A fast algorithm for subspace state-space system identification via exploitation of the displacement structure

Nicola Mastronardi [a,d,2], Daniel Kressner [b,3], Vasile Sima [a,e,4], Paul Van Dooren [c,3,5], and Sabine Van Huffel [a,1,2,3]

[a] *Department of Electrical Engineering, ESAT-SISTA/COSIC, Katholieke Universiteit Leuven, Kardinaal Mercierlaan 94, 3001 Heverlee, Belgium*

[b] *Department of Mathematics, University of Chemnitz, Chemnitz, Germany*

[c] *Department of Mathematical Engineering, Université Catholique de Louvain, Avenue Georges Lemaitre 4 B-1348 Louvain-la-Neuve, Belgium*

[d] *Dipartimento di Matematica, Università della Basilicata, via N. Sauro 85, 85100 Potenza, Italy*

[e] *National Institute for Research & Development in Informatics, Bd. Mareşal Averescu, Nr. 8–10, 71316 Bucharest 1, Romania*

Abstract

Two recent approaches [15,16] 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 $H^T H$, 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.

Key words: generalized Schur algorithm, Hankel and block–Hankel matrices, subspace identification, QR decomposition, singular value decomposition.

Preprint submitted to Elsevier Preprint

1 Introduction

Subspace based system identification has become very popular in the last decade [4]. The success of this state–space identification approach is mainly due to the fact that it relies on a simple matrix decomposition for which reliable numerical algorithms are available. Its major drawback, on the other hand, is that large "data" matrices are involved and that this may lead to high computing and storage costs. We now briefly recall the basic formulation of the problem. Let u_k and y_k be the m–dimensional input vector and the l–dimensional output vector, respectively, of the 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, one chooses $N \gg 2(m+l)s$ and constructs the $N \times 2(m+l)s$ matrix $H = [U_{2s,N}^T Y_{2s,N}^T]$, where $U_{2s,N}$ and $Y_{2s,N}$ are block–

[1] This author is a Senior Research Associate with the F.W.O. (Fund for Scientific Research-Flanders).
[2] This work is supported by UE Programme "Training and Mobility of Researchers" project (contract ERBFMRXCT970160) entitled "Advanced Signal Processing of Medical Magnetic Resonance Imaging and Spectroscopy".
[3] This work is supported by the Belgian Programme on Interuniversity Poles of Attraction (IUAP-4/2 & 24), initiated by the Belgian State, Prime Minister's Office for Science, and by a Concerted Research Action (GOA) project of the Flemish Community, entitled "Mathematical Engineering for Information and Communications Systems Technology".
[4] This work is supported by the European Community BRITE-EURAM III *Thematic Networks Programme NICONET* (project BRRT–CT97-5040).
[5] This work is partly supported by the National Science Foundation contract CCR-97-96315.

Hankel matrices defined in terms of the input and output data, respectively :

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

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

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 paper is organized as follows. In §2 the generalized Schur algorithm to compute the Cholesky factor of a symmetric positive definite matrix is described and in §3 this algorithm is applied to the matrix H. The rank–deficient case is described in §4 and some numerical experiments are reported in §5.

2 The Schur algorithm for positive semi-definite matrices

2.1 The Cholesky factor and generator of A

We summarize here the key properties of the generalized Schur algorithm to compute the Cholesky factor of a (symmetric) positive semi-definite (psd) matrix, which will be used in the next section. More details can be found in [7,8]. Let A be a psd matrix of order n, then we define its *displacement*

$$\nabla A = A - Z^T A Z,$$

using a *generalized shift* matrix Z of the same dimension. Here we only require the shift matrix Z to be strictly upper triangular (and hence nilpotent) and we specialize later on to a particular choice of Z. We call the rank of ∇A the *displacement rank* α of A and we assume that it is significantly smaller than n. Let the symmetric matrix ∇A have p positive eigenvalues and $q \doteq \alpha - p$ negative eigenvalues then it has a factorization

$$\nabla A \doteq G^T \Sigma G, \quad \Sigma \doteq \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix}, \quad G \doteq \begin{bmatrix} G_p \\ G_q \end{bmatrix}. \tag{1}$$

