Large Matrix Inversion using State Space Techniques

Alle-Jan van der Veen and Patrick Dewilde

A new computational technique is presented by which large structured matrices can be inverted. The specified matrix is viewed as the input-output operator of a time-varying system. Recently developed state space algorithms which apply to such systems are then used to compute a QR factorization first and subsequently the inverse of the matrix, starting from a state realization of the matrix. The new algorithms apply in principle to any matrix. They are efficient if the structure of the matrix is such that the number of states of its time-varying state realization is small in comparison to its dimensions.

1. INTRODUCTION

In a number of applications in signal processing, such as inverse filtering, spectrum estimation, as well as in certain finite element modeling applications, the basic algebraic operation consists of a QR factorization, a Cholesky factorization, or a matrix inversion. Such matrices can be fairly large but, due to the properties of the signals or physical geometry from which the matrices originate, the matrices are not fully random but are structured in some way. For example, in stationary environments, the estimated covariance matrices of measured signals have a Toeplitz structure, and efficient algorithms (Schur recursions) exist to factor such matrices or their inverse. Schur recursions can be generalized to apply to general Toeplitz matrices [1]. The computation of the inverse of a Toeplitz matrix goes via Gohberg/Semencul recursions [2]. The resulting algorithms have computational complexity of order (n^2) for matrices of size $(n \times n)$, as compared to (n^3) for algorithms that do not take the Toeplitz structure into account. For large matrices with many zero entries, the inverse (or rather, the application of the inverse to a vector) can be computed iteratively using the Lanczos method.

In this paper, we consider matrices with a different structure, which would correspond, for example, to applications with non-stationary signals. The underlying idea is to model a given matrix by a time-varying state realization. Such a representation is fairly general: any matrix can be modeled in this way. Efficient

1993 IEEE Workshop on VLSI Signal Processing, Veldhoven, The Netherlands.

algorithms will result if the state dimension of the time-varying realization is relatively low in comparison with the size of the matrix.

Using algorithms recently developed for the factorization of time-varying systems (they are generalizations of the corresponding time-invariant results), it is now possible to compute QR factorizations, Cholesky factorizations, and matrix inversions, by acting on state space matrices only. The computational complexity is thus shown to be *linear* in the size of the matrix, once a low-dimensional state realization of it is known. Some of these results are collected in this paper, and applied to the computation of the inverse of a large structured matrix.

Matrix representation by time-varying state realizations

Let $T = [T_{ij}]_{i,j=1}^{n}$ be a matrix with entries T_{ij}. For additional generality, we will allow T to be a block matrix so that its 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 time interval in discrete time, then the multiplication of a vector by this matrix,

$$[y_1 \quad y_2 \qquad\qquad y_n] = [u_1 \quad u_2 \qquad\qquad u_n]T\,,$$

corresponds to the application of the related system to the signal represented by u. The i-th row of the matrix is the impulse response of the system due to an impulse at time i, *i.e.*, an input vector $u = [0 \qquad 0 \;\; 1 \;\; 0 \qquad 0]$. The system is causal if the matrix is block upper.

Let $\mathbf{T}_k \;{}_1^n$, $\mathbf{T}_k \;{}_1^n$ be a series of matrices with block entries

$$\mathbf{T}_k = \begin{matrix} A_k & C_k \\ B_k & D_k \end{matrix} \quad , \qquad \mathbf{T}_k = \begin{matrix} A_k & C_k \\ B_k & 0 \end{matrix} \quad , \qquad k = 1, \qquad , n\,.$$

and consider the time-varying forward and backward state recursions, for $k = 1, \qquad , n$,

$$(\mathbf{T}) \quad \begin{matrix} x_{k+1} & = & x_k A_k + u_k B_k \\ y_k & = & x_k C_k + u_k D_k \end{matrix} \qquad\qquad (\mathbf{T}\;\;) \quad \begin{matrix} x_{k-1} & = & x_k A_k + u_k B_k \\ y_k & = & x_k C_k \end{matrix} \tag{1}$$

$$z_k = y_k + y_k\,, \qquad\qquad x_1 = [\,\cdot\,]\,, x_n = [\,\cdot\,]\,.$$

Here, $[\,\cdot\,]$ denotes a matrix in which one (or both) dimensions are vanishing. The intermediate quantities in the recursion are x_k, the forward state, and x_k, the backward state. The 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 or have constant dimensions. The relation between $u = [u_1, u_2, \qquad , u_n]$

and $z = [z_1, z_2, \quad , z_n]$, as generated by the above state recursions, is

