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:

$$\begin{cases} x_{k+1} = Ax_k + Bu_k + Ke_k \\ y_k = Cx_k + Du_k + e_k \end{cases} \quad (1)$$

$$E\left[e_p e_q^T\right] = R_e \delta_{pq} \geqslant 0 \quad (2)$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{l\times n}$, $D \in \mathbb{R}^{l\times m}$, $K \in \mathbb{R}^{n\times l}$ 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:

$$U_p = \begin{bmatrix} U_{(0)} \\ \cdots \\ U_{(i-1)} \end{bmatrix}, \quad U_{p+} = \begin{bmatrix} U_{(0)} \\ \cdots \\ U_{(i)} \end{bmatrix},$$

$$U_f = \begin{bmatrix} U_{(i)} \\ \cdots \\ U_{(2i-1)} \end{bmatrix}, \quad U_{f-} = \begin{bmatrix} U_{(i+1)} \\ \cdots \\ U_{(2i)} \end{bmatrix}$$

where $U_{(i)} = \begin{bmatrix} u_i & \cdots & u_{i+j-1} \end{bmatrix} \in \mathbb{R}^{m\times j}$ and the subscripts p and f denote past and future data, respec-

Delgado C. and dos Santos P. (2004).
RECURSIVE MOESP TYPE SUBSPACE IDENTIFICATION ALGORITHM.
In *Proceedings of the First International Conference on Informatics in Control, Automation and Robotics*, pages 176-182
DOI: 10.5220/0001146801760182

tively. In a similar way, $Y_{(i)} \in \mathbb{R}^{l \times j}$, Y_p, Y_{p+}, Y_f and Y_{f-} are block Hankel matrices, built with output data.

With both input and output data we can also define the block hankel matrices

$$H_p = \begin{bmatrix} U_p \\ Y_p \end{bmatrix}, \quad H_{p+} = \begin{bmatrix} U_{p+} \\ Y_{p+} \end{bmatrix},$$
$$HU = \begin{bmatrix} H_p \\ U_f \end{bmatrix}, \quad H^+U^- = \begin{bmatrix} H_{p+} \\ U_{f-} \end{bmatrix}$$

and the orthogonal projection matrices

$$\begin{aligned} Z_i &= Y_f/_{HU} = Y_f \Pi_{HU} \\ Z_{i+1} &= Y_f/_{H^+U^-} = Y_f \Pi_{H^+U^-} \end{aligned}$$

where $\Pi_M = M^+M = M^T\left(MM^T\right)^+ M$ denotes the orthogonal projection into the row space of M and $(.)^+$ denotes the Moore-Penrose pseudoinverse

Other important matrices are the extended observability matrix $\left(\Gamma_i \in \mathbb{R}^{li \times n}\right)$ and the block Toeplitz matrix $\left(H_i^d \in \mathbb{R}^{li \times mi}\right)$, built with Markov parameters (Delgado and dos Santos, 2004b).

3 MOESP-TYPE ALGORITHMS

The MOESP algorithms can be described as sequences of two main steps (van Overschee and de Moor, 1996). First, the model order n and a basis for the column space of Γ_i are determined: $\widehat{\Gamma}_i = U_n S_n^{1/2}$, where U_n and S_n are given by the Singular Value Decomposition of $Z_i \Pi_{U_f \perp} = Z_i - Z_i/_{Uf}$:

$$\begin{aligned} Z_i \Pi_{U_f \perp} &= USV^T = \quad (3) \\ &= \begin{bmatrix} U_n & U_{n^\perp} \end{bmatrix} \begin{bmatrix} S_n & 0 & 0 \\ 0 & S_{n^\perp} & 0 \end{bmatrix} \\ &\quad \times \begin{bmatrix} V_n^T \\ V_{n1^\perp}^T \\ V_{n2^\perp}^T \end{bmatrix} \\ &= U_n S_n V_n^T + U_{n^\perp} S_{n^\perp} V_{n1^\perp}^T \approx \\ &\approx U_n S_n V_n^T \end{aligned}$$