The matrix G is called the *generator* of A and since Z is nilpotent one can reconstruct A via the formula

$$A = \sum_{i=0}^{n-1} (Z^i)^T G^T \Sigma G Z^i.$$

The generator and intermediate results derived from transformations of the generator, allow to reconstruct the Cholesky factor of the psd matrix A :

$$A = R^T \cdot R, \quad R = \begin{bmatrix} r_{1,1} & r_{1,2} & \dots & r_{n,n} \\ & r_{2,2} & \dots & r_{2,n} \\ & & \ddots & \vdots \\ & & & r_{n,n} \end{bmatrix}.$$

Notice that if A has rank $r < n$ then so will the factor R which will have its last $n - r$ rows equal to zero :

$$A = R^T \cdot R, \quad R = \begin{bmatrix} r_{1,1} & \dots & \dots & r_{n,n} \\ & \ddots & & \vdots \\ & & r_{r,r} \cdot & \cdot r_{r,n} \end{bmatrix}.$$

If the leading $r \times r$ principal submatrix of A is nonsingular then $r_{i,i}, i = 1, \dots, r$ are all nonzero. Otherwise the "profile" of the trapezoidal matrix R indents to the right each time the nullity of the $i \times i$ principal submatrix of A increases. In our application, A is a product of the type $H^T H$ which clearly is a psd matrix. Its Cholesky factor R is, up to a sign matrix $D = \text{diag}(\pm 1, \dots, \pm 1)$, also the R_H factor of the $Q_H R_H$ decomposition of H : $R_H = DR$. Hence both the problem of computing the R_H factor of the QR factorization of H and that of computing the Cholesky factor of $H^T H$ are equivalent. We discuss now the computation of the Cholesky factor R of A starting from the generator of ∇A.

One easily shows that the *generator* G is not unique. We say that the generator $\tilde{G}$ is *proper* if its first column is zero except possibly its leading element. The following theorem holds for proper generators [7].

Theorem 1 *Let*

$$A = \begin{bmatrix} a_{11} & \mathbf{a}_{12} \\ \mathbf{a}_{21} & A_{22} \end{bmatrix}$$

be a positive semi-definite matrix with proper generator

$$\tilde{G} = \left[\begin{array}{cccc} \tilde{g}_{1,1} & \tilde{g}_{1,2} & \cdots & \tilde{g}_{1,n} \\ \hline 0 & \tilde{g}_{2,2} & \cdots & \tilde{g}_{2,n} \\ \vdots & \vdots & & \vdots \\ 0 & \tilde{g}_{\alpha,2} & \cdots & \tilde{g}_{\alpha,n} \end{array}\right] \doteq \left[\begin{array}{c} \tilde{G}_1 \\ \hline \\ \tilde{G}_2 \\ \\ \end{array}\right],$$

then $\tilde{G}_1$ is the first row of the Cholesky factor R of A. Furthermore the generator matrix for the Schur complement $\hat{A} = A - \tilde{G}_1^T \tilde{G}_1$ is given by $\hat{G}$, where

$$\hat{A} = \left[\begin{array}{c|c} 0 & \mathbf{0} \\ \hline & \\ \mathbf{0} & A_{22} - \mathbf{a}_{21} a_{11}^{-1} \mathbf{a}_{12} \\ & \end{array}\right], \quad \hat{G} = \left[\begin{array}{c} \tilde{G}_1 Z \\ \hline \\ \tilde{G}_2 \\ \\ \end{array}\right].$$

■

We observe that the first column of $\hat{G}$ is zero, which needs to be the case since the first column and row of $\hat{A}$ are zero.

