T_0 is an upper matrix with upper inverse.*

Realizations can be computed from a realization $\mathbf{T} = \{A_k, B_k, C_k, D_k\}_1^n$ of T as follows. Let $Y_1 = [\,\cdot\,]$, and recursively compute unitary matrices $\mathbf{W}_k$ such that

$$\mathbf{W}_k^* \begin{bmatrix} Y_k & \\ & I \end{bmatrix} \begin{bmatrix} A_k & C_k \\ B_k & D_k \end{bmatrix} = \begin{bmatrix} Y_{k+1} & 0 \\ 0 & 0 \\ * & X_k \end{bmatrix}, \qquad Y_k Y_k^* > 0\,, X_k X_k^* > 0\,. \tag{3}$$

Each $\mathbf{W}_k$ is obtained from a QR-factorization which puts zero matrices of maximal possible dimensions at the indicated positions. Partition the rows of $\mathbf{W}_k$ compatibly with the right hand side of equation (3). Let $(\cdot)^\dagger$ denote a pseudo-inverse. Then realizations of V and T_0 are

$$\mathbf{V}_k = \mathbf{W}_k \begin{bmatrix} I & 0 \\ 0 & 0 \\ 0 & I \end{bmatrix}, \qquad (\mathbf{T}_0)_k \; = \; \begin{bmatrix} I & \\ & X_k^{\dagger *} \end{bmatrix} \begin{bmatrix} A_k & C_k \\ C_k^* Y_k^* Y_k A_k + D_k^* B_k & C_k^* Y_k^* Y_k C_k + D_k^* D_k \end{bmatrix}.$$

Using theorems 4 and 5, a QR-factorization of T follows as $T = (UV)T_0$, where U is block lower and unitary, V is block upper and isometric, and T_0 is *outer*: upper triangular with upper inverse. These matrices have realizations as stated in the theorems, if proper substitutions are made.

If T is invertible, then T^{-1} follows from the above QR-factorization as $T^{-1} = T_0^{-1}V^*U^*$, where T_0^{-1} is upper triangular and has a realization that is obtained from that of T_0 using theorem

2. The realizations of V^* and U^* follow trivially from those of V and U. All calculations in determining the factorization are forward recursions based on state realizations; only the explicit computation of the inverse requires backward recursions in order to compute the transfer matrix of the given realization.

NUMERICAL EXAMPLE

To illustrate the above theorems with a numerical example, consider the following matrix T:

$$T = \begin{bmatrix} 0.2500 & 0.0500 & 0.0270 & -0.0056 & -0.0119 & -0.0081 \\ 0.0276 & 0.5550 & 0.0250 & 0.0910 & 0.0558 & 0.0219 \\ 0.0183 & 0.6055 & 0.3415 & 0.0350 & 0.0883 & 0.0615 \\ 0.0089 & 0.2927 & 0.5191 & 0.3428 & 0.0495 & 0.0855 \\ 0.0038 & 0.1268 & 0.2249 & 0.5159 & 0.3442 & 0.0500 \\ 0.0022 & 0.0728 & 0.1291 & 0.2961 & 0.6017 & 0.5576 \end{bmatrix}$$

Computation of the ranks of the time-varying Hankel matrices H_k and H'_k defined in equation (2) reveals that

$$\begin{aligned} [d_k]_1^6 &= [\text{rank } H_k]_1^6 &= [0\ \ 1\ \ 2\ \ 2\ \ 2\ \ 1] \\ [d'_k]_1^6 &= [\text{rank } H'_k]_1^6 &= [0\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1] \end{aligned}$$

so that, according to theorem 1, T has a time-varying state realization $\mathbf{T}$, $\mathbf{T}'$ with at most two states in the forward recursions, and at most one state in the backward recursion. Such realizations can be computed using the theory presented in more detail in [3, 4].

Using a realization of T in output normal form, the inner-coprime factors U and Δ in the factorization $T = U\Delta$ are computed as

$$U = \begin{bmatrix} \underline{0} & \underline{1.000} & 0 & 0 & 0 & 0 & \cdot \\ -0.798 & 0 & \underline{0.603} & 0 & 0 & 0 & \cdot \\ -0.531 & 0 & -0.702 & \underline{0.475} & 0 & 0 & \cdot \\ -0.257 & 0 & -0.339 & -0.787 & \underline{0.447} & 0 & \cdot \\ -0.111 & 0 & -0.147 & -0.341 & -0.776 & \underline{0.498} & \cdot \\ -0.064 & 0 & -0.084 & -0.196 & -0.445 & -0.867 & \underline{\cdot} \end{bmatrix} \qquad \begin{aligned} [(M_U)_k] &= [1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1] \\ [(N_U)_k] &= [2\ \ 1\ \ 1\ \ 1\ \ 1\ \ 0] \\ [(d'_U)_k] &= [0\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1] \end{aligned}$$

