FAST IMPLEMENTATION OF THE QR FACTORIZATION IN SUBSPACE IDENTIFICATION

N. Mastronardi, P. Van Dooren, S. Van Huffel

Università della Basilicata, Potenza, Italy, Katholieke Universiteit Leuven, Leuven, Belgium
Catholic University of Louvain, Louvain-la-Neuve, Belgium
Katholieke Universiteit Leuven, Leuven, Belgium

Two recent approaches [4, 14] in subspace identification problems require the computation of the R factor of the QR factorization of a block–Hankel matrix H, which, in general has a huge number of rows. Since the data are perturbed by noise, the involved matrix H is, in general, full rank. It is well known that, from a theoretical point of view, the R factor of the QR factorization of H is equivalent to the Cholesky factor of the correlation matrix H^TH, apart from a multiplication by a sign matrix. In [12] a fast Cholesky factorization of the correlation matrix, exploiting the block–Hankel structure of H, is described. In this paper we consider a fast algorithm to compute the R factor based on the generalized Schur algorithm. The proposed algorithm allows to handle the rank–deficient case.

Keywords: identification methods; least squares solutions; multivariable systems; singular values.

INTRODUCTION

Subspace based system identification techniques have been studied intensively in the last decades [4]. The major drawbacks of these direct state–space identification techniques are the high computation and storage costs.

Let u_k and y_k be the m–dimensional input vector and the l–dimensional output vector, respectively, of the basic linear time–invariant state space model

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k, \\ y_k &= Cx_k + Du_k + v_k, \end{aligned}$$

where x_k is the n–dimensional state vector at time k, $\{w_k\}$ and $\{v_k\}$ are state and output disturbances or noise sequences, and A, B, C and D are unknown real matrices of appropriate dimensions.

For non–sequential data processing, the $N \times 2(m+l)s$, $N \gg 2(m+l)s$, matrix $H = [U_{1,2s,N}^T Y_{1,2s,N}^T]$ is constructed, where $U_{1,2s,N}$ and $Y_{1,2s,N}$ are block–Hankel matrices defined in terms of the input and output data, respectively,

$$U_{1,2s,N} = \begin{bmatrix} u_1 & u_2 & u_3 & \cdots & u_N \\ u_2 & u_3 & u_4 & \cdots & u_{N+1} \\ u_3 & u_4 & u_5 & \cdots & u_{N+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ u_{2s} & u_{2s+1} & u_{2s+2} & \cdots & u_{N+2s-1} \end{bmatrix},$$

$$Y_{1,2s,N} = \begin{bmatrix} y_1 & y_2 & y_3 & \cdots & y_N \\ y_2 & y_3 & y_4 & \cdots & y_{N+1} \\ y_3 & y_4 & y_5 & \cdots & y_{N+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{2s} & y_{2s+1} & y_{2s+2} & \cdots & y_{N+2s-1} \end{bmatrix}.$$

Then the R factor of a QR factorization $H = QR$, is used for data compression. In [12] a fast Cholesky factorization of the correlation matrix, exploiting the block–Hankel structure of H, is described. In this paper we consider a fast algorithm to compute the R factor of the QR factorization of H based on the generalized Schur algorithm, exploiting its *displacement structure*. The implementation of the algorithm depends, in particular, on the involved matrix. The paper is organized as follows. First the generalized Schur algorithm to compute the Cholesky factor of a symmetric positive definite matrix is described. Then the generalized Schur algorithm applied to the matrix H is considered. The rank–deficient case is also discussed. Finally, some numerical experiments are reported.

THE GENERALIZED SCHUR ALGORITHM FOR SYMMETRIC POSITIVE DEFINITE MATRICES

In this section the key properties of the generalized Schur algorithm to compute the Cholesky factor of a symmetric positive definite (s.p.d.) matrix, which will be used in the next section, are summarized. More details can be found in [7, 8].