The generalized Schur algorithm just consists of a recursive use of this Theorem : via a transformation Θ (defined below) the generator G of the current matrix A is put in proper form $\tilde{G}$. This yields the current row of the Cholesky factor and the generator of the Schur complement is trivially obtained from a shift Z applied to the first row of the generator. We refer to [7] for more details. The complexity of this algorithm is that of the transformation Θ since the shift Z does not imply any operations. In the next section we describe briefly the construction of Θ.

2.2 Reduction of the generator to proper form

The first row of the Cholesky factor R of A is thus obtained from a proper generator G of A. Reducing a non-proper generator of A to a proper generator $\tilde{G}$, is obtained by applying a transformation Θ to the generator G. In order not to change the product $G^T \Sigma G$ it suffices to choose Θ to be Σ-unitary, i.e. :

$$\Theta^T \Sigma \Theta = \Sigma,$$

since then

$$\tilde{G}^T \Sigma \tilde{G} = (\Theta G)^T \Sigma (\Theta G) = G^T \Sigma G.$$

Typically the matrix Θ is constructed as follows :

$$\Theta \doteq \left[\begin{array}{cc|cc} \gamma & & \sigma & \\ & I_{p-1} & & \\ \hline \sigma & & \gamma & \\ & & & I_{q-1} \end{array}\right] \cdot \left[\begin{array}{c|c} H_p & \\ \hline & H_q \end{array}\right]. \tag{2}$$

The blocks H_p and H_q of the second factor are $p \times p$ and $q \times q$ Householder transformations reducing the first column of G as follows

$$\left[\begin{array}{c|c} H_p & \\ \hline & H_q \end{array}\right] \left[\begin{array}{c} G_p \\ \hline G_q \end{array}\right] = \left[\begin{array}{cc} x_{11} & \mathbf{x}_{12} \\ 0 & \\ \vdots & X_{22} \\ 0 & \\ \hline y_{11} & \mathbf{y}_{12} \\ 0 & \\ \vdots & Y_{22} \\ 0 & \end{array}\right]. \tag{3}$$

The first factor only transforms the rows containing x_{11} and y_{11} and eliminates y_{11} provided $\rho \doteq -y_{11}/x_{11}$ is smaller than one in modulus :

$$\begin{bmatrix} \gamma & \sigma \\ \sigma & \gamma \end{bmatrix} \cdot \begin{bmatrix} x_{11} & \mathbf{x}_{12} \\ y_{11} & \mathbf{y}_{12} \end{bmatrix} = \begin{bmatrix} \tilde{x}_{11} & \tilde{\mathbf{x}}_{12} \\ 0 & \tilde{\mathbf{y}}_{12} \end{bmatrix}.$$

This 2×2 transformation is constructed from $\gamma \doteq 1/\sqrt{1-\rho^2}$, $\sigma \doteq \rho/\sqrt{1-\rho^2}$ and is called a *hyperbolic rotation* since it satisfies $\gamma^2 - \sigma^2 = 1$ [1]. Also note that $\tilde{x}_{11} = x_{11}\sqrt{1-\rho^2}$. When it is implemented in factored form :

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

one shows that the generalized Schur algorithm is backward stable and that it has the same complexity as the unfactored implementation [13].

It follows already from (3) that the $(1,1)$ element of the psd matrix A equals $a_{11} = x_{11}^2 - y_{11}^2 \geq 0$. Therefore if $a_{11} \neq 0$ the above transformation can be performed. On the other hand, if $a_{11} = 0$ then the whole row $\mathbf{a}_{12}$ must be zero since otherwise A would not be psd. Since $\mathbf{a}_{12} = x_{11}\mathbf{x}_{12} - y_{11}\mathbf{y}_{12}$ this also implies that $[x_{11},\ \mathbf{x}_{12}\ ] = \pm [y_{11},\ \mathbf{y}_{12}\ ]$ and that both these rows can just be deleted from the generator [5]. In other words, if $a_{11} = 0$ a simplification can be performed to the current generator $\tilde{G}$. The complexity of the reduction of G to proper form $\tilde{G}$ is essentially that of the Householder transformations H_p and H_q which costs $4(p+q)n$ flops. If $r = \text{rank}\ A$ steps are performed, this algorithm thus requires a total of $4r(p+q)n$ flops. We point out that this is an overestimate since the number of nonzero columns n of the generator decreases at each step and that potentially the number of rows $p+q$ may decrease as well.

