Pergamon

0005-1098(95)00072-0

Automatica, Vol. 31, No. 12, pp. 1853–1864, 1995

0005-1098/95 $9.50 + 0.00

A Unifying Theorem for Three Subspace System Identification Algorithms*

PETER VAN OVERSCHEE† and BART DE MOOR†

A unifying theorem indicates strong similarities between three different published subspace identification algorithms for combined deterministic–stochastic systems.

Key Words—System identification; subspace methods; multivariable systems; state-space methods; linear algebra; Kalman filters; difference equations; stochastic systems.

Abstract—The aim of this paper is to indicate and explore the similarities between three different subspace algorithms for the identification of combined deterministic–stochastic systems. The similarities between these algorithms have been obscured, due to different notations and backgrounds. It is shown that all three algorithms are special cases of one unifying theorem. The comparison also reveals that the three algorithms use exactly the same subspace to determine the order and the extended observability matrix, but that the weighting matrix, used to calculate a basis for the column space of the observability matrix is different in the three cases.

1. INTRODUCTION

A number of algorithms to identify multi-input multi-output (MIMO) combined deterministic–stochastic systems have been published. In contrast to 'classical' algorithms (Ljung, 1987, Söderström and Stoica, 1989), these subspace algorithms do not suffer from the problems caused by a priori parametrizations and non-linear optimizations. They identify MIMO systems in a very simple and elegant way. In this paper, we shall indicate and explore some striking similarities between three different subspace algorithms for the identification of combined deterministic–stochastic systems. This comparison is done through the introduction of a unifying theorem, of which all three published algorithms are a special case. We believe that this observation will contribute considerably to a further understanding of subspace algorithms for system identification.

We consider three different algorithms in this paper. The first is that due to Larimore (1990). It is based on statistical arguments, and makes extensive use of principal angles and directions. The method is often referred to as 'canonical variate analysis' (CVA). The second algorithm we shall consider is the MOESP algorithm of Verhaegen (1994). MOESP stands for 'multivariable output-error state space'. The third algorithm is the N4SID algorithm of Van Overschee and De Moor (1994), which is also treated from a different point of view by Viberg *et al.* (1993). N4SID stands for 'numerical algorithms for subspace state space system identification' and should be read as a Californian license plate: *enforce it.* The last two algorithms (MOESP and N4SID) are based on geometrical and linear algebra concepts.

The identification problem considered in the combined deterministic–stochastic identification papers is the following. Let $u_k \in \mathbb{R}^m$ and $y_k \in \mathbb{R}^l$ be the observed input and output generated by the unknown system

$$x_{k+1} = Ax_k + Bu_k + w_k, \tag{1}$$

$$y_k = Cx_k + Du_k + v_k, \tag{2}$$

with

$$\mathbf{E}\left[\begin{pmatrix} w_k \\ v_k \end{pmatrix}\begin{pmatrix} w_l^{\mathrm{T}} & v_l^{\mathrm{T}} \end{pmatrix}\right] = \begin{pmatrix} Q & S \\ S^{\mathrm{T}} & R \end{pmatrix}\delta_{kl} \geq 0 \tag{3}$$

and $A, Q \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{l\times n}$, $D \in \mathbb{R}^{l\times m}$, $S \in \mathbb{R}^{n\times l}$ and $R \in \mathbb{R}^{l\times l}$ (here $\mathbf{E}$ denotes the expected value operator and δ_{kl} the Kronecker delta). $v_k \in \mathbb{R}^l$ and $w_k \in \mathbb{R}^n$ are unobserved, Gaussian-distributed, zero-mean, white noise vector sequences. $\{A, C\}$ is assumed to be observable while $\{A, [B\ Q^{1/2}]\}$ is assumed to be controllable. The main problem is then stated as follows. Given input and output measurements $u_1, \ldots, u_N$ and $y_1, \ldots, y_N$, where N is large and

* Received 7 February 1994; revised 18 July 1994; received in final form 8 November 1994. The original version of this paper was presented at the 10th IFAC Symposium on System Identification, which was held in Copenhagen, Denmark during 4–6 July 1994. The Published Proceedings of this IFAC meeting may be ordered from: Elsevier Science Limited, The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, U.K. This paper was recommended for publication in revised form by Associate Editor B. Wahlberg under the direction of Editor Torsten Söderström. Corresponding author Dr Peter Van Overschee. Tel. +32 16 321095; Fax +32 16 321986. E-mail peter.vanoverschee@esat.kuleuven.ac.be.