$$z \;=\; u \quad \begin{array}{cccccc} D_1 & B_1C_2 & B_1A_2C_3 & B_1A_2A_3C_4 & & \\ B_2C_1 & D_2 & B_2C_3 & B_2A_3C_4 & & \\ B_3A_2C_1 & B_3C_2 & D_3 & B_3C_4 & & \vdots \\ B_4A_3A_2C_1 & B_4A_3C_2 & B_4C_3 & D_4 & & B_{n-2}A_{n-1}C_n \\ \vdots & & & \vdots & \ddots & B_{n-1}C_n \\ & & & B_nA_{n-1}C_{n-2} & B_nC_{n-1} & D_n \end{array}$$

so that the state recursions can be used to compute a vector-matrix multiplication $z = uT$, where the matrix T is of the above form. Accordingly, we will say that a matrix T has a (time-varying) *state realization* if there exist matrices $\mathbf{T}_k \; {}^{n}_{1}$, $\mathbf{T}_k \; {}^{n}_{1}$ such that the block entries of T are given by

$$T_{ij} = \quad \begin{array}{lll} D_i\,, & & i = j\,, \\ B_iA_{i+1} & A_{j-1}C_j\,, & i < j\,, \\ B_i\,A_{i-1} & A_{j+1}C_j\,, & i > j\,. \end{array}$$

The computation of a 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 (d^2n) (this can be reduced further to (dn) by considering special types of realizations), and a matrix inversion has complexity (d^2n) rather than (n^3), as we show in section 2.

Computation of a state realization

At this point, a first question that emerges is whether, for any given matrix, a state realization exists. If so, then subsequent questions are (*i*) how to find it, and (*ii*) what will be its complexity. To answer these questions, define the submatrices

$$H_k \quad = \quad \begin{array}{cccc} T_{k-1,k} & T_{k-1,k+1} & & T_{k-1,n} \\ T_{k-2,k} & T_{k-2,k+1} & & \vdots \\ \vdots & & \ddots & T_{2,n} \\ T_{1,k} & & T_{1,n-1} & T_{1,n} \end{array} \tag{2}$$

$$H_k \quad = \quad \begin{array}{cccc} T_{k,k-1} & T_{k,k-2} & & T_{k,1} \\ T_{k+1,k-1} & T_{k+1,k-2} & & \vdots \\ \vdots & & \ddots & T_{n-1,1} \\ T_{n,k-1} & & T_{n,2} & T_{n,1} \end{array} \quad . \tag{3}$$

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, 4])** *Let T be an $n \times n$ matrix, and for $k = 1, \quad , n$, let $d_k =$ rank(H_k), $d_k = \mathrm{rank}(H_k)$. Then there are time-varying state realizations that realize T, and the minimal dimension of x_k and x_k of any state realization of T is equal to d_k and d_k, respectively.*

Let $H_k = Q_k R_k = [Q_{1,k} \quad Q_{2,k}] \begin{smallmatrix} R_{1,k} \\ 0 \end{smallmatrix}$ be a QR factorization of H_k, where Q_k is a unitary matrix and R_k has the indicated structure such that $R_{1,k}$ has rank d_k. Likewise, let $H_k = Q_k R_k = [Q_{1,k} \;\; Q_{2,k}] \begin{smallmatrix} R_{1,k} \\ 0 \end{smallmatrix}$. Then a realization of T is given by

$$\mathbf{T}: \begin{array}{lcl} A_k &=& [0 \quad Q_{1,k}^*]Q_{1,k+1} \\ B_k &=& (Q_{1,k+1})(1,:) \\ C_k &=& R_{1,k}(:,1) \\ D_k &=& T_{k,k} \end{array} \qquad \mathbf{T}: \begin{array}{lcl} A_k &=& [0 \quad Q_{1,k+1}^*]Q_{1,k} \\ B_k &=& Q_{1,k}(1,:) \\ C_k &=& R_{1,k+1}(:,1) \\ D_k &=& 0\,. \end{array}$$

In this theorem, $(\cdot)^*$ stands for complex conjugate transpose. For a matrix X, the notation $X(1,:)$ denotes the first row of X, and $X(:,1)$ the first column.

Hence, the state dimension of the realization (which determines the computational complexity of multiplications and inversions using state realizations) is directly related to the ranks of the Hankel matrices. Once factorizations of the H_k and H_k for $k = 1, \quad , n$ are known, it is possible to derive minimal realizations of a given matrix. The realization formulas given above yield a realization that is in input normal form: it satisfies $A_k^* A_k + B_k^* B_k = I$ and $A_k{}^* A_k + B_k{}^* B_k = I$.

A realization algorithm that is less sensitive to the presence of additive noise on the entries of T would use singular value decompositions (SVDs) of the Hankel matrices, rather than the QR factorization, and adjust their rank by setting small singular values equal to zero. It is also possible to compute optimal approximate realizations of lower system order [5]. The derivation of the factorizations is computationally the most demanding part of the whole procedure. Improvements can be obtained by using updating schemes for the factorizations (since H_k and H_{k+1} have many entries in common), and by considering submatrices of H_k (as it is known that if H_k has rank d_k, then it is enough to consider a submatrix of H_k whose rank is also equal to d_k [6, 4]).

