RECURSIVE MOESP TYPE SUBSPACE IDENTIFICATION ALGORITHM

Catarina J. M. Delgado
Faculdade de Economia do Porto
Rua Dr. Roberto Frias, 4200-464 Porto - Portugal

P. Lopes dos Santos
Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, 4200-465 Porto - Portugal

Keywords: System Identification, Subspace methods, State-space models, Linear Systems, Parameter estimation.

Abstract: In this paper two recursive algorithms, based on MOESP type subspace identification, are presented in two versions. The main idea was to show that we can represent subspace identification methods as sequences of least squares problems and implement them through sequences of modified Householder algorithms. Therefore, it is possible to develop iterative algorithms with most of the advantages of this kind of methods, and still improve the numerical efficiency, in order to deal with real-time applications and minimize the computational burden.

1 INTRODUCTION

The Subspace Identification algorithms aim to estimate, from measured input / output data sequences \( \{u_k\} \) and \( \{y_k\} \), respectively, the system described by:

\[
\begin{align*}
x_{k+1} &= A x_k + B u_k + K e_k \\
y_k &= C x_k + D u_k + e_k \\
E[e_p e_q^T] &= R_e \delta_{pq} \geq 0
\end{align*}
\]

where \( A \in \mathbb{R}^{n \times n} \), \( B \in \mathbb{R}^{n \times m} \), \( C \in \mathbb{R}^{l \times n} \), \( D \in \mathbb{R}^{l \times l} \), \( K \in \mathbb{R}^{n \times l} \) and \( x_k \in \mathbb{R}^n \). The sequence \( \{e_k\} \in \mathbb{R}^l \) is a white noise stochastic process and the input data sequence is assumed to be a persistently exciting quasi-stationary deterministic sequence (Ljung, 1987).

Most of the past contributions ((Larimore, 1990), (Moonen et al., 1989), (van Overschee and de Moor, 1996), (Verhaegen, 1994), (Viberg et al., 1991)) are centered in the off-line identification of time-invariant systems field, by developing non-recursive algorithms. Even though the on-line identification can be applied to a wider class of systems (time-variant, non-linear...), it has been a little forgotten due to its high computation and storage costs. When a particular subspace off-line identification algorithm (such as the algorithm introduced by Van Overschee and De Moor (van Overschee and de Moor, 1996)) is adapted to a recursive identification algorithm ((Cho et al., 1994), (Ljung, 1983), (Gustafsson et al., 1998), (Oku and Kimura, 2000)), not only the computational burden is reduced, but also the application scope is enlarged, preserving most of the advantages of this kind of methods.

This paper is organized as follows: in section 2, the notation is introduced and, in section 3, two basic MOESP-type algorithms are described: the PO-MOESP (Verhaegen, 1994) and the R-MOESP (van Overschee and de Moor, 1996). Sections 4 and 5 are dedicated to the main results: section 4 introduces the 2QR recursive version for the MOESP algorithms and section 5 the 1QR version. Finally, in section 6, some simulation results are shown and, in section 7, the conclusions are presented.

2 NOTATION

We define the following block Hankel matrices, built with input data:

\[
U_p = \begin{bmatrix} U_{(0)} & \ldots & U_{(i-1)} \\ U_{(i)} & \ldots & U_{(i)} \end{bmatrix},
U_{p+} = \begin{bmatrix} U_{(0)} \\ \vdots \\ U_{(i)} \end{bmatrix},
U_f = \begin{bmatrix} U_{(i)} \\ \ldots \\ U_{(2i-1)} \end{bmatrix},
U_{f-} = \begin{bmatrix} U_{(i)} \\ \vdots \\ U_{(2i)} \end{bmatrix}
\]

where \( U_{(i)} = [u_i \ldots u_{i+j-1}] \in \mathbb{R}^{m \times j} \) and the subscripts \( p \) and \( f \) denote past and future data, respec-
When the measurements are noise corrupted, it may not be straightforward to distinguish the "nonzero" singular values from the "zero" singular values (S_n^+). Therefore, a better estimate of the system order is obtained by trying different values and then comparing the simulation errors (van Overschee and de Moor, 1996).