When the measurements are noise corrupted, it may not be straightforward to distinguish the "nonzero" singular values (S_n) from the "zero" singular values $(S_{n^\perp})$. 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 Γ_i, matrices A and C are obtained by solving the following linear equation, in a least squares sense (van Overschee and de Moor, 1996):

$$\begin{bmatrix} \Gamma_{i-1}^+ Z_{i+1} \\ Y_{(i)} \end{bmatrix} = \begin{bmatrix} A & K_{BD} \\ C & \end{bmatrix} \begin{bmatrix} \Gamma_i^+ Z_i \\ U_f \end{bmatrix} + \rho \quad (4)$$

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

$$\begin{aligned} K_{BD} &= \begin{bmatrix} K_A \\ K_C \end{bmatrix} = \quad (5) \\ &= \begin{bmatrix} B & \Gamma_{i-1}^+ H_{i-1}^d \\ D & 0 \end{bmatrix} - \begin{bmatrix} A \\ C \end{bmatrix} \Gamma_i^+ H_i^d = \\ &= \begin{bmatrix} N_1 \begin{bmatrix} D \\ B \end{bmatrix} & \ldots & N_i \begin{bmatrix} D \\ B \end{bmatrix} \end{bmatrix} \end{aligned}$$

Lopes dos Santos and Martins de Carvalho (dos Santos and de Carvalho, 2003) have shown that K_A can be written as $K_A = K_p K_C = \left(-A\left(\overline{\Gamma}_i^T \overline{\Gamma}_i\right)^{-1} C^T\right) K_C$. Therefore, we can work only with:

$$\begin{aligned} K_{C(B,D)} &= \begin{bmatrix} K_p \\ I_l \end{bmatrix}^+ K_{BD} = \quad (6) \\ &= \begin{bmatrix} N_{C1} \begin{bmatrix} D \\ B \end{bmatrix} & \ldots & N_{Ci} \begin{bmatrix} D \\ B \end{bmatrix} \end{bmatrix} \end{aligned}$$

where

$$\begin{aligned} N_{C1} &= \begin{bmatrix} I_l - L_{C1} & -L_{C2} & \ldots & -L_{Ci} \end{bmatrix} G_i \\ N_{Ck} &= \begin{bmatrix} -L_{Ck} & \ldots & -L_{Ci} & 0 \end{bmatrix} G_i \ (k > 1) \\ L_A &= \begin{bmatrix} L_{A1} & L_{A2} & \ldots & LA \end{bmatrix} = A\Gamma_i^+ \\ L_C &= \begin{bmatrix} L_{C1} & L_{C2} & \ldots & L_{Ci} \end{bmatrix} = C\Gamma_i^+ \\ G_i &= \begin{bmatrix} I_l & 0 \\ 0 & \Gamma_{i-1} \end{bmatrix} \end{aligned}$$

Equation (6) can be rewritten as:

$$\begin{bmatrix} K_{C1} \\ \ldots \\ K_{Ci} \end{bmatrix} = \begin{bmatrix} N_{C1} \\ \ldots \\ N_{Ci} \end{bmatrix} \begin{bmatrix} D \\ B \end{bmatrix} \quad (7)$$

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 Γ_i (Verhaegen, 1994):

$$\widehat{C} = \begin{bmatrix} I_l & 0 \end{bmatrix} \widehat{\Gamma}_i \quad (8)$$

$$\widehat{A} = \Gamma_{i-1}^+ \overline{\Gamma}_i \quad (9)$$

where $\overline{\Gamma}_i = \left[\begin{array}{cc} 0 & I_{l(i-1)} \end{array}\right] \widehat{\Gamma}_i$. These estimates of A and C are the same as the estimates produced by the method of Van Overschee and De Moor ((Chiuso and Picci, 2001), (Delgado and dos Santos, 2004a), (dos Santos and de Carvalho, 2004)).