† ESAT, Katholieke Universiteit Leuven, Kardinaal Mercierlaan 94, 3001 Leuven (Heverlee), Belgium.

sometimes (e.g. for statistical analysis reasons) is required to go to infinity ($N \to \infty$). Given the fact that these two sequences (u_k and y_k) are generated by an unknown combined deterministic–stochastic model of the form described by (1)–(3), find A, B, C, D, Q, R and S up to within a similarity transformation.

Before we proceed, we note that Viberg *et al.* (1991) presented a comparison along the same line as developed here. In their paper the similarities between three subspace algorithms were also investigated. However, the subspace algorithms discussed and compared by Viberg *et al.* (1991) consider the identification of purely deterministic systems. This paper presents an extension of these results to the identification of combined deterministic–stochastic systems as in (1)–(3).

Subspace algorithms basically consist of two steps (see Fig. 1). As a first step, the algorithm computes a certain characteristic subspace from the given input–output data, which coincides with the subspace generated by the columns of the extended observability matrix of the system (Γ_i). The dimension of this subspace is equal to n, the order of the system to be identified. Thus, in the first step, the order and the extended observability matrix of the system are determined directly from the given input–output data (full lines in Fig. 1). For the second step, the considered algorithms apply either of the following strategies.

- Two algorithms (Viberg *et al.*, 1993; Verhaegen, 1994) determine two system matrices (A and C) directly from the extended observability subspace. This is typically done by making use of the shift invariance of the subspace spanned by the columns of the extended observability matrix. This method is widely used in realization theory (Kung, 1978), stochastic identification (Akaike, 1975; Van Overschee and De Moor, 1993) and direction of arrival estimation (Paulraj *et al.*, 1986). After A and C have been determined, they are used to determine the remaining system matrices (B, D, Q, R, S). This is illustrated on the left-hand side of Fig. 1.
- The other two algorithms (Larimore, 1990; Van Overschee and De Moor, 1994) use in the second step the extended observability matrix to implicitly determine two state sequences. When these state sequences are then combined with the original input–output data, all system matrices (A, B, C, D, Q, R, S) can be determined directly by solving a set of equations in a least-squares sense. This is illustrated on the right-hand side of Fig. 1. There is a certain elegance about this second strategy, since all the unknowns are determined in one step, and not in two as in the first strategy.

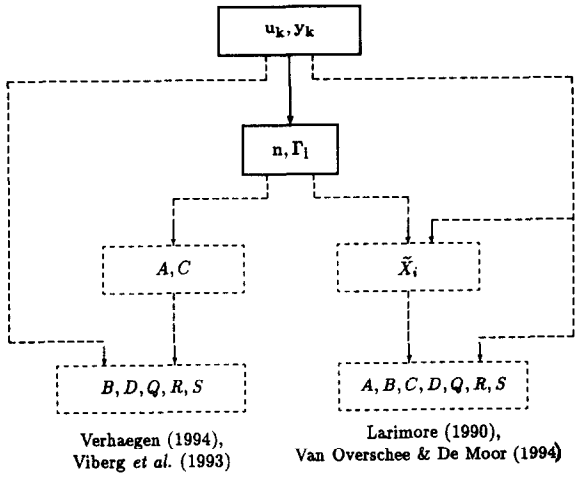

Fig. 1. The two steps of a subspace algorithm. The full lines represent the first step: the determination of the order (n) and the extended observability matrix (Γ_i) from the input–output data u_k, y_k. This first step is analyzed in the paper. The dotted lines represent the two possibilities for the second step. The left-hand side illustrates the strategy followed by Verhaegen (1994) and Viberg *et al.* (1993), while the right-hand side illustrates the strategy of Larimore (1990) and Van Overschee and De Moor (1994).

Since all algorithms determine in a first step the order and the extended observability matrix of the system, we shall focus our attention in this paper on that problem. We introduce a unifying theorem that allows the determination of the order and the extended observability matrix from input–output data. The basic subspace in this theorem is obtained by an oblique projection, as was already mentioned by Van Overschee and De Moor (1994) and Viberg *et al.* (1993). The only difference between the algorithms lies in the use of different weighting matrices.