In the second main step, the system matrices are estimated. There are two approaches to estimate those matrices: the method introduced by Van Overschee, with the R-MOESP algorithm (van Overschee and de Moor, 1996), and the method introduced by Verhaegen, with the PO-MOESP algorithm (Verhaegen, 1994).

### 3.1 R-MOESP Algorithm.

When we know an estimate of Π_i, matrices A and C are obtained by solving the following linear equation, in a least squares sense (van Overschee and de Moor, 1996):

\[
\begin{bmatrix}
\Gamma_i+ Z_{i+1} \\
Y_{(i)}
\end{bmatrix} = \begin{bmatrix}
A \\
C
\end{bmatrix} K_{BD} 
\begin{bmatrix}
\Gamma_i \\
U_f
\end{bmatrix} + \rho \tag{4}
\]

where \(\Gamma_i = \begin{bmatrix} I_{(i-1)} & 0 \end{bmatrix} \Gamma_i \). The matrix \(K_{BD} \) is then used to estimate \(B \) and \(D \), since

\[
K_{BD} = \begin{bmatrix}
K_A \\
K_C
\end{bmatrix} = \begin{bmatrix}
B & \Gamma_i+ H_d^{i-1} \\
D & 0
\end{bmatrix} = \begin{bmatrix}
A & C \end{bmatrix} \Gamma_i H_d^{i-1} = \begin{bmatrix}
N_C_1 & D \\
B & 0 \\
B & B
\end{bmatrix}
\]

Lopes dos Santos and Martins de Carvalho (dos Santos and de Carvalho, 2003) have shown that \(K_A \) can be written as \(K_A = K_{PB} K_C = \begin{bmatrix} -A (\Gamma_i^T \Gamma_i)^{-1} C^T \end{bmatrix} K_C \). Therefore, we can work only with:

\[
K_{C(B,D)} = \begin{bmatrix}
K_{PB} \\
K_B
\end{bmatrix} = \begin{bmatrix}
K_{C1} & D \\
B & 0 \\
B & B
\end{bmatrix}
\]

where

\[
K_{C1} = \begin{bmatrix}
I_k - L_{C1} & -L_{C2} & \ldots & -L_{C1} \\
\vdots & \ddots & \ddots & \vdots \\
0 & \ldots & -L_{C1} & 0
\end{bmatrix} \begin{bmatrix}
G_1 \\
G_{Ck} \\
\vdots \\
G_{i-1}
\end{bmatrix}
\]

Equation (6) can be rewritten as:

\[
\begin{bmatrix}
K_{C1} \\
K_{C1}
\end{bmatrix} = \begin{bmatrix}
N_{C1} \\
N_{C1}
\end{bmatrix} \begin{bmatrix}
D \\
B
\end{bmatrix} \tag{7}
\]

and B and D estimated in the least squares sense.

### 3.2 PO-MOESP Algorithm

Another approach to determine matrices A and C is to use the shift-invariance structure in \(\Gamma_i \) (Verhaegen, 1994):

\[
\hat{C} = \begin{bmatrix}
I_k & 0
\end{bmatrix} \hat{\Gamma}_i \tag{8}
\]

\[
\hat{A} = \Gamma_{i-1}^+ \hat{\Gamma}_i \tag{9}
\]
where $\Gamma_i = [0 \quad I_{n-i+1}] \hat{\Gamma}$. These estimates of A and C are the same as the estimates produced by the method of Van Overschee and De Moor ((Chiuso and Santos and de Carvalho, 2004), (Delgado and dos Santos, 2004a), (dos Santos and de Carvalho, 2004)).

Then, matrices B and D can be estimated by linear regression:

$$vec(\hat{y}_k) = [M_D \quad M_B] \begin{bmatrix} \theta_D \\ \theta_B \end{bmatrix}$$

(10)

where

$$M_D = (u_k^T \otimes I_t)$$

(11)

$$M_B = \sum_{t=1}^{k-1} (u_t^T \otimes CA^{k-t-1})$$

(12)

$$\theta_D = vec(D)$$

(13)

$$\theta_B = vec(B)$$

(14)

and $\hat{y}_k$ is the simulated output, given by

$$\hat{y}_k = D u_k + \sum_{t=1}^{k-1} CA^{k-t-1} B u_t = \hat{y}_D k + \hat{y}_B k$$