3 Fast computation of the R factor of the QR factorization of H

We show here how to compute the gererator G of $A = H^T H$ where $H \in \mathbb{R}^{N,2(m+l)s}$ is the block Hankel matrix described in the first section with blocks of sizes $1 \times m$ and $1 \times l$:

$$H = \left[\begin{array}{cccc|cccc} u_1^T & u_2^T & \dots & u_{2s}^T & y_1^T & y_2^T & \dots & y_{2s}^T \\ u_2^T & \iddots & \iddots & \vdots & y_2^T & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & \vdots & \vdots & \iddots & \iddots & \vdots \\ u_N^T & \dots & \dots & u_{N+2s-1}^T & y_N^T & \dots & \dots & y_{N+2s-1}^T \end{array}\right].$$

The shift matrices used in this context are the matrices

$$Z_m \doteq \begin{bmatrix} 0_m & I_m & & \\ & 0_m & \ddots & \\ & & \ddots & I_m \\ & & & 0_m \end{bmatrix}, \ Z_l \doteq \begin{bmatrix} 0_l & I_l & & \\ & 0_l & \ddots & \\ & & \ddots & I_l \\ & & & 0_l \end{bmatrix}, \ Z \doteq Z_m \oplus Z_l. \quad (4)$$

The following theorem then gives a contruction of a generator for A.

Theorem 2 *Given the QR factorization of the first block columns :*

$$\left[\begin{array}{c|c} u_1^T & y_1^T \\ u_2^T & y_2^T \\ \vdots & \vdots \\ u_N^T & y_N^T \end{array}\right] = \begin{bmatrix} q_1^T \\ q_2^T \\ \vdots \\ q_N^T \end{bmatrix} R_1 \quad (5)$$

where R_1 can be assumed upper trapezoidal of row rank $k \leq m+l$ and $q_i \in \mathbb{R}^k$; define the product

$$\left[\begin{array}{ccc|ccc} C_{u,1} & \dots & C_{u,2s} & C_{y,1} & \dots & C_{y,2s} \end{array}\right] = \begin{bmatrix} q_1 & \dots & q_N \end{bmatrix} H. \quad (6)$$

Then a generator G for $H^T H$ is given by

$$G = \left[\begin{array}{c|c} G_u & G_y \end{array}\right], \quad \Sigma = I_{k+1} \oplus -I_{k+1} \quad (7)$$

where

$$G_u = \left[\begin{array}{cccc} C_{u,1} & C_{u,2} & \dots & C_{u,2s} \\ 0 & u_{N+1}^T & \dots & u_{N+2s-1}^T \\ \hline 0 & C_{u,2} & \dots & C_{u,2s} \\ 0 & u_1^T & \dots & u_{2s-1}^T \end{array}\right], \quad G_y = \left[\begin{array}{cccc} C_{y,1} & C_{y,2} & \dots & C_{y,2s} \\ 0 & y_{N+1}^T & \dots & y_{N+2s-1}^T \\ \hline 0 & C_{y,2} & \dots & C_{y,2s} \\ 0 & y_1^T & \dots & y_{2s-1}^T \end{array}\right].$$

■

Proof :
In order to prove the result we consider the displacement matrix $\nabla H^T H$:

$$\left[\begin{array}{c|c} U_{2s,N}U_{2s,N}^T - Z_m^T U_{2s,N}U_{2s,N}^T Z_m & U_{2s,N}Y_{2s,N}^T - Z_m^T U_{2s,N}Y_{2s,N}^T Z_l \\ \hline Y_{2s,N}U_{2s,N}^T - Z_l^T Y_{2s,N}U_{2s,N}^T Z_m & Y_{2s,N}Y_{2s,N}^T - Z_l^T Y_{2s,N}Y_{2s,N}^T Z_l \end{array}\right] \tag{8}$$

which ought to be equal to

$$\left[\begin{array}{c|c} G_u^T \Sigma G_u & G_u^T \Sigma G_y \\ \hline G_y^T \Sigma G_u & G_y^T \Sigma G_y \end{array}\right]. \tag{9}$$