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

$$vec(\widehat{y}_k) = [M_D \quad M_B] \left[\begin{array}{c} \theta_D \\ \theta_B \end{array}\right] \tag{10}$$

where

$$\begin{aligned} M_D &= (u_k^T \otimes I_l) && (11)\\ M_B &= \sum_{t=1}^{(k-1)} (u_t^T \otimes CA^{k-t-1}) && (12)\\ \theta_D &= vec(D) && (13)\\ \theta_B &= vec(B) && (14) \end{aligned}$$

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

$$\widehat{y}_k = Du_k + \sum_{t=1}^{k-1} CA^{k-t-1}Bu_t = \widehat{y}_{Dk} + \widehat{y}_{Bk} \tag{15}$$

with

$$\begin{aligned} vec(\widehat{y}_{Dk}) &= vec(Du_k) = && (16)\\ &= (u_k^T \otimes I_l)vec(D) \\ vec(\widehat{y}_{Bk}) &= vec\left(\sum_{t=1}^{k-1} CA^{k-t-1}Bu_t\right) && (17)\\ &= \sum_{t=1}^{k-1} \left(u_t^T \otimes CA^{k-t-1}\right) vec(B) \end{aligned}$$

4 RECURSIVE 2QR APPROACH

4.1 The weighted projection

The estimation of Γ_i from the singular value decomposition of $Z_i\Pi_{U_f{}^\perp}$ is implemented through the LQ decomposition of $\left[\begin{smallmatrix} U_f \\ H_p \end{smallmatrix}\right]$:

$$\begin{aligned} \left[\begin{array}{c} U_f \\ H_p \end{array}\right] &= LQ_L = && (18)\\ &= \left[\begin{array}{ccc} L_U & 0 & 0 \\ L_{H1} & L_{H2} & 0 \end{array}\right] \left[\begin{array}{c} Q_{L1} \\ Q_{L2} \\ Q_{L3} \end{array}\right] \end{aligned}$$

where $L_{U1} \in \mathbb{R}^{mi\times mi}$, $L_{H1} \in \mathbb{R}^{(m+l)i\times mi}$, $L_{H2} \in \mathbb{R}^{(m+l)i\times(m+l)i}$ and $Q_{L1} \in \mathbb{R}^{mi\times j}$, $Q_{L2} \in \mathbb{R}^{(m+l)i\times j}$, $Q_{L3} \in \mathbb{R}^{(j-(2m+l)i)\times j}$. Matrices L_{U1} and L_{H2} are two lower triangular matrices and matrix Q_L is orthonormal and

$$\begin{aligned} Q_LQ_L^T &= I_j \Leftrightarrow Q_{Lp}Q_{Lq}^T = \left\{\begin{array}{l} I, \text{if } p = q \\ 0, \text{if } p \neq q \end{array}\right. && (19)\\ Q_L^TQ_L &= \sum_{k=1}^{3} Q_{Lk}^TQ_{Lk} = I_j && (20) \end{aligned}$$

From this decomposition,

$$\begin{aligned} \Pi_{Uf} &= U_f^T(U_fU_f^T)^{-1}U_f = Q_{L1}^TQ_{L1} && (21)\\ \Pi_{HU} &= \left[\begin{array}{cc} Q_{L1}^T & Q_{L2}^T \end{array}\right] \left[\begin{array}{c} Q_{L1} \\ Q_{L2} \end{array}\right] && (22)\\ \Pi_{U_f{}^\perp} &= I_j - \Pi_{Uf} = \left[\begin{array}{cc} Q_{L2}^T & Q_{L3}^T \end{array}\right] \left[\begin{array}{c} Q_{L2} \\ Q_{L3} \end{array}\right] && (23)\\ \Pi_{HU^\perp} &= I_j - \Pi_{HU} = Q_{L3}^TQ_{L3} && (24) \end{aligned}$$

If we consider the least squares problem

$$Y_f = [\theta_{Uf} \quad \theta_{Hp}] \left[\begin{array}{c} U_f \\ H_p \end{array}\right] \tag{25}$$

we can rewrite it as

$$B_{Q_L} = [\theta_{Uf} \quad \theta_{Hp}] \left[\begin{array}{ccc} L_{U1} & 0 & 0 \\ L_{H1} & L_{H2} & 0 \end{array}\right] \tag{26}$$

where $B_{QL} = \left[\begin{array}{ccc} B_U & B_H & B_{UH^\perp} \end{array}\right] = Y_fQ_L^T$. Then,

$$\begin{aligned} Z_i &= Y_f\Pi_{HU} = [B_U \quad B_H] \left[\begin{array}{c} Q_{L1} \\ Q_{L2} \end{array}\right] && (27)\\ &= B_UQ_{L1} + B_HQ_{L2} \end{aligned}$$