Let $\hat{A}$ and $\hat{Z}$ be a s.p.d. matrix and a shift matrix of order

$\hat{N}$

$$\hat{Z} = \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ 1 & \ddots & \ddots & \vdots \\ & \ddots & \ddots & \vdots \\ 0 & & 1 & 0 \end{bmatrix},$$

respectively. Let

$$\nabla \hat{A} = \hat{A} - \hat{Z}\hat{A}\hat{Z}^T.$$

Then rank$(\nabla \hat{A})$ is called the *displacement rank* of $\hat{A}$. If the displacement rank of $\hat{A}$ is $\hat{M}$, then

$$\nabla \hat{A} = \sum_{i=1}^{\hat{p}} \hat{g}_i^{(+)} \hat{g}_i^{(+)T} - \sum_{i=1}^{\hat{q}} \hat{g}_i^{(-)} \hat{g}_i^{(-)T}, \qquad (1)$$

$\hat{M} = \hat{p} + \hat{q}$.

The vectors $\hat{g}_i^{(+)}, \hat{g}_i^{(-)} \in \mathbb{R}^{\hat{N}}$ are called the *positive* and the *negative generators* of $\hat{A}$, respectively. Let

$$\hat{G} = \left[\; \hat{g}_1^{(+)}, \cdots, \hat{g}_{\hat{p}}^{(+)}, \hat{g}_1^{(-)}, \cdots, \hat{g}_{\hat{q}}^{(-)} \; \right]^T.$$

The *generator matrix* $\hat{G}$ is said to be in *proper form* if the entries in the first non–zero column are equal to zero, except for a non-zero in the first row. Define I_r the identity matrix of order r. Let $J_{\hat{p},\hat{q}} = I_{\hat{p}} \oplus (-I_{\hat{q}})$, where the symbol $\oplus$ denotes the concatenated direct sum, i.e., $A \oplus B = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}$. Then (1) can be written in the following way

$$\nabla \hat{A} = \hat{G}^T J_{\hat{p},\hat{q}} \hat{G}.$$

A matrix $\hat{Q}$ is said $J_{\hat{p},\hat{q}}$–orthogonal if $\hat{Q}^T J_{\hat{p},\hat{q}} \hat{Q} = J_{\hat{p},\hat{q}}$. A generator matrix is not unique. In fact, if $\hat{Q}$ is a $J_{\hat{p},\hat{q}}$–orthogonal matrix, then $\hat{Q}\hat{G}$ is a generator matrix, too.

The following theorem holds [7].

Theorem 1 *Let $\hat{A}_1$ be a s.p.d. matrix. Let $\hat{G}^{(1)} = [g_1^{(1,+)}, \ldots, g_{\hat{p}}^{(1,+)}, g_1^{(1,-)}, \ldots, g_{\hat{q}}^{(1,-)}]^T$ be a generator matrix in proper form of $\hat{A}_1$ with respect to the shift matrix $\hat{Z}$. Then ${g_1^{(1,+)}}^T$ is the first row of $\hat{R}$, the Cholesky factor[1] of $\hat{A}$. Furthermore the generator matrix for the Schur complement*

$$\hat{A}_2 = \hat{A}_1 - g_1^{(1,+)} g_1^{(1,+)T}$$

is given by

$$\hat{G}^{(2)} = \left[\; \hat{Z} g_1^{(1,+)}, \cdots, g_{\hat{p}}^{(1,+)}, g_1^{(1,-)}, \cdots, \hat{g}_{\hat{q}}^{(1,-)} \; \right]^T.$$

[1]Let F be a full rank rectangular matrix. Then $F^T F$ is a s.p.d. matrix, with Cholesky decomposition $\tilde{R}^T \tilde{R}$, $\tilde{R}$ upper triangular. If $\breve{R}$ is the R factor of the QR decomposition of F, it is well known that $\breve{R} = \Theta \tilde{R}$, where $\Theta = \text{diag}(\pm 1, \ldots, \pm 1)$. Hence both the problem of computing the R factor of the QR factorization of F and that of computing the Cholesky factor of $F^T F$ are equivalent.

We observe that the entries of the first row (and of the first column) of $\hat{A}_2$, are equal to 0. The entries of the first column of $\hat{G}^{(2)}$ are equal to 0, too. To compute the second row of the Cholesky factor, we need to find a $J_{\hat{p},\hat{q}}$–orthogonal matrix $\hat{Q}_2$ such that $\tilde{G}^{(2)} = \hat{Q}_2 \hat{G}^{(2)}$ is in proper form. Then the first row of $\tilde{G}^{(2)}$ becomes the second row of $\hat{R}$.