$$\Delta = \begin{bmatrix} \underline{-0.035} & -0.858 & -0.368 & -0.255 & -0.181 & -0.113 \\ \underline{0.250} & 0.050 & 0.027 & -0.006 & -0.012 & -0.008 \\ 0 & \underline{0.214} & -0.445 & -0.187 & -0.147 & -0.113 \\ 0 & 0 & \underline{-0.348} & -0.487 & -0.232 & -0.164 \\ 0 & 0 & 0 & \underline{-0.379} & -0.513 & -0.249 \\ 0 & 0 & 0 & 0 & \underline{-0.351} & -0.459 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \underline{\cdot} \end{bmatrix} \qquad \begin{aligned} [(M_\Delta)_k] &= [2\ \ 1\ \ 1\ \ 1\ \ 1\ \ 0] \\ [(N_\Delta)_k] &= [1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1] \\ [(d_\Delta)_k] &= [0\ \ 2\ \ 3\ \ 3\ \ 2\ \ 1] \end{aligned}$$

These matrices are block-matrices: U_{ij} is a matrix of size $(M_U)_i \times (N_U)_j$. Note that the dimensions of some of the blocks are vanishing, signified by the '$\cdot$'-entries. Such entries occur automatically in the formalism whenever, in theorem 4, $\left[\begin{smallmatrix} A'_k \\ B'_k \end{smallmatrix}\right]$ is already a square unitary matrix.

The next step is to factor Δ into $\Delta = VT_0$, where V is upper and an isometry, and T_0 is upper with upper inverse. Theorem 5 yields

$$V = \begin{bmatrix} \underline{-0.137} & -0.960 & 0.171 & -0.098 & 0.074 & 0.124 \\ \underline{0.991} & -0.133 & 0.024 & -0.014 & 0.010 & 0.017 \\ 0 & \underline{-0.247} & -0.679 & 0.388 & -0.295 & -0.491 \\ 0 & 0 & \underline{-0.713} & -0.393 & 0.299 & 0.497 \\ 0 & 0 & 0 & \underline{-0.828} & -0.289 & -0.481 \\ 0 & 0 & 0 & 0 & \underline{-0.857} & 0.515 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \underline{\cdot} \end{bmatrix} \qquad \begin{array}{lcl} [(M_V)_k] & = & [2\ 1\ 1\ 1\ 1\ 0] \\ [(N_V)_k] & = & [1\ 1\ 1\ 1\ 1\ 1] \\ [(d_V)_k] & = & [0\ 1\ 1\ 1\ 1\ 1] \end{array}$$

$$T_0 = \begin{bmatrix} \underline{0.252} & 0.167 & 0.077 & 0.029 & 0.013 & 0.008 \\ 0 & \underline{0.870} & 0.459 & 0.292 & 0.211 & 0.138 \\ 0 & 0 & \underline{0.488} & 0.430 & 0.234 & 0.175 \\ 0 & 0 & 0 & \underline{0.458} & 0.477 & 0.238 \\ 0 & 0 & 0 & 0 & \underline{0.409} & 0.441 \\ 0 & 0 & 0 & 0 & 0 & \underline{-0.157} \end{bmatrix} \qquad \begin{array}{lcl} [(M_{T_0})_k] & = & [1\ 1\ 1\ 1\ 1\ 1] \\ [(N_{T_0})_k] & = & [1\ 1\ 1\ 1\ 1\ 1] \\ [(d_{T_0})_k] & = & [0\ 1\ 2\ 3\ 2\ 1] \end{array}$$

We have thus obtained the QR-factorization $T = UVT_0$, where T_0 is outer and can be inverted straightforwardly. It should be stressed that all computations can be kept in state space terms so that U, V and T_0 are never computed explicitly. This is important in large structured matrix applications, where the number of points in time (the number of rows and columns of T) is large, but the state dimensions are relatively small. In such cases, the discussed connection with time-varying systems provides an efficient algorithm to compute the exact inverse of a given matrix.

REFERENCES

1. I. Gohberg and A. Semencul, “On the Inversion of Finite Toeplitz Matrices and their Continuous Analogs,” *Mat. Issled.*, vol. 2, pp. 201–233, 1972.
2. 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.
3. A.J. van der Veen and P.M. Dewilde, “Time-varying Computational Networks: Realization, Orthogonal Embedding and Structural Factorization,” in *Proc. SPIE, “Advanced Signal Processing Algorithms, Architectures, and Implementations”, III* (F.T. Luk, ed.), vol. 1770, (San Diego), pp. 164–177, July 1992.
4. 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.
5. P.M. Dewilde and A.J. van der Veen, “On the Hankel-Norm Approximation of Upper-Triangular Operators and Matrices,” *to appear in Integral Equations and Operator Theory*, 1993.
6. A.J. van der Veen, “Computation of the Inner-Outer Factorization for Time-varying Systems,” in *Challenges of a Generalized System Theory* (M. Verhaegen et al., ed.), Essays of the Royal Dutch Academy of Sciences, (Amsterdam, The Netherlands), 1993.