This paper is organized as follows. In Section 2 we introduce some notation. Section 3 presents the unifying theorem, which is the main result of this paper. In Section 4 it is shown how the three algorithms fit in this unifying theory. Section 5 gives an intuitive interpretation of the weighting matrices. The theory is illustrated by an example in Section 6. Section 7 summarizes the main results of this paper.

2. NOTATION

In this section we introduce the notation used for input and output block Hankel matrices, for projections and for some matrix operations.

Input and output block Hankel matrices are defined as

$$U_{0|i-1} \stackrel{\text{def}}{=} \begin{pmatrix} u_0 & u_1 & u_2 & \dots & u_{j-1} \\ u_1 & u_2 & u_3 & \dots & u_j \\ \vdots & \vdots & \vdots & & \vdots \\ u_{i-1} & u_i & u_{i+1} & \dots & u_{i+j-2} \end{pmatrix},$$

$$Y_{0|i-1} \stackrel{\text{def}}{=} \begin{pmatrix} y_0 & y_1 & y_2 & \dots & y_{i-1} \\ y_1 & y_2 & y_3 & \dots & y_j \\ \vdots & \vdots & \vdots & & \vdots \\ y_{i-1} & y_i & y_{i+1} & \dots & y_{i+j-2} \end{pmatrix},$$

where we assume that $j \to \infty$ throughout the paper. i is a user-defined index, which is 'large enough'. The subscripts on U and Y denote the subscript of the first and last element of the first column.

We denote the 'past' input block Hankel matrix by $U_{0|i-1}$ and the 'future' input block Hankel matrix by $U_{i|2i-1}$. A similar notation applies for the past and future output block Hankel matrices. Furthermore, for notational convenience and following Larimore (1990), we define p, f and u as follows: the past inputs and outputs

$$p \stackrel{\text{def}}{=} \begin{pmatrix} U_{0|i-1} \\ Y_{0|i-1} \end{pmatrix},$$

the future outputs

$$f \stackrel{\text{def}}{=} (Y_{i|2i-1}),$$

and the future inputs

$$u \stackrel{\text{def}}{=} (U_{i|2i-1}).$$

The extended ($i > n$) observability matrix Γ_i (where the subscript i denotes the number of block rows) is defined as

$$\Gamma_i \stackrel{\text{def}}{=} \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{i-1} \end{pmatrix}.$$

The Kalman filter state sequence $\tilde{X}_i$ is defined as in Van Overschee and De Moor (1994):

$$\tilde{X}_i \stackrel{\text{def}}{=} (\tilde{x}_i \quad \tilde{x}_{i+1} \quad \tilde{x}_{i+2} \quad \dots \quad \tilde{x}_{i+j-1}).$$

Each column is the output of a non-steady-state Kalman filter built from the matrices of the system (if they were known). The j columns of $\tilde{X}_i$ are thus the outputs of a bank of j Kalman filters in parallel (see also Fig. 2). When the system matrices A, B, C, D, Q, R, and S are known, this sequence can be determined easily by combining the input–output data with the known system matrices. The point is that this state sequence can also be obtained directly from the input–output data, without any knowledge of the system matrices. This observation is at the heart of the approach elaborated by Van Overschee and De Moor (1994), to which we refer for more details.

Π_A denotes the operator that projects the row space of a matrix onto the row space of A (which is assumed to be of full rank):