If we pose

$$\tilde{G}^{(2)} = [g_1^{(2,+)}, \ldots, g_{\hat{p}}^{(2,+)}, g_1^{(2,-)}, \ldots, g_{\hat{q}}^{(2,-)}]^T,$$

then a generator matrix for the Schur complement

$$\hat{A}_3 = \hat{A}_2 - g_1^{(2,+)} g_1^{(2,+)T}$$

is given by

$$\hat{G}^{(3)} = [\hat{Z} g_1^{(2,+)}, \ldots, g_{\hat{p}}^{(2,+)}, g_1^{(2,-)}, \ldots, g_{\hat{q}}^{(2,-)}]^T.$$

By successively repeating the described procedure, we obtain the triangular factor $\hat{R}$.

Generalized Schur algorithm

Input: $\hat{G}^{(1)} = [g_1^{(1,+)}, \ldots, g_{\hat{p}}^{(1,+)}, g_1^{(1,-)}, \ldots, g_{\hat{q}}^{(1,-)}]^T$
Output: $\hat{R}$
`for` $i = 1 : \hat{N}$,
  Find a $J_{\hat{p},\hat{q}}$–orthogonal matrix $\hat{Q}_i$
  such that $\tilde{G}^{(i)} = \hat{Q}_i \hat{G}^{(i)}$ is in proper form.
  i–th row of $\hat{R}$:= first row of $\tilde{G}^{(i)}$
  $\hat{G}^{(i+1)} = [\hat{Z} g_1^{(i,+)}, \ldots, g_{\hat{p}}^{(i,+)}, g_1^{(i,-)}, \ldots, g_{\hat{q}}^{(i,-)}]^T$
`end`

The computational complexity is $O(\hat{M}\hat{N}^2)$. Hence the generalized Schur algorithm can be efficiently used when $\hat{M} \ll \hat{N}$, i.e., when $\hat{A}$ is a Toeplitz–like matrix.

The next section describes the particular choice of $\hat{Q}_i, \hat{G}^{(i)}$ and $\hat{Z}$ for solving the System Identification problem.

REDUCTION OF THE GENERATORS MATRIX IN PROPER FORM

At each iteration of the generalized Schur algorithm a $J_{\hat{p},\hat{q}}$–orthogonal matrix Q_i is chosen in order to reduce the generator matrix in proper form. This can be accomplished in a few different ways (see for instance [10, 8, 13]). However, for the sake of the stability, we consider an implementation that allow us to compute the R factor also when the matrix H is rank deficient.

First of all, we observe that the product of two $J_{\hat{p},\hat{q}}$–orthogonal matrices is a $J_{\hat{p},\hat{q}}$–orthogonal matrix. Furthermore, if Q_1 and Q_2 are orthogonal matrices of order $\hat{p}$ and $\hat{q}$, respectively, then the orthogonal matrix $Q = Q_1 \oplus Q_2$ is a $J_{\hat{p},\hat{q}}$–orthogonal matrix, too.

If $|\rho| < 1$, the matrix product, called *modified hyperbolic rotation* [1],

$$\begin{aligned} T &= \begin{bmatrix} T_{1,1} & T_{1,2} \\ T_{2,1} & T_{2,2} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ \rho & \sqrt{1-\rho^2} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{1-\rho^2}} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \rho \\ 0 & 1 \end{bmatrix}, \end{aligned}$$

is J–orthogonal w.r.t. the matrix

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

Let $\hat{G}^{(i)}$ be the generator matrix at the beginning of the ith iteration of the generalized Schur algorithm. The $J_{\hat{p},\hat{q}}$–orthogonal matrix Q_i such that $\tilde{G}^{(i)} = Q_i \hat{G}^{(i)}$ is in proper form, can be constructed in the following way. Let

$$\hat{G}^{(i)} = \left[\begin{array}{c} \hat{G}^{(i,+)} \\ \hline \hat{G}^{(i,-)} \end{array} \right]$$

where

$$\hat{G}^{(i,+)} = \begin{bmatrix} g_1^{(i,+)T} \\ \vdots \\ g_{\hat{p}}^{(i,+)T} \end{bmatrix} \text{ and } \hat{G}^{(i,-)} = \begin{bmatrix} g_1^{(i,-)T} \\ \vdots \\ g_{\hat{q}}^{(i,-)T} \end{bmatrix}.$$