As to the weighted projection, it can be expressed as

$$\begin{aligned} Z_i\Pi_{U_f{}^\perp} &= \theta_{Hp}H_p\Pi_{Uf}^\perp = && (28)\\ &= \left[\begin{array}{cc} \theta_{Uf} & \theta_{Hp} \end{array}\right] \left[\begin{array}{c} U_f \\ H_p \end{array}\right] \Pi_{Uf}^\perp = \\ &= Z_i\Pi_{U_f{}^\perp} = Z_i - (Z_i/U_f) = \\ &= Z_i - (Z_iQ_{L1}^TQ_{L1}) = \\ &= B_HQ_{L2} \end{aligned}$$

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

$$(B_HQ_{L2})(B_HQ_{L2})^T = B_HB_H^T \tag{29}$$

and, if $B_HQ_{L2} = USV^T$,

$$(B_HQ_{L2})(B_HQ_{L2})^T = US^2U^T = B_HB_H^T \tag{30}$$

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

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

$$\begin{bmatrix} \Gamma_{i-1}^{+} Z_{i+1} \\ Y_{(i)} \end{bmatrix}^T = \begin{bmatrix} \Gamma_i^{+} Z_i \\ U_f \end{bmatrix}^T \theta_{ACK} \qquad (31)$$

with

$$\theta_{ACK} = \begin{bmatrix} A & K_A \\ C & K_C \end{bmatrix}^T$$

is the same as the least squares solution of

$$\begin{aligned} B_{(N)} &= Q_{(N)}^T \begin{bmatrix} \Gamma_{i-1}^{+} Z_{i+1} \\ Y_{(i)} \end{bmatrix}^T = \qquad (32) \\ &= \begin{bmatrix} R_{(N)} \\ 0 \end{bmatrix} \theta_{ACK} \end{aligned}$$

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

$$\begin{aligned} Z_{i+1} &= Y_{f-}\Pi_{H+U-} = \qquad (33) \\ &= \begin{bmatrix} B_{U-} & B_{H+} \end{bmatrix} \begin{bmatrix} Q_{L1(i+1)} \\ Q_{L2(i+1)} \end{bmatrix} \end{aligned}$$

where

$$\begin{bmatrix} U_{f-} \\ H_{p^+} \end{bmatrix} = L_{(i+1)} \begin{bmatrix} Q_{L1(i+1)} \\ Q_{L2(i+1)} \\ Q_{L3(i+1)} \end{bmatrix} \qquad (34)$$

$$B_{U-} = Y_{f-} Q_{L1(i+1)}^T \qquad (35)$$

$$B_{H+} = Y_{f-} Q_{L2(i+1)}^T \qquad (36)$$

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 updates 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 Nr > 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 $\left(\phi_{U(N+1)} \text{ and } \phi_{Y(N+1)}\right)$.
   (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 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 Γ_i and Γ_{i-1}
   (a) Determine $\widehat{\Gamma}_i = U_n S_n^{1/2}$, where the system order n is assumed to be known (3) and
   $$B_{H(N+1)} = USV^T = U_n S_n V_n^T \qquad (37)$$
   (b) Compute $\widehat{\Gamma}_{i-1} = \begin{bmatrix} I_{l(i-1)} & 0 \end{bmatrix} \widehat{\Gamma}_i$
3. Matrices A, C and K_{BD}
   (a) Update the QR decomposition matrices $R_{AC(N)}$ and $B_{AC(N)}$:
   $$\begin{bmatrix} R_{AC(N)} \\ r_{N+1}^T \end{bmatrix} = Q_{AC(N+1)} \begin{bmatrix} R_{AC(N+1)} \\ 0 \end{bmatrix}$$
   $$\begin{bmatrix} B_{AC(N+1)} \\ \times \end{bmatrix} = Q_{AC(N+1)}^T \begin{bmatrix} B_{AC(N)} \\ b_{N+1}^T \end{bmatrix}$$
   where
   $$r_{N+1} = \begin{bmatrix} \widehat{\Gamma}_i^{+} z_{i(N+1)} \\ \begin{bmatrix} 0 & I_{mi} \end{bmatrix} \phi_{U(N+1)} \end{bmatrix}$$
   $$b_{N+1} = \begin{bmatrix} \widehat{\Gamma}_{i-1}^{+} z_{i+1(N+1)} \\ \begin{bmatrix} 0 & I_{li} \end{bmatrix} \phi_{Y(N+1)} \end{bmatrix}$$
   (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 $\left(\phi_{U(N+1)} \text{ and } \phi_{Y(N+1)}\right)$.
   (b) Update the decompositions (18) and (34).
2. The column space of Γ_i, $\overline{\Gamma}_i$ and Γ_{i-1}
   (a) Determine $\widehat{\Gamma}_i = U_n S_n^{1/2}$, where the system order n is assumed to be known (3) and