(15)

with

$$vec(\hat{y}_D k) = vec(D u_k) = (u_k^T \otimes I_t)vec(D)$$

(16)

$$vec(\hat{y}_B k) = vec(\sum_{t=1}^{k-1} CA^{k-t-1} B u_t)$$

(17)

and $L_{H2}$ are two lower triangular matrices and matrix $Q_L$ is orthonormal and

$$Q_L Q_L^T = I_j \Leftrightarrow Q_{Lp} Q_{Lp}^T = \begin{cases} I, & \text{if } p = q \\ 0, & \text{if } p \neq q \end{cases}$$

(19)

$$Q_L^T Q_L = \sum_{k=1}^{3} Q_{Lk} Q_{Lk} = I_j$$

(20)

From this decomposition,

$$\Pi_{U_f} = U_f^T (U_f U_f^T)^{-1} U_f = Q_{L1} Q_{L1}^T$$

(21)

$$\Pi_{H_U} = \begin{bmatrix} Q_{L1} & Q_{L2} \\ Q_{L1} & Q_{L3} \end{bmatrix}$$

(22)

$$\Pi_{U_f^+} = I_j - \Pi_{U_f} = \begin{bmatrix} Q_{L2}^T \\ Q_{L3} \end{bmatrix} \begin{bmatrix} Q_{L1} \\ Q_{L3} \end{bmatrix}$$

(23)

$$\Pi_{H_{U^+}} = I_j - \Pi_{H_L} = Q_{L3} Q_{L3}$$

(24)

If we consider the least squares problem

$$Y_f = [\theta_{U_f} \quad \theta_{H_p}] \begin{bmatrix} U_f \\ H_p \end{bmatrix}$$

(25)

we can rewrite it as

$$B_{Q_L} = [\theta_{U_f} \quad \theta_{H_p}] \begin{bmatrix} L_{U1} & 0 & 0 \\ L_{H1} & L_{H2} & 0 \end{bmatrix}$$

(26)

where $B_{QL} = \begin{bmatrix} B_U \\ B_H \end{bmatrix}$. Then,

$$Z_i = \Pi_{U_f}$$

(27)

As to the weighted projection, it can be expressed as

$$Z_i \Pi_{U_f} = \begin{bmatrix} \theta_{H_p} \end{bmatrix} \begin{bmatrix} U_f \\ H_p \end{bmatrix} = \begin{bmatrix} \theta_{U_f} \quad \theta_{H_p} \end{bmatrix}$$

(28)

In a recursive approach, the dimension of the matrices involved must not depend on the number of measurements $(N = j - 2i + 1)$. Since $Q_{L2} \in \mathbb{R}^{(m+i)(m+i)}$, this could be a problem in the SVD step. However,

$$(B_{H} Q_{L2}) (B_{H} Q_{L2})^T = B_{H} B_{H}^T$$

(29)

and, if $B_{H} Q_{L2} = U S V^T$,

$$(B_{H} Q_{L2})^T = U S^2 V^T = B_{H} B_{H}^T$$

(30)

This means that $B_{H}$ has the same left singular vectors and the same singular values as $Z_i \Pi_{U_f}$. Since we only need $U_n$ and $S_n$ to determine a basis for the column space of $\Gamma_i$, we only have to deal with $B_{H} \in \mathbb{R}^{l \times (m+i)}$, in order to estimate $\Gamma_i$.

4 RECURSIVE 2QR APPROACH

4.1 The weighted projection

The estimation of $\Gamma_i$ from the singular value decomposition of $Z_i \Pi_{U_f}$ is implemented through the LQ decomposition of $[U_f \ H_p]$:

$$\begin{bmatrix} U_f \\ H_p \end{bmatrix} = L Q_L = \begin{bmatrix} L_U \\ L_{H1} \quad L_{H2} \end{bmatrix} \begin{bmatrix} Q_{L1} \\ Q_{L2} \quad Q_{L3} \end{bmatrix}$$

(18)

where $L_{U1} \in \mathbb{R}^{m \times m}$, $L_{H1} \in \mathbb{R}^{(m+i) \times m}$, $L_{H2} \in \mathbb{R}^{(m+i) \times (m+i)}$ and $Q_{L1} \in \mathbb{R}^{m \times (m+i)}$, $Q_{L2} \in \mathbb{R}^{(m+i) \times (m+i)}$, $Q_{L3} \in \mathbb{R}^{(m+i) \times (m+i)}$. Matrices $L_{U1}$
4.2 The system matrices