Let $Q_{i,1}$ and $Q_{i,2}$ be two orthogonal matrices of order $\hat{p}$ and $\hat{q}$, respectively, such that $\tilde{G}^{(i,+)} = Q_{i,1}\hat{G}^{(i,+)}$ and $\tilde{G}^{(i,-)} = Q_{i,2}\hat{G}^{(i,-)}$ are both in proper form and let $\hat{Q}_i = Q_{i,1} \oplus Q_{i,2}$. $Q_{i,1}$ and $Q_{i,2}$ can be either Householder matrices or products of Givens rotations chosen in order to annihilate all the entries of the ith column except for the first one of the matrices $\hat{G}^{(i,+)}$ and $\hat{G}^{(i,-)}$, respectively.

Let $v \in \mathbb{R}^2$ whose components are the nonzero entries in the ith column of $\tilde{G}^{(i,+)}$ and $\tilde{G}^{(i,-)}$, respectively. Consider the J–orthogonal matrix $\hat{T}$ such that the second component of the vector $\hat{v} = \hat{T}v$ is 0. Let

$$\tilde{T}_i = \begin{bmatrix} \hat{T}_{1,1} & 0 & \hat{T}_{1,2} & 0 \\ 0 & I_{\hat{p}-1} & 0 & 0 \\ \hat{T}_{2,1} & 0 & \hat{T}_{2,2} & 0 \\ 0 & 0 & 0 & I_{\hat{q}-1} \end{bmatrix},$$

where $I_{\hat{p}-1}$ and $I_{\hat{q}-1}$ are identity matrices of order $\hat{p}-1$ and $\hat{q}-1$, respectively, and the 0's in the matrix $\tilde{T}_i$ are null matrices of appropriate dimension. It is easy to see that $\tilde{T}_i$ is a $J_{\hat{p},\hat{q}}$–orthogonal matrix. The $J_{\hat{p},\hat{q}}$–orthogonal matrix $\hat{Q}_i\tilde{T}_i$ is such that $\hat{Q}_i\tilde{T}_i\hat{G}^{(i)}$ is in proper form. Hence $Q_i = \hat{Q}_i\tilde{T}_i$ is the sought matrix.

FAST COMPUTATION OF THE R FACTOR OF THE QR FACTORIZATION OF H

In this section the computation of the R factor of the QR factorization of H, defined in §1, exploiting the displacement structure of H^TH is described. We first show that the displacement rank of H^TH is $2(m+l+1)$ at most, with the same number of positive and negative generators. Hence the generalized Schur algorithm to compute the R factor requires $O\left(s^2(m+l)^3\right)$ flops. Furthermore the computation of the generators requires $O\left((m+l)N\right)$ flops.

Let

$$W = H^TH = \left[\begin{array}{c|c} U_{1,2s,N}U_{1,2s,N}^T & U_{1,2s,N}Y_{1,2s,N}^T \\ \hline Y_{1,2s,N}U_{1,2s,N}^T & Y_{1,2s,N}Y_{1,2s,N}^T \end{array} \right]$$

Let Z_m and Z_l be shift matrices of order m and l, respectively. Denote

$$Z_{2s,m} = \bigoplus_{i=1}^{2s} Z_m, \; Z_{2s,l} = \bigoplus_{i=1}^{2s} Z_l,$$

and

$$Z = Z_{2s,m} \bigoplus Z_{2s,l}.$$

Partition $U_{1,2s,N}$ and $Y_{1,2s,N}$ in the following way,

$$U_{1,2s,N} = \begin{bmatrix} \hat{u}_1 & \hat{u}_2 \\ \hat{u}_3 & \hat{u}_4 \end{bmatrix}, \quad Y_{1,2s,N} = \begin{bmatrix} \hat{y}_1 & \hat{y}_2 \\ \hat{y}_3 & \hat{y}_4 \end{bmatrix},$$

with $\hat{u}_1 \in \mathbb{R}^{m\times(N-1)}, \hat{u}_2 \in \mathbb{R}^{m\times 1}, \hat{u}_3 \in \mathbb{R}^{(2s-1)m\times(N-1)}, \hat{u}_4 \in \mathbb{R}^{(2s-1)m\times 1}$, and $\hat{y}_1 \in \mathbb{R}^{l\times(N-1)}, \hat{y}_2 \in \mathbb{R}^{l\times 1}, \hat{y}_3 \in \mathbb{R}^{(2s-1)l\times(N-1)}, \hat{y}_4 \in \mathbb{R}^{(2s-1)l\times 1}$. Then