$$\Pi_A \stackrel{\text{def}}{=} A^{\mathrm{T}}(AA^{\mathrm{T}})^{-1}A$$

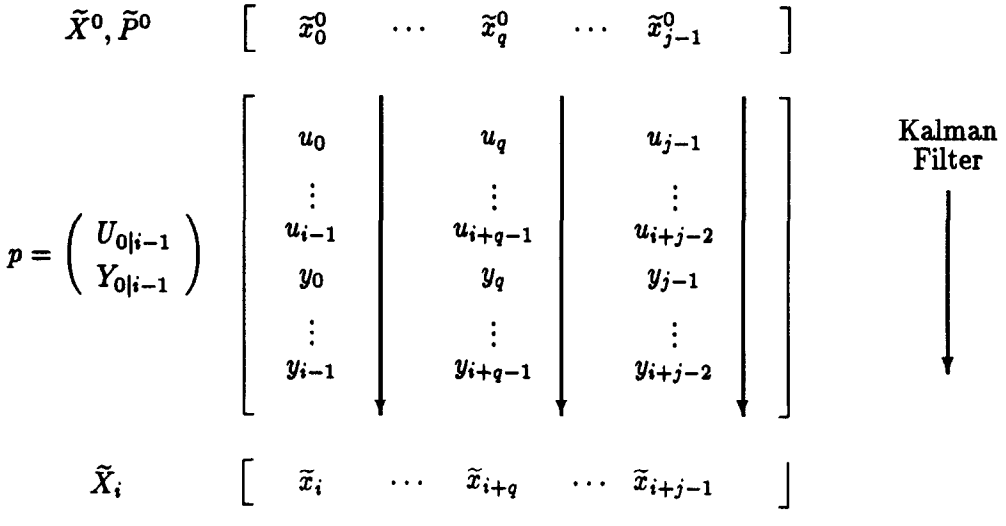

Fig. 2. Interpretation of the sequence $\tilde{X}_i$ as a sequence of non-steady-state Kalman filter state estimates based upon i measurements of u_k and y_k. When the system matrices A, B, C, D, Q, R and S are known, the state $\tilde{x}_{i+q}$ can be determined from a non-steady-state Kalman filter as follows. Start the filter at time q, with an initial state estimate $\tilde{x}_q^0$, which is a function of the past and future inputs as explained by Van Overschee and De Moor (1994). Now iterate the non-steady-state Kalman filter over i time steps (the downwards vertical arrow). The Kalman filter will then return a state estimate $\tilde{x}_{i+q}$. This procedure could be repeated for each of the j columns, and thus we speak about a *bank* of non-steady-state Kalman filters. The major observation in subspace algorithms is that the system matrices A, B, C, D, Q, R and S do not have to be known to determine the state sequence $\tilde{X}_i$. It can be determined directly from input–output data.

$A^\perp$ denotes the subspace perpendicular to the row space of A. B/A is shorthand for the projection of the row space of B onto the row space of A:

$$B/A \stackrel{\text{def}}{=} B\Pi_A = BA^T(AA^T)^{-1}A.$$

The Frobenius norm squared of a matrix $A \in \mathbb{R}^{m\times n}$ is defined as

$$\|A\|_F^2 \stackrel{\text{def}}{=} \sum_{k=1}^{m}\sum_{l=1}^{n} a_{kl}^2.$$

The Moore–Penrose inverse of a matrix A is denoted by $A^\dagger$ and the $n \times n$ identity matrix by I_n.

3. A UNIFYING FRAMEWORK

In this section we present a unifying framework for the determination of the order and the extended observability matrix of a system, directly from given input–output data. In a first subsection we shall stress the intuition that leads to the theorem. In a second subsection we state and prove the theorem, and show how it ties in with the intuition of the first subsection. In Section 4 we show that the three aforementioned algorithms are special cases of the unifying theorem.

3.1. *Intuition*

The goal of an identification procedure is to find a model that behaves in approximately the same way as the process under consideration. This goal is 'classically' solved by minimizing a 'prediction error criterion'. This criterion expresses the 'prediction performance' of the model on the given data set. The minimizing solution is designated as the optimal model (see e.g. Ljung, 1987). In the framework of subspace identification, we reach the identification goal by solving two subsequent problems.

Optimal prediction. As stated before, we want to find a model that will predict the behavior of the process sufficiently accurately. This can be formulated as follows: predict the future outputs (f) as accurately as possible, using all the information that can be obtained from the past (p), and using the knowledge of the inputs that will be presented to the system in the future (u).

Complexity reduction. Apart from the fact that we want to find a model that can predict the future, we also want the complexity of this model to be as low as possible. The model has to be as compact as possible. Or, equivalently, we want to reduce the complexity of the amount of 'information' of the past that we need to keep track of to predict the future.

We shall now formulate this intuition mathematically.