It follows from (5,6) that $R_1 = [C_{u,1} C_{y,1}]$ and hence

$$[C_{u,1} C_{y,1}]^T \left[\begin{array}{c|c} C_{u,1} \dots C_{u,2s} & C_{y,1} \dots C_{y,2s} \end{array}\right] = R_1^T \left[\begin{array}{ccc} q_1 & \dots & q_N \end{array}\right] H,$$

which are the first block rows of the sub-blocks of (8). This thus verifies the first block rows and block columns of the equality between (8) and (9). The rest easily follows from the block Hankel structure of H. ■

Note that if the first block columns of H in (5) have full rank then R_1 is square invertible and $k = m + l$. If moreover the whole matrix H has full column rank, then the generalized Schur algorithm will not encounter any singularities. But since the low rank case is of particular interest here, singularities in the generalized Schur algorithm will be encountered and hence lead to a lower complexity of the algorithm.

The above theorem also shows that the displacement rank of $H^T H$ is at most $2(k + 1) \leq 2(m + l + 1)$, with the same number of positive and negative generators. Hence the generalized Schur algorithm to compute the R factor requires about $(8Nrk)$ flops. To compute the generator G of $H^T H$, the QR

Table 1
Numerical results for Example 1

R_M # flops	R_S # flops	backward error R_M	backward error R_S	numerical rank
31660	9489	1.51×10^{-16}	5.20×10^{-15}	5

factorization (5) requires $(6N(m+l)^2)$ flops and the product (6) requires less than $(4Nk(m+l)s)$ flops. We recall that $k \leq (m+l)$ and $r \leq 2(m+l)s$ but that equality is obtained when no rank deficiency is detected. The most time consuming steps are then clearly the generalized Schur algorithm and the product (6).

4 The generalized Schur algorithm for rank–deficient matrices

Our description of the generalized Schur algorithm allows to handle rank deficient matrices $H^T H$. In this case we can drop some rows of the generator during the algorithm. For this we need a tolerance, say $\delta \doteq \eta \|H^T H\|$ where η is the requested relative accuracy. Referring to the description of section 2, we test if $x_{11}^2 - y_{11}^2 \leq \delta$. We then check as well if the leading row $\mathbf{a}_{12}$ of the current Schur complement is small. If so, the currently computed row of the Cholesky factor is neglectable and we delete the two corresponding rows of the generator. It is possible that $\mathbf{a}_{12}$ is much larger than δ although $a_{11} \leq \delta$. In this case the deletion of a row of the Cholesky factor will yield residual errors $\|H^T H - R^T R\|$ of the same size. This is analyzed in this section. From the first example we can conclude that the described procedure works accurately when it is applied to a matrix H with a *sufficiently large* gap between *significant* singular values and *negligible* ones. On the other hand, a loss of accuracy in the computed factor R is observed when the distribution of the *small* singular values of H shows a uniform and slow decrease. The relative accuracy η is chosen equal to 10^{-13} in both examples.

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

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

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

respectively. The rank of the matrix H is 5 and $\|H^T H\|_1 = 6.31 \times 10^5$.