To estimate matrices $A, B, C$ and $D$, one must solve two least squares problems: (4) and (7). In the last one, both matrices $K_C$ and $N_C$ have constant dimensions, but in the first one that does not happen. Therefore, we will implement (4) recursively, using the $LQ$ decomposition, to avoid this dimension problem, since the least squares solution of

$$\begin{bmatrix} \Gamma_i & Z_{i+1} \end{bmatrix}^T Y_{(i)} = \begin{bmatrix} \Gamma_i^+ Z_i \end{bmatrix} U_f \theta_{ACK}$$

with

$$\theta_{ACK} = \begin{bmatrix} A & KA \\ C & KC \end{bmatrix}$$

is the same as the least squares solution of

$$B_{(N)} = Q_{(N)}^T \begin{bmatrix} \Gamma_i^+ Z_{i+1} \\ Y_{(i)} \end{bmatrix} = \begin{bmatrix} R_{(N)} \\ 0 \end{bmatrix} \theta_{ACK}$$

To update $R_{(N)}$ and $B_{(N)}$ we need the last columns of $Y_{(i)}, U_f, Z_i$ and $Z_{i+1}$. The last column of $Z_i$ is computed directly from (27) and the last column of $Z_{i+1}$ is computed from

$$Z_{i+1} = Y_{(i)} - \Pi_{H_f + U_f} - \begin{bmatrix} B_{U-} & B_{H+} \end{bmatrix}^T \begin{bmatrix} Q_{z_1(i+1)} \\ Q_{z_2(i+1)} \end{bmatrix}$$

where

$$\begin{bmatrix} U_f^- \\ H_p^+ \end{bmatrix} = L_{(i+1)} \begin{bmatrix} Q_{z_1(i+1)} \\ Q_{z_2(i+1)} \end{bmatrix}$$

$$B_{U-} = Y_{(i)} - Q_{z_1(i+1)}$$

$$B_{H+} = Y_{(i)} - Q_{z_2(i+1)}$$

However a problem may here arise, since the rowspaces considered at the $k$-th iteration are not quite the same as in the $(k+1)$-th iteration. In order to deal with this we have implemented two options: recursive least squares with a forgetting factor or the sliding window technique.

4.3 The modified Householder algorithm

Since our versions of subspace algorithms are sequences of least squares problems implemented through the $QR$ decomposition (or the similar $LQ$), we modified the Householder algorithm for matrix factorization updating, to update both $B$ and $R$, at once. In fact, we apply the same Householder reflection to change both $R_{(N)}$ and $B_{(N)}$. As $R_{(N)}$ is already a upper triangular matrix, the QR decomposition of $R_{(N+1)}$ is faster to compute, because now there is only one line to set to zero. Therefore, the Householder Algorithm can be adapted in order to modify only two rows of the matrix in each iteration $k$ (the last row and the $k$-th one), since the others remain the same. Another version of this modified algorithm considers the update of the decomposition matrices when $N_r > 1$ rows are added.

4.4 Recursive R-MOESP-2QR algorithm

1. LQ decomposition and the orthogonal projections

(a) Update the last columns of the block Hankel matrices $U$ and $Y$ $(\phi U_{(N+1)}$ and $\phi Y_{(N+1)}$).

(b) Update the decompositions (18) and (34). Although $L_{(N+1)}$ and $B_{(N+1)}$ have one more line, after the $LQ$ decomposition, this last column of $L_{(N+1)}$ has only zero elements and it can be removed. So, the matrices $L_{(N+1)}$ and $B_{(N+1)}$ still have the dimensions of $L_{(N)}$ and $B_{(N)}$.

(c) Determine $\hat{z}_i$ and $z_{i+1}$, the last columns of $Z_i$ and $Z_{i+1}$, as in (27) and (33).