$$\begin{aligned} \nabla_W &= W - ZWZ^T \\ &= \begin{bmatrix} \hat{u}_1\hat{u}_1^T & \hat{u}_1\hat{u}_3^T & \hat{u}_1\hat{y}_1^T & \hat{u}_1\hat{y}_3^T \\ \hat{u}_3\hat{u}_1^T & 0 & \hat{u}_3\hat{y}_1^T & 0 \\ \hat{y}_1\hat{u}_1^T & \hat{y}_1\hat{u}_3^T & \hat{y}_1\hat{y}_1^T & \hat{y}_1\hat{y}_3^T \\ \hat{y}_3\hat{u}_1^T & 0 & \hat{y}_3\hat{y}_1^T & 0 \end{bmatrix} \\ &\quad + g_{m+l+1}^{(+)} {g_{m+l+1}^{(+)}}^T - g_{m+l+1}^{(-)} {g_{m+l+1}^{(-)}}^T \\ &= \hat{W} + g_{m+l+1}^{(+)} {g_{m+l+1}^{(+)}}^T - g_{m+l+1}^{(-)} {g_{m+l+1}^{(-)}}^T \end{aligned}$$

where $g_{m+l+1}^{(+)} = [\hat{u}_2^T \; \hat{u}_4^T \; \hat{y}_2^T \; \hat{y}_4^T]^T, g_{m+l+1}^{(-)} = ZH(1,:)^T$. The symmetric matrix $\hat{W}$ has rank $2(m+l)$ at most. Hence the displacement rank of W is $2(m+l+1)$, *i.e.*,

$$\begin{aligned} \nabla_W = W - ZWZ^T &= \sum_{i=1}^{m+l+1} g_i^{(+)} {g_i^{(+)}}^T \\ &\quad - \sum_{i=1}^{m+l+1} g_i^{(-)} {g_i^{(-)}}^T. \end{aligned}$$

The computation of the remaining generators requires the computation of $\hat{W}$, that is, of the first m rows and the rows $2sm+i$, $i = 1 : l$. This can be accomplished in $O\left((l+m)N\right)$ flops. We observe that the computation of the matrix $\hat{W}$ requires $O(2(s(m+l)^2N)$ flops. Furthermore we point out that the matrix H, involved in the computation of $\hat{W}$, is not explicitly stored. The computation indeed is done considering only the vectors y and u, taking into account the block–Hankel structure of H.

COMPUTATIONAL BURDEN

To compute the generators of $H^T H$, the product

$$H^T H(:, [1:m, 2sm+1:2sm+l]) \qquad (2)$$

is needed. This can be accomplished in $2N(m+l)$ flops. All the techniques described in the appendix require $O(s(m+l))$ flops to compute the generators. If Householder matrices are chosen as the orthogonal matrices involved at each iteration, the total number of flops of the generalized Schur algorithm is $8s^2(m+l)^3 + 12s^2(m+l)^2$. As N is significantly larger than s, m, l the bulk of the algorithm is the computation of (2).

THE GENERALIZED SCHUR ALGORITHM FOR RANK–DEFICIENT MATRICES

In this section we describe how the generalized Schur algorithm can be modified to compute the R factor in case H is a rank–deficient matrix. More details can be found in [5]. We show that the computational complexity is reduced to $O(rs(m+l))$ in this case, where r is the rank of H.