2. MATRIX INVERSION

In this section, we will show how a state realization of the inverse of a matrix can be computed from a state realization of the given matrix. We start by considering a simple case, in which the matrix is block upper and it is known that its inverse is again block upper. Such a matrix, when viewed as a system, is known in system theory language as being *outer* (minimal phase system). Not all block-upper matrices are outer: simple examples where T is block upper and T^{-1} is block lower are given in [4]. 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. The dichotomy can be computed using an inner-outer factorization (theorem 6 below).

For the case of a general matrix (mixed upper-lower), it is shown how this matrix can be mapped by a unitary matrix to a block upper matrix. However, as the inverse of this matrix is possibly not block upper. it is in general necessary to perform the inner-outer factorization, which factors the matrix into a unitary block upper matrix (whose inverse is block lower and obtained by a simple transposition) and a block upper matrix whose inverse is known to be upper, too.

State complexity of the inverse

Suppose that T is an invertible matrix with a state realization of low complexity. We will first show that (under conditions) the inverse has a state realization of the same complexity.

Proposition 2. *Let T be an invertible $n \times n$ matrix with Hankel matrices $(H_T)_k$ and $(H_T)_k$ defined by (2), (3), for $k = 1, \quad , n$. Put $\mathrm{rank}\,(H_T)_k = d_k$ and $\mathrm{rank}(H_T)_k = d_k$.*

If, for each k, at least one of the submatrices $[T_{ij}]_{i,j=1}^{k-1}$ or $[T_{ij}]_{i,j=k}^{n}$ is invertible, then $S = T^{-1}$ has Hankel matrices with the same ranks: $\mathrm{rank}\,(H_S)_k = d_k$ and $\mathrm{rank}(H_S)_k = d_k$.

(It is conjectured that the proposition is still true if the condition on the invertibility of the submatrices is lifted.)

Proof We will use Schur's inversion lemma. In general, let A, B, C, D be matrices such that A and D are square, and A is invertible, then

$$\begin{matrix} A & B \\ C & D \end{matrix} = \begin{matrix} I & 0 \\ CA^{-1} & I \end{matrix} \quad \begin{matrix} A & 0 \\ 0 & D - CA^{-1}B \end{matrix} \quad \begin{matrix} I & A^{-1}B \\ 0 & I \end{matrix} \quad .$$

If in addition the inverse of this block matrix exists, then $D - CA^{-1}B$ is invertible and the inverse of the block matrix is given by

$$\begin{aligned} \begin{matrix} A & B \\ C & D \end{matrix} &= \begin{matrix} I & -A^{-1}B \\ 0 & I \end{matrix} \quad \begin{matrix} A^{-1} & 0 \\ 0 & (D - CA^{-1}B)^{-1} \end{matrix} \quad \begin{matrix} I & 0 \\ -CA^{-1} & I \end{matrix} \\ &= \begin{matrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{matrix} \quad . \end{aligned}$$

In particular, D is invertible, $\mathrm{rank}\,B$ $= \mathrm{rank}\,B$, $\mathrm{rank}\,C$ $= \mathrm{rank}\,C$. The proposition follows if $\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}$ is taken to be a partioning of T, such that $B = (H_T)_k$ and $C = (H_T)_k$.

Inversion of an upper matrix with upper inverse

If a matrix is block upper and has an inverse which is again block upper (*i.e.*, the corresponding time-varying system is outer), then it is straightforward to derive a state realization of the inverse of the matrix, given a state realization of the matrix itself. The realization can even be obtained locally: it is, at point k, only dependent on the realization of the given matrix at point k.

Theorem 3. **([7])** *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 $\mathbf{T} = \; A_k, B_k, C_k, D_k \; {}_1^n$, then a realization of S is given by*

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

The theorem is proved merely by rewriting the state equations corresponding to $\mathbf{T}$ in (1).

Mapping a matrix to block-upper

In order to use the above inversion theorem on a matrix T which is not block upper, we compute a kind of QR factorization of T as $T = U\Delta$, where U is block lower and unitary, and Δ is block upper. Since U is unitary, its inverse is equal to its Hermitian transpose and can trivially be obtained. We first consider the special case where T is lower triangular. This case is related to the inner-coprime factorization in [5].