2. The column space of $\Gamma_i$ and $\Gamma_{i-1}$

(a) Determine $\hat{\Gamma}_i = U_n S_{n}/2$, where the system order $n$ is assumed to be known (3) and

$$B_{H_{(N+1)}} = U_S V_T = U_n S_n V_T$$

(b) Compute $\hat{\Gamma}_{i-1} = \begin{bmatrix} I_{(i-1)} & 0 \end{bmatrix} \hat{\Gamma}_i$

3. Matrices $A, C$ and $K_{BD}$

(a) Update the QR decomposition matrices $R_{AC(N)}$ and $B_{AC(N)}$:

$$\begin{bmatrix} R_{AC(N)} \\ r_{N+1}^T \end{bmatrix} = Q_{AC(N)} \begin{bmatrix} R_{AC(N+1)} \\ 0 \end{bmatrix}$$

$$B_{AC(N+1)} = Q_{AC(N)}^T \begin{bmatrix} B_{AC(N)} \\ b_{N+1}^T \end{bmatrix}$$

where

$$r_{N+1} = \begin{bmatrix} \hat{\Gamma}_{N+1}^+ z_{(N+1)} \\ 0 \end{bmatrix}$$

$$b_{N+1} = \begin{bmatrix} \hat{\Gamma}_{N+1}^+ z_{(N+1)} \\ 0 \end{bmatrix}$$

(b) Solve, in the least squares sense, the problem $B_{AC(N+1)} = R_{AC(N+1)} \theta_{AC(N+1)}$, where

$$\theta_{AC(N+1)} = \begin{bmatrix} A_{(N+1)}^T \\ C_{(N+1)}^T \\ K_{A(N+1)}^T \\ K_{C(N+1)}^T \end{bmatrix}$$
4. Matrices B and D
(a) Determine $K_{C(N+1)}$ from $K_{(N+1)}$ and matrix $N_C$, as in (6)
(b) Solve the least squares problem (7) recursively
$$
\begin{bmatrix}
B_{K(k-1)} \\
K_{Ck}
\end{bmatrix} = \begin{bmatrix}
R_{N(k-1)} \\
N_{Ck}
\end{bmatrix} \begin{bmatrix}
D_{(k)} \\
B_{(k)}
\end{bmatrix}
$$

4.5 Recursive PO-MOESP-2QR algorithm

1. LQ decomposition and the orthogonal projections
(a) Update the last columns of the block Hankel matrices $U$ and $Y$ ($\phi_{U(N+1)}$ and $\phi_{Y(N+1)}$)
(b) Update the decompositions (18) and (34).

2. The column space of $\Gamma_1, \Gamma_i$ and $\Gamma_{i-1}$
(a) Determine $\hat{\Gamma}_i = U_n S_n^{1/2}$, where the system order $n$ is assumed to be known (3) and
$$
B_{iH_{N+1}} = USV^T = U_n S_n V_n^T
$$
(b) Compute $\hat{\Gamma}_{i-1} = \begin{bmatrix} I_{l(i-1)} & 0 \end{bmatrix} \hat{\Gamma}_i$
(c) Compute $\hat{\Gamma}_i = \begin{bmatrix} 0 & I_{l(i-1)} \end{bmatrix} \hat{\Gamma}_i$

3. Matrices $A$ and $C$
(a) Determine $\hat{A} = \hat{\Gamma}_{i-1} \hat{\Gamma}_i$
(b) Determine $\hat{C} = [ I_l \ 0 ] \hat{\Gamma}_i$

4. Matrices $B$ and $D$ ("minimum simulation error method")
(a) Update the QR decomposition
$$
\begin{bmatrix}
B_{BD(N)} \\
Y_{BD}
\end{bmatrix} = \begin{bmatrix}
R_{BD(N)} \\
X_D \ X_B
\end{bmatrix} \begin{bmatrix}
\theta_D \\
\theta_B
\end{bmatrix}
$$
where $B_{BD(N)} \in \mathbb{R}^{m(n+l) \times 1}, R_{BD(N)} \in \mathbb{R}^{m(n+l) \times m(n+l)}$ and
$$
\begin{align*}
X_B &= \sum_{t=1}^{k-1} u_t^T \odot (CA^{k-1-t}) \in \mathbb{R}^{l \times mn} \\
X_D &= u_k^T \odot I_l \in \mathbb{R}^{l \times lm} \\
Y_{BD} &= y_k \in \mathbb{R}^{l \times 1} \\
\theta_B &= vec(B) \in \mathbb{R}^{mn \times 1} \\
\theta_D &= vec(D) \in \mathbb{R}^{md \times 1}
\end{align*}
$$
(b) Determine $vec(B)$ and $vec(D)$, by solving, through a back substitution process,
$$
B_{BD(N+1)} = R_{BD(N+1)} \begin{bmatrix}
\theta_D \\
\theta_B
\end{bmatrix}
$$
(c) Recover matrices $B$ and $D$ from $vec(B)$ and $vec(D)$, respectively.