Without loss of generality, we suppose that the first k, $k = 1, \ldots, 2s(m+l)-1$, columns of H are full rank and the $(k+1)$–th column linearly depends on the first k columns, i.e.,

$$H = \left[\begin{array}{c|c|c} H_1 & H_1 w & H_2 \end{array}\right],$$

where $H_1 \in \mathbb{R}^{N\times k}$, $w \in \mathbb{R}^k$, $H_2 \in \mathbb{R}^{N\times 2(m+l)s-k-1}$. Then

$$H^T H = \left[\begin{array}{c|cc} H_1^T H_1 & H_1^T H_1 w & H_1^T H_2 \\ \hline w^T H_1^T H_1 & w^T H_1^T H_1 w & w^T H_1^T H_2 \\ H_2^T H_1 & H_2^T H_1 w & H_2^T H_2 \end{array}\right].$$

The Schur complement of $H^T H$ with respect to $H_1^T H_1$ is

$$(H^T H)^{(k)} = \left[\begin{array}{c|c} 0 & 0^T \\ \hline 0 & H_2^T(I - H_1(H_1^T H_1)^{-1} H_1^T) H_2 \end{array}\right].$$

We observe that the first row (and column) of $(H^T H)^{(k)}$ is a null vector. Let $\tilde{G}^{(k,+)}$ and $\tilde{G}^{(k,-)}$ be the matrix of positive and negative generators of $(H^T H)^{(k)}$, respectively, in proper form. Then the first row of $\tilde{G}^{(k,+)}$ is equal to the first row of $\tilde{G}^{(k,-)}$. Hence these rows can be dropped from the corresponding matrices, reducing the number of generators by 2 and the $(k+1)$th row of the R factor is a null vector.

From a numerical point of view we can say that the described procedure works in an accurate way when it is applied to matrices H such that the gap between the *large* singular values and the *negligible* ones is *"sufficiently"* large. A loss of accuracy in the computed factor R happens when the distribution of the *small* singular values of H shows a uniform and slow decrease. This behaviour of the algorithm is described in the following examples.

Table 1: Numerical results for Example 1

R_M # flops	R_S # flops	b. e. R_M	b. e. R_S	n. r.
31660	6418	1.51×10^{-16}	6.22×10^{-15}	5

Example 1 *Consider the matrix* $H = [U^T | Y^T]$, *with* $Y = U$, *where the first row and the last column of* U *are*

$$[\ 40 \quad 39 \quad 38 \quad \ldots \quad 3 \quad 2 \quad 1 \quad 2 \quad 2 \quad 3\ ],$$

$$[\ 3 \quad 2 \quad 2 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7\ ]^T,$$

respectively. The rank of the matrix H *is* 5.

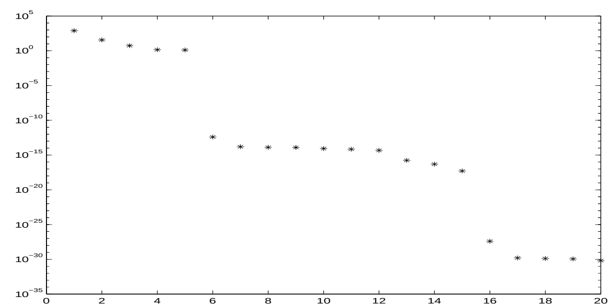

Figure 1: *Distribution of the singular values, in logarithmic scale, of the matrix considered in Example 1*

In Table 1 the results of the computation of the R *factor of the matrix* H *by means of the standard QR and the generalized Schur algorithm are shown. We denote by* R_M, b.e.R_S, b.e.R_* *and* n.r., the R factor of the QR factorization of H computed by the `matlab` function `triu(qr(H))` and by the generalized Schur algorithm, the backward error of $H^T H$ defined as

$$\frac{\|H^T H - R_*^T R_*\|_1}{\|H^T H\|_1},$$

and the rank of H detected by the generalized Schur algorithm, respectively. In this case, the R factor is accurately computed by the generalized Schur algorithm, because of the big difference between the significant singular values and the negligible ones of H (see Figure 1).

Example 2 *This is the fourth application considered in the next section. In Figure 2 we can see that the distribution of the small singular values of the involved matrix* H *slightly decreases. We point out that the correlation matrix* $H^T H$ *computed by* `matlab` *is not numerically*

Table 2: Numerical results for Example 2

R_M # flops	R_S # flops	b. e. R_M	b. e. R_S	n. r.
12492731	3521	1.27×10^{-14}	5.31×10^{-2}	16

s.p.d. because of the nearly rank deficiency of H. So, in this case the fast Cholesky factorization, exploiting the block–Hanckel structure of H and described in [12], can not be used. In Table 2 we can see that, although the generalized Schur algorithm is very fast w.r.t. the standard QR algorithm, the achieved accuracy is not satisfactory.