Proposition 4. **([5])** *Let T be a block lower matrix, with state realization $\mathbf{T} = \; A_k, B_k, C_k, D_k \; {}_1^n$ 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*

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

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

Note that the realization of T as obtained using theorem 1 has already the required normalization. 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 inputs and 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 5. *Let T be a block matrix with realizations $\mathbf{T} = A_k, B_k, C_k, 0\ {}_1^n$, $\mathbf{T} = A_k, B_k, C_k, D_k\ {}_1^n$. Then $T = U\Delta$, with U a block lower unitary matrix, Δ a block upper matrix, having realizations*

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

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

Inner-outer factorization

At this point, we have transformed a general matrix T to a block upper matrix Δ. In order to use theorem 3 to find the inverse of a block upper matrix, it must be known that the inverse is again upper. As discussed at the beginning of this section, this will not necessarily be the case for Δ. Hence, the final case to consider in order to connect theorem 5 with theorem 3 is a block upper matrix which is not invertible or whose inverse is not block upper. Before theorem 3 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, according to the following theorem.

Theorem 6. **([8, 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.*

Let $\mathbf{T} = A_k, B_k, C_k, D_k\ {}_1^n$ be a realization of T, and put $Y_1 = [\,\cdot\,]$. Recursively compute unitary matrices $\mathbf{W}_k$ such that the following product has zeros of maximal possible dimensions at the indicated positions:

$$\mathbf{W}_k^* \begin{matrix} Y_k & \\ & I \end{matrix} \begin{matrix} A_k & C_k \\ B_k & D_k \end{matrix} = \begin{matrix} Y_{k+1} & 0 \\ 0 & 0 \\ (B_{T_0})_k & (D_{T_0})_k \end{matrix} \,, \qquad \begin{matrix} Y_{k+1} Y_{k+1}^* > 0 \,, \\ (D_{T_0})_k (D_{T_0})_k^* > 0 \,. \end{matrix} \tag{4}$$

Partition the rows of $\mathbf{W}_k$ compatibly with the right hand side of equation (4). Then realizations of V and T_0 are

$$\mathbf{V}_k = \mathbf{W}_k \begin{matrix} I & 0 \\ 0 & 0 \\ 0 & I \end{matrix} \,, \qquad (\mathbf{T}_0)_k = \begin{matrix} A_k & C_k \\ (B_{T_0})_k & (D_{T_0})_k \end{matrix} \,.$$

Note that each $\mathbf{W}_k$ can be obtained from a QR-factorization of the matrix $\begin{matrix} D_k & B_k \\ Y_kC_k & Y_kA_k \end{matrix}$. Since we obtain a factorization $T = VT_0$, where V is an isometry and T_0 is invertible, the inner-outer factorization is a kind of QR factorization. It is, in a sense, remarkable that it can be computed using QR factorizations of state space matrices.

Inversion of a general matrix

At this point, all ingredients are present for the computation of a matrix inverse using state space techniques. We will assume that, for a given matrix T, a time-varying state realization $\mathbf{T}, \mathbf{T}$ has been derived using theorem 1. Such a realization is in input normal form. Subsequently, using theorems 5 and 6, 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 an upper triangular invertible matrix 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 3. The realizations of V^* and U^* follow trivially from those of V and U. We have thus obtained a realization of the factors of T^{-1}. A realization of T^{-1} itself can be derived, if necessary, by computing a combined realization of the products. Again, without special effort such a realization will not necessarily be minimal, *i.e.,* its state dimensions can be larger than the minimal state dimension as derived in proposition 2.

3. CONCLUDING REMARKS

In this paper, a recently developed technique to invert a matrix has been presented. It is intended for the inversion of large matrices that have time-varying state realizations with a low number of states. The derived algorithm acts solely on state space realizations and is a single-directional (forward) recursion consisting of local state matrix multiplications and QR factorizations. It results in state realizations of the factors of a QR factorization of the inverse. Starting from a state realization of T, the computational complexity is (nd^2), where n is the size of the matrix and d is the average number of states at each point. A number of questions are remaining for future research. We mention (1) starting from the realization procedure in theorem 1, what is a more efficient way to compute a minimal state realization of T (possibly with certain assumptions on T, and combined with an approximation scheme), (2) does there exist a technique so that the complexity of the derived realization of T^{-1} minimal, *i.e.,* equal to the complexity claimed in proposition 2, and (3) what is the numerical stability of the technique.

References

[1] 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.

[2] I. Gohberg and A. Semencul, “On the Inversion of Finite Toeplitz Matrices and their Continuous Analogs,” *Mat. Issled.*, vol. 2, pp. 201–233, 1972.

[3] 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.

[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] I. Gohberg, M.A. Kaashoek, and L. Lerer, “Minimality and Realization of Discrete Time-Varying Systems,” in *Time Variant Systems and Interpolation* (I. Gohberg, ed.), vol. OT 56, pp. 261–296, Birkḧauser Verlag, 1992.

[7] 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.

[8] 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.

Acknowledgements

This research was supported in part by the commission of the EC under the ESPRIT BRA program 6632 (NANA2).

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