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, R_S, backward error R_*, and numerical rank, the R factor of

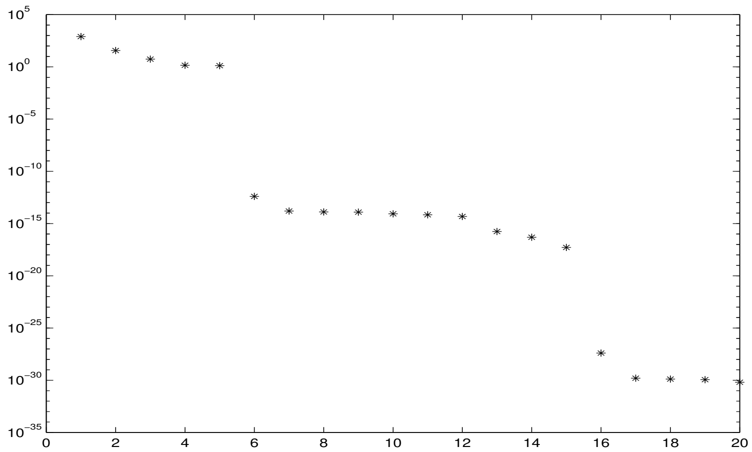

Fig. 1. *Distribution of the singular values, in logarithmic scale, of the matrix considered in Example 1*

Table 2
Numerical results for Example 2

R_M # flops	R_S # flops	backward error R_M	backward error R_S	numerical rank
12.49×10^6	40.61×10^4	1.27×10^{-14}	2.92×10^{-2}	18

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 s.p.d. because of the nearly rank deficiency of H. Furthermore $\|H^T H\|_1 = 3.99 \times 10^4$. 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.*

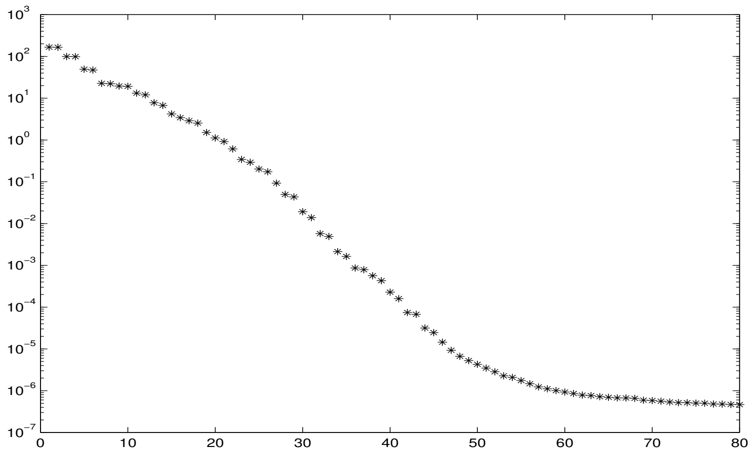

Fig. 2. *Distribution of the singular values, in logarithmic scale, of the matrix considered in Example 2*

5 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

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

At each iteration of the generalized Schur algorithm, two Householder matrices and one modified hyperbolic rotation are computed in order to reduce the generator 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 some results for the computation of the R factor of the QR factorization of H are presented. Rel. residual denotes

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

The results in Table 4 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.

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	backward error R_S	Rel.residual
1	Glass tubes	6.76×10^7	2.61×10^6	2.20×10^{-15}	8.73×10^{-14}
2	Labo dryer	7.01×10^6	3.36×10^5	8.30×10^{-15}	9.48×10^{-13}
3	Glass oven	7.63×10^7	6.38×10^6	3.73×10^{-15}	7.91×10^{-12}
4	Mechanical flutter	1.25×10^7	4.89×10^3	2.92×10^{-2}	0.39×10^0
5	Flexible robot arm	1.25×10^7	4.76×10^5	4.13×10^{-15}	3.38×10^{-5}
6	Evaporator	1.82×10^8	1.11×10^7	6.26×10^{-15}	5.13×10^{-14}
7	CD player arm	5.77×10^7	2.59×10^6	5.01×10^{-15}	2.08×10^{-8}
8	Ball and beam	1.22×10^7	4.67×10^5	7.59×10^{-15}	6.79×10^{-13}
9	Wall temperature	4.67×10^7	1.64×10^6	2.45×10^{-14}	3.76×10^{-12}

6 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.

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), pp. 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 (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.