$$B_{H(N+1)} = USV^T = U_n S_n V_n^T$$

   (b) Compute $\widehat{\Gamma}_{i-1} = \begin{bmatrix} I_{l(i-1)} & 0 \end{bmatrix} \widehat{\Gamma}_i$
   (c) Compute $\widehat{\overline{\Gamma}}_i = \begin{bmatrix} 0 & I_{l(i-1)} \end{bmatrix} \widehat{\Gamma}_i$
3. Matrices A and C
   (a) Determine $\widehat{A} = \widehat{\Gamma}_{i-1}^{+} \widehat{\overline{\Gamma}}_i$
   (b) Determine $\widehat{C} = \begin{bmatrix} I_l & 0 \end{bmatrix} \widehat{\Gamma}_i$
4. Matrices B and D ("minimum simulation error method")
   (a) Update the QR decomposition

$$\begin{bmatrix} B_{BD(N)} \\ \mathcal{Y}_{BD} \end{bmatrix} = \begin{bmatrix} R_{BD(N)} \\ \mathcal{X}_D \;\; \mathcal{X}_B \end{bmatrix} \begin{bmatrix} \theta_D \\ \theta_B \end{bmatrix}$$

   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

$$\begin{aligned} \mathcal{X}_B &= \sum_{t=1}^{k-1} u_t^T \otimes (CA^{k-1-t}) \in \mathbb{R}^{l\times mn} \\ \mathcal{X}_D &= u_k^T \otimes I_l \in \mathbb{R}^{l\times lm} \\ \mathcal{Y}_{BD} &= y_k \in \mathbb{R}^{l\times 1} \\ \theta_B &= vec(B) \in \mathbb{R}^{mn\times 1} \\ \theta_D &= vec(D) \in \mathbb{R}^{ml\times 1} \end{aligned}$$

   (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 $Z_i \Pi_{U_f{}^\perp}$ are obtained from a unique LQ decomposition:

$$\begin{bmatrix} U_f \\ H_p \\ Y_{(i)} \end{bmatrix} = \begin{bmatrix} L_U & 0 & 0 & 0 \\ L_{H1} & L_{H2} & 0 & 0 \\ L_{Y1} & L_{Y2} & L_{Y3} & 0 \end{bmatrix} \begin{bmatrix} Q_{L1} \\ Q_{L2} \\ Q_{L3} \\ Q_{L4} \end{bmatrix} \quad (38)$$

In fact,

$$\Pi_{Uf} = U_f^T (U_f U_f^T)^{-1} U_f = Q_{L1}^T Q_{L1} \quad (39)$$

$$\Pi_{HU} = \begin{bmatrix} Q_{L1}^T & Q_{L2}^T \end{bmatrix} \begin{bmatrix} Q_{L1} \\ Q_{L2} \end{bmatrix} \quad (40)$$

and, since the column space of $\begin{bmatrix} H_{p+}^T & U_{f-}^T \end{bmatrix}$ is spanned by the columns of $\begin{bmatrix} H_p^T & U_f^T & Y_{(i)}^T \end{bmatrix}$,

$$\Pi_{H^+U^-} = \begin{bmatrix} Q_{L1}^T & Q_{L2}^T & Q_{L3}^T \end{bmatrix} \begin{bmatrix} Q_{L1} \\ Q_{L2} \\ Q_{L3} \end{bmatrix} \quad (41)$$