5 RECURSIVE 1QR APPROACH

In this approach, $Z_i, Z_{i+1}$ and $Z_i \Pi_{U_f +}$ are obtained from a unique LQ decomposition:
$$
\begin{bmatrix}
U_f \\
H_p \\
Y_{(i)}
\end{bmatrix} = \begin{bmatrix}
L_U & 0 & 0 & 0 \\
L_{H1} & L_{H2} & 0 & 0 \\
L_{Y1} & L_{Y2} & L_{Y3} & 0
\end{bmatrix} \begin{bmatrix}
Q_{L1} \\
Q_{L2} \\
Q_{L3} \\
Q_{L4}
\end{bmatrix}
$$

In fact,
$$
\begin{align*}
\Pi_{U_f} &= U_f^T (U_f U_f^T)^{-1} U_f = Q_{L1} Q_{L1}^T & (39) \\
\Pi_{HU} &= [ Q_{L1}^T \ Q_{L2}^T ] \begin{bmatrix}
Q_{L1} \\
Q_{L2}
\end{bmatrix} & (40)
\end{align*}
$$
and, since the column space of $[ H_{p+1}^T \ U_f^T ]$ is spanned by the columns of $[ H_{p+1}^T \ U_f^T \ Y_{(i)}^T ]$,
$$
\begin{align*}
\Pi_{H+U-} &= [ Q_{L1}^T \ Q_{L2}^T \ Q_{L3}^T ] \begin{bmatrix}
Q_{L1} \\
Q_{L2} \\
Q_{L3}
\end{bmatrix} & (41)
\end{align*}
$$
Therefore, if we define
$$
Y_f Q_L = [ B_1 \ B_2 \ B_3 \ B_4 ] =
$$
$$
= [ Y_f Q_L ] = [ b_1 \ b_2 \ b_3 \ b_4 ]
$$
then
$$
Z_i = [ B_1 \ B_2 ] \begin{bmatrix}
Q_{L1} \\
Q_{L2}
\end{bmatrix} & (43)
$$
$$
Z_{i+1} = [ B_{m1} \ B_{m2} \ B_{m3} ] \begin{bmatrix}
Q_{L1} \\
Q_{L2} \\
Q_{L3}
\end{bmatrix} & (44)
$$
and $Z_i \Pi_{U_f +} = B_2 Q_{L2}$.

5.1 Recursive R-MOESP-1QR algorithm

1. LQ decomposition and the orthogonal projections
(a) Update the last columns of the block Hankel matrices $U$ and $Y$ ($\phi_{U(N+1)}$ and $\phi_{Y(N+1)}$)
(b) Update the decomposition (38)
(c) Determine $\hat{Z}_i$ and $\hat{Z}_{i+1}$, the last columns of $Z_i$ and $Z_{i+1}$, as in (43) and (44).

2. Determine $\hat{\Gamma}_i = U_n S_n^{1/2}$, where
$$
B_{2(N+1)} = U_n S_n V_n^T
$$
and the system order $n$ is assumed to be known (3).
Compute $\hat{\Gamma}_{i-1} = [ I_{l(i-1)} \ 0 ] \hat{\Gamma}_i$
3. and 4. Implemented as in the algorithm described in section (4.4)

180
5.2 Recursive PO-MOESP-1QR algorithm

1. LQ decomposition and the orthogonal projections
   (a) Update the last columns of the block Hankel matrices $U$ and $Y$ ($\phi_U(N+1)$ and $\phi_Y(N+1)$).
   (b) Update the decomposition (38)
2. The column space of $\Gamma_i$, $\Gamma_{i-1}$
   (a) Determine $\hat{\Gamma}_i = U_nS_n^{1/2}$, where the system order $n$ is assumed to be known (3) and
      $B_{2(N+1)} = USV^T = U_nS_nV_n^T$ (46)
   (b) Compute $\hat{\Gamma}_{i-1} = [ I_{l(i-1)} \ 0 ] \hat{\Gamma}_i$
   (c) Compute $\hat{\Gamma}_i = [ 0 \ I_{l(i-1)} ] \hat{\Gamma}_i$
3. and 4. Implemented as in the algorithm described in section (4.5)

6 NUMERICAL EXAMPLE

It was considered a “real-life” process (van Overschee and de Moor, 1996), a laboratory setup acting like a dryer. The order is four and it can be determined through the singular values, shown in figure 1.