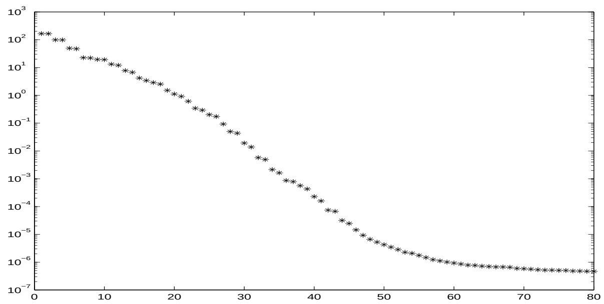

Figure 2: *Distribution of the singular values, in logarithmic scale, of the matrix considered in Example 2*

NUMERICAL RESULTS

In this section results computing the R matrix by means of the generalized Schur algorithm are summarized. The data sets considered are publicly available on the DAISY web site

`www.esat.kuleuven.ac.be/sista/daisy`

For all the considered applications, the generators have been computed using the second technique described in the Appendix. At each iteration of the generalized Schur algorithm, two products of Givens rotations and a modified hyperbolic rotation are performed in order to reduce the generator matrix in proper form. All the numerical results have been obtained on a Sun workstation Ultra 5 using `Matlab 5.3`.

Table 3 gives a summary description of the applications considered in our comparison, indicating the number of inputs m, the number of outputs l, the number of block rows s, the total number of data samples used t and the number of rows of H.

In Table 4 and 5 some results for the computation of the R factor of the QR factorization of H are presented. Rel. residual denotes

$$\frac{\||R_M| - |R_S|\|_1}{\||R_M|\|_1}.$$

Table 3: Summary description of applications.

Appl. #	Application	m	l	s	t	N
1	Glass tubes	2	2	20	1401	1361
2	Labo dryer	1	1	15	1000	970
3	Glass oven	3	6	10	1247	1227
4	Mechanical flutter	1	1	20	1024	960
5	Flexible robot arm	1	1	20	1024	984
6	Evaporator	3	3	10	6305	6285
7	CD player arm	2	2	15	2048	2018
8	Ball and beam	1	1	20	1000	960
9	Wall temperature	2	1	20	6800	1640

Table 4: Comparative results for the computation of the R factor.

Appl. #	Application	R_M # flops	R_S # flops
1	Glass tubes	6.76×10^{7}	2.39×10^{6}
2	Labo dryer	7.01×10^{6}	3.00×10^{5}
3	Glass oven	7.63×10^{7}	9.48×10^{6}
4	Mechanical flutter	1.25×10^{7}	3.21×10^{3}
5	Flexible robot arm	1.25×10^{7}	4.26×10^{5}
6	Evaporator	1.82×10^{8}	9.71×10^{6}
7	CD player arm	5.77×10^{7}	2.33×10^{6}
8	Ball and beam	1.22×10^{7}	4.18×10^{5}
9	Wall temperature	4.67×10^{7}	3.81×10^{6}

Table 5: Comparative results for the computation of the R factor.

Appl. #	Application	backward error R_S	Rel.residual
1	Glass tubes	7.30×10^{-15}	3.64×10^{-13}
2	Labo dryer	6.93×10^{-15}	2.00×10^{-12}
3	Glass oven	4.87×10^{-14}	1.32×10^{-9}
4	Mechanical flutter	* * * * **	* * * * * * *
5	Flexible robot arm	4.36×10^{-15}	1.69×10^{-4}
6	Evaporator	1.89×10^{-14}	9.35×10^{-14}
7	CD player arm	3.72×10^{-14}	1.32×10^{-9}
8	Ball and beam	8.63×10^{-15}	5.83×10^{-13}
9	Wall temperature	5.52×10^{-15}	6.28×10^{-12}

The results in Table 4 and 5 are comparable with those described in [12], where the R factor is obtained considering the Cholesky factorization of the correlation matrix $H^T H$, and exploiting the block-Hankel structure of H. The analysis of the fourth application is described in Example 2 of the previous section. We can not compare, in this case, the elements of the matrices computed by both considered algorithms, since the generalized Schur detects a rank deficiency in the matrix H.

CONCLUSIONS

In this paper the generalized Schur algorithm to compute the R factor of the QR factorization of block–Hankel matrices, arising in some subspace identification problems, is described.