Therefore, if we define

$$\begin{aligned} Y_f Q_L^T &= \begin{bmatrix} B_1 & B_2 & B_3 & B_4 \end{bmatrix} = \quad (42) \\ &= \begin{bmatrix} Y_{(i)} Q_L^T \\ Y_{f-} Q_L^T \end{bmatrix} = \\ &= \begin{bmatrix} b_1 & b_2 & b_3 & b_4 \\ B_{m1} & B_{m2} & B_{m3} & B_{m4} \end{bmatrix} \end{aligned}$$

then

$$Z_i = \begin{bmatrix} B_1 & B_2 \end{bmatrix} \begin{bmatrix} Q_{L1} \\ Q_{L2} \end{bmatrix} \quad (43)$$

$$Z_{i+1} = \begin{bmatrix} B_{m1} & B_{m2} & B_{m3} \end{bmatrix} \begin{bmatrix} Q_{L1} \\ Q_{L2} \\ Q_{L3} \end{bmatrix} \quad (44)$$

and $Z_i \Pi_{U_f{}^\perp} = B_2 Q_{L2}$.

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 $\left(\phi_{U(N+1)} \text{ and } \phi_{Y(N+1)}\right)$.
   (b) Update the decomposition (38)
   (c) Determine z_i and z_{i+1}, the last columns of Z_i and Z_{i+1}, as in (43) and (44).
2. Determine $\widehat{\Gamma}_i = U_n S_n^{1/2}$, where

$$B_{2(N+1)} = USV^T = U_n S_n V_n^T \quad (45)$$

   and the system order n is assumed to be known (3). Compute $\widehat{\Gamma}_{i-1} = \begin{bmatrix} I_{l(i-1)} & 0 \end{bmatrix} \widehat{\Gamma}_i$
3. and 4. Implemented as in the algorithm described in section (4.4)

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 $\left(\phi_{U(N+1)} \text{ and } \phi_{Y(N+1)}\right)$.
   (b) Update the decomposition (38)
2. The column space of Γ_i, $\overline{\Gamma}_i$ and Γ_{i-1}
   (a) Determine $\widehat{\Gamma}_i = U_n S_n^{1/2}$, where the system order n is assumed to be known (3) and
   $$B_{2(N+1)} = USV^T = U_n S_n V_n^T \qquad (46)$$
   (b) Compute $\widehat{\Gamma}_{i-1} = \left[\begin{array}{cc} I_{l(i-1)} & 0 \end{array}\right] \widehat{\Gamma}_i$
   (c) Compute $\widehat{\overline{\Gamma}}_i = \left[\begin{array}{cc} 0 & I_{l(i-1)} \end{array}\right] \widehat{\Gamma}_i$
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.

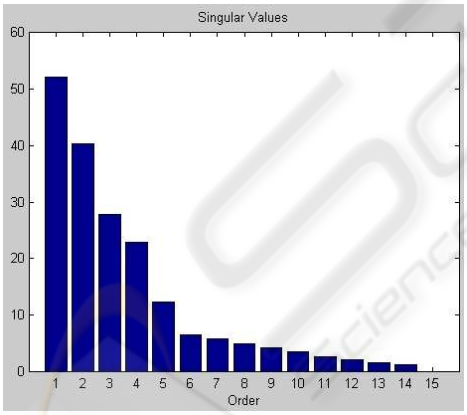

Figure 1: Singular values of the sytem, for the first 200 measurements. We have assumed that the order of the system is 4.

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 heateds 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.

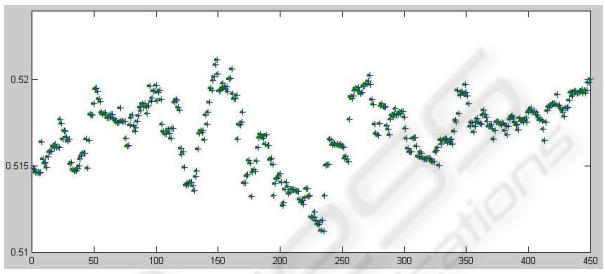

Figure 2: Snapshot (N=200 to N=650) of the trajectory of the estimated eigenvalues, for the algorithms here proposed (version 1QR) and the offline version of PO-MOESP (N=200 to N=650).