![Figure 1: Singular values of the system, for the first 200 measurements. We have assumed that the order of the system is 4.](image1)

There is one input only, the voltage over the heating device (a mesh of resistor wires), and one output, the air temperature, measured by a thermocouple at the output – air is fanned through a tube and heated at the inlet. We have compared only the 1QR versions, since they produce basically the same estimates than the 2QR versions, using $i = 15$, a forgetting factor of 0.995 in the second least squares problem of the R-MOESP and, for initialization, we have used the first 200 samples of a set of 1000. Figure 2 shows a snapshot of the estimated eigenvalues of $A$, with both algorithms and the offline version of PO-MOESP (N=200 to N=650). They are all very close and converging to the true values. In figure 3, we show the frequency response of the estimated system, for the offline R-MOESP algorithms (left) and the recursive R-MOESP-1QR algorithm (right), for the first 350 measurements.

![Figure 2: Snapshot (N=200 to N=650) of the trajectory of the estimated eigenvalues, for the algorithms here proposed (version 1QR) and the offline version of PO-MOESP (N=200 to N=650).](image2)

![Figure 3: Frequency response of the offline estimated system (left) and the recursive estimated system (right).](image3)

As to the numerical efficiency, the main differences between the proposed algorithms lie in the first step. In fact, when dealing with LQ decompositions, a small difference in the size of two matrices has a big effect on the number of flops required to complete a LQ decomposition: approximately $\left(\frac{3}{2}n_i^2 + 2n_r^2\right)$ flops (Golub and van Loan, 1996), on a $n_r \times (n_r + 1)$ matrix (the size of matrices in the online algorithms).

If we compare the less efficient algorithm from Van Overschee and De Moor ($n_r = 2(m + l) i = 180$) with its most efficient version, the R-MOESP-1QR ($n_r = (2m + l) i + l = 138$), we can see that less
than 23% in the number of rows produces less 57.7% of the number of flops (7.8 \times 10^6 in the Van Overschee and De Moor’s algorithm and 3.31 \cdot 10^6 in the R-MOESP-1QR version).

7 CONCLUSIONS

In this paper some algorithms for subspace on-line identification have been introduced. They are based on the PO-MOESP (Verhaegen, 1994) and R-MOESP (van Overschee and de Moor (1996) techniques, and therefore implemented through LQ decompositions. However, unlike the original algorithms, the proposed recursive algorithms are based on a least squares interpretation of the orthogonal projections. In fact, \(Z_i, Z_{i+1}\) and even \(Z_i \Pi_j, Z_{i+1}\) are related to least squares problems. This allows us to deal with LQ decompositions of smaller matrices, improving the numerical efficiency of the algorithms without any loss of accuracy. We can also use a modified Householder algorithm, specially developed to improve the efficiency of this LQ-based least squares problems.

Two versions were proposed: the 2QR, where two LQ decompositions are needed to compute \(Z_i, Z_{i+1}\) and even \(Z_i \Pi_j, Z_{i+1}\), and the 1QR version, where only a single LQ decomposition is needed. This last version is, as expected, more efficient (and therefore more interesting) than the first one. However, when compared with the iterative version developed directly from the offline algorithm, the 2QR may also be more efficient than the later. This happens due to the size of the matrices involved. In fact, the LQ decomposition of a \(n_r \times (n_r + 1)\) matrix (the size of matrices in the online algorithms), needs approximately \(\frac{n_r^2}{2} + 2n_r^2\) flops (Golub and van Loan, 1996). This means that two LQ decompositions of lower dimension matrices may be more efficient than one single LQ decomposition, of a matrix with bigger \(n_r\).

Further improvements will be made, in order to consider the update (although partial) of the singular value decomposition. It is also expected the more accurate study of the convergence question.

REFERENCES