It is shown that the generalized Schur algorithm is significantly faster than the classsical QR factorization. A rank–revealing implementation of the generalized Schur algorithm in case of rank–deficient matrices is also discussed.

Algorithmic details and numerical results have been presented.

ACKNOWLEDGEMENTS

This paper presents research results of the Belgian Programme on Interuniversity Poles of Attraction (IUAP P4-02 and P4-24), initiated by the Belgian State, Prime Minister's Office - Federal Office for Scientific, Technical and Cultural Affairs, of the European Community TMR Programme, Networks, project CHRX-CT97-0160, of the Brite Euram Programme, Thematic Network BRRT-CT97-5040 'Niconet', of the Concerted Research Action (GOA) projects of the Flemish Government MEFISTO-666 (Mathematical Engineering for Information and Communication Systems Technology), of the IDO/99/03 project (K.U.Leuven) "Predictive computer models for medical classification problems using patient data and expert knowledge", of the FWO "Krediet aan navorsers" G.0326.98 and the FWO project G.0200.00.

References

[1] A.W. Bojanczyk, R.P. Brent, P. Van Dooren and F.R. De Hoog, A note on downdating the Cholesky factorization, *SIAM J. Sci. Stat. Comput.*, 1 (1980), pp. 210–220.

[2] Y.M. Cho, G.Xu and T. Kailath, Fast recursive identification of state space models via exploitation of displacement structure, *Automatica*, Vol.30. No.1, (1994) pp. 45–59.

[3] J. Chun, T. Kailath and H. Lev–ari, Fast parallel algorithms for QR and triangular factorization, *SIAM J. Sci. and Stat. Comp.*, Vol. 8, No. 6 (1987), pp. 899–913.

[4] B. De Moor, P. Van Overschee and W. Favoreel, Numerical Algorithms for Subspace State-Space System Identification – An Overview, in *Applied and Computational Control, Signals and Circuits*, Vol. 1, Chapter 6, Birkhäuser, Boston, 1999.

[5] K.A. Gallivan, S. Thirumalai, P. Van Dooren and V. Vermaut, High performance algorithms for Toeplitz and block Toeplitz matrices, *Linear Algebra Appl.*, 241-243,(1996) 343-388.

[6] T. Kailath, S. Kung and M. Morf, Displacement ranks of matrices and linear equations, *J. Math. Anal. Appl.*, 68, (1979), pp. 395–407.

[7] T. Kailath and A.H. Sayed, Displacement structure: theory and applications, *SIAM Review*, 37, (1995), pp. 297–386.

[8] T. Kailath, Displacement structure and array algorithms, in *Fast Reliable Algorithms for Matrices with Structure*, T. Kailath and A. H. Sayed, Ed., SIAM, Philadelphia, 1999.

[9] M. Moonen, B. De Moor, L. Vandenberghe and J. Vandewalle, On- and off-line identification of linear state space models, *Int. J. Control*, 49(1), (1989) pp. 219-232.

[10] C.M. Rader and A.O. Steihardt, Hyperbolic Householder transforms, *SIAM J. Matrix Anal. Appl.*, 9, (1988), pp. 269–290.

[11] V. Sima, Subspace–based algorithms for multivariable system identification, *Studies in Informatics and Control*, Vol.5, No.4, (1996) pp.335–344.

[12] V. SIMA, Cholesky or QR factorization for data compression in subspace–based identification?, *Proceedings Second NICONET Workshop*, Paris–Versailles, Dec. 3, (1999), pp.75–80.

[13] M. Stewart and P. Van Dooren, Stability issues in the factorization of structured matrices, *SIAM J. Matrix Anal. Appl.*, Vol. 18, (1997), pp. 104–118.

[14] P. Van Overschee and B. De Moor, *Subspace Identification for Linear Systems: Theory, Implementation, Applications*, Kluwer Academic Publishers, Dordrecht, 1996.

[15] P. Van Overschee and B. De Moor, N4SID: Two subspace algorithms for the identification of combined deterministic-stochastic systems, *Automatica*, 30(1), (1994) pp. 75–93.

[16] M.Verhaegen, Subspace model identification. Part 3:Analysis of the ordinary output–error state–space model identification algorithm, *Int. J. Control*, 58(3) (1993), pp. 555–586.