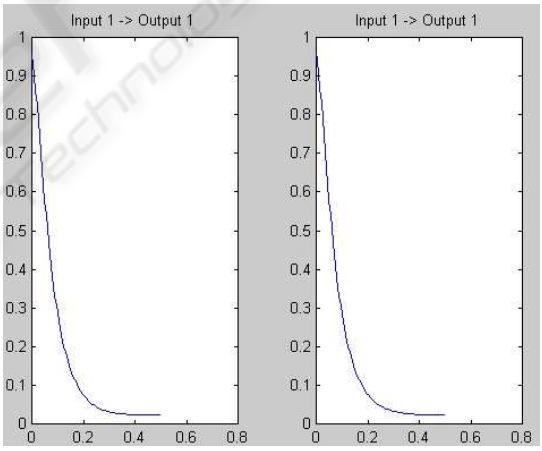

Figure 3: Frequency response of the offline estimated system (left) and the recursive estimated system (right).

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 $\left(\frac{4}{3}n_r^3 + 2n_r^2\right)$ 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 ($7.8 \cdot 10^6$ 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 $Z_i \Pi_{U_f{}^\perp}$ 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 $Z_i \Pi_{U_f{}^\perp}$, 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 on-line algorithms), needs approximately $\left(\frac{4}{3}n_r^3 + 2n_r^2\right)$ 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

Chiuso, A. and Picci, G. (2001). Asymptotic variance of subspace estimates. In *Proc. CDC'01 - 40th Conference on Decision and Control, Orlando, Florida, USA*, pages 3910–3915.

Cho, Y., Xu, G., and Kailath, T. (1994). Fast recursive identification of state space models via exploitation of displacement structure. *Automatica*, 30:45–59.

Delgado, C. and dos Santos, P. L. (2004a). Analysis and efficiency improvement of the estimation of matrices a and c in some subspace identification methods. In *Proc. SICPRO'05 – 4th International Conference on System Identification and Control Problems, Moscow, Russia (Accepted for Publication)*.

Delgado, C. and dos Santos, P. L. (2004b). Exploring the linear relations in the estimation of matrices b and d in subspace identification methods. In *Proc. ICINCO'04 – 1st International Conference on Informatics in Control, Automation and Robotics, Setúbal, Portugal*.

dos Santos, P. L. and de Carvalho, J. L. M. (2003). Improving the numerical efficiency of the b and d estimates produced by the combined deterministic-stochastic subspace identification algorithms. In *Proc. CDC'03 – 42nd Conference on Decision and Control, Maui, Hawaii, USA*, pages 3473–3478.

dos Santos, P. L. and de Carvalho, J. L. M. (2004). A new insight to the a and c matrices extraction in a MOESP type subspace identification algorithm. In *Proc. Controlo'04 – 6th Portuguese Conference on Automatic Control, Faro, Portugal*.

Golub, G. and van Loan, C. (1996). *Matrix Computations*. Johns Hopkins University Press, 3rd edition.

Gustafsson, T., Lovera, M., and Verhaegen, M. (1998). A novel algorithm for recursive instrumental variable based subspace identification. In *Proc. CDC'98, Conference on Decision and Control*, pages 3920–3925.

Larimore, W. (1990). Canonical variate analysis in identification, filtering and adaptive control. In *Proc. CDC'90 – 29th Conference on Decision and Control*, pages 596–604.

Ljung, L. (1983). *Theory and Practice of Recursive Identification*. The MIT Press.

Ljung, L. (1987). *System Identification: Theory for the User*. Prentice-Hall.

Moonen, M., de Moor, B., Vandenberghe, L., and Vandewalle, J. (1989). On- and off-line identification of linear state space models. *International Journal of Control*, 49:219–232.

Oku, H. and Kimura, H. (2000). Recursive algorithms of the state-space subspace system identification. *Automatica*.

van Overschee, P. and de Moor, B. (1996). *Subspace Identification for linear systems*. Kluwer Academic Publishers.

Verhaegen, M. (1994). Identification of the deterministic part of MIMO state space models given in innovations form from input-output data. *Automatica*, 30:61–74.

Viberg, M., Ottersten, B., Wahlberg, B., and Ljung, L. (1991). Performance of subspace based state-space systems identification methods. In *Proc. IFAC'91 – 12th IFAC World Congress, Sydney, Australia*.