Optimal prediction. Inspired by the linearity of the system, we propose to combine the past (p) and the future inputs (u) linearly to predict the future outputs (f). We denote the linear combinations respectively by L_p and L_u. The quality of the prediction is measured in the Frobenius norm. Mathematically, the first part of the identification goal then becomes

$$\min_{\substack{L_p \in \mathbb{R}^{li\times(m+l)i} \\ L_u \in \mathbb{R}^{li\times mi}}} \left\| f - (L_p \quad L_u)\begin{pmatrix} p \\ u \end{pmatrix} \right\|_F^2. \tag{4}$$

The optimal combination of the past (p) to predict the future is thus $L_p p$. Geometrically, the row space of $L_p p$ can be interpreted as the *oblique projection* of the row space of f along the row space of u on the row space of p (see below). This oblique projection is denoted by $\mathcal{O} \in \mathbb{R}^{li\times j}$:

$$\mathcal{O} \stackrel{\text{def}}{=} L_p p.$$

Complexity reduction. As a second step, we need to reduce the complexity of $\mathcal{O}$. Since the rows of $\mathcal{O}$ span an li-dimensional subspace in the j-dimensional ambient space, we can introduce a complexity reduction by reducing the subspace dimension to n (the order of the system). Intuitively, this implies that we only have to remember n different directions of the past to predict the future. Mathematically, the second step can be formulated as

$$\min_{\mathcal{R} \in \mathbb{R}^{li\times j}} \|W_1(\mathcal{O} - \mathcal{R})W_2\|_F^2, \tag{5}$$

$$\text{constrained by rank } (\mathcal{R}) = n,$$

where $W_1 \in \mathbb{R}^{li\times li}$ and $W_2 \in \mathbb{R}^{j\times j}$ are user-defined weighting matrices (see below). These weighting matrices determine which part of the 'information' of $\mathcal{O}$ is important to retain. A more rigorous interpretation of the weighting matrices is presented in Section 5. Since we do not want to lose any 'information' (rank) owing to the weighting, we should make sure that

$$\text{rank } (W_1 \mathcal{O} W_2) = \text{rank } (\mathcal{O}).$$

3.2. *A unifying theorem*

In this subsection we state and prove the unifying theorem. We also illustrate how the intuition of the previous subsection ties in to this Theorem.

Theorem 1. *Unifying theorem.*

1. The process noise w_k and the measurement noise v_k are not identically zero.

2. The input u_k is uncorrelated with the process noise w_k and the measurement noise v_k.
3. The input u_k is persistently exciting of order $2i$ (Ljung, 1987).
4. An infinite number of measurements are available: $j \to \infty$.
5. W_1 is of full rank and W_2 is such that rank (p) = rank (pW_2).

Then

$$\mathcal{O} = [(f/u^\perp)(p/u^\perp)^T][(p/u^\perp)(p/u^\perp)^T]^{-1}p \quad (6)$$

and, with the singular-value decomposition

$$W_1 \mathcal{O} W_2 = (U_1 \quad U_2)\begin{pmatrix} S_1 & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}, \quad (7)$$

we have the following.

(a) The order of the system (1)–(3) is equal to the number of non-zero singular values in (7).

(b) The optimal reduction $\mathcal{R}$ can be taken equal to

$$\mathcal{R} = W_1^{-1} U_1 S_1 V_1^T W_2^\dagger.$$

(This is the minimum-norm solution of the optimization problem (5). When W_2 is rank-deficient, there is more than one solution. See also the Appendix.)

(c) The extended observability matrix Γ_i can be taken equal to:

$$\Gamma_i = W_1^{-1} U_1 S_1^{1/2}. \quad (8)$$

(Note that Γ_i is only determined up to a non-singular similarity transformation T. This implies that $W_1^{-1} U_1 S_1^{1/2} T$ is also a valid choice for Γ_i.)

(d) The part of the Kalman state sequnce $\tilde{X}_i$ that lies in the column space of W_2 can be recovered from

$$\tilde{X}_i W_2 = S_1^{1/2} V_1^T. \quad (9)$$

(Using the same similarity transformation T as in (c), we find $\tilde{X}_i W_2 = T^{-1} S_1^{1/2} V_1^T$.)

(e) The 'full' Kalman state sequence $\tilde{X}_i$ can be recovered from

$$\tilde{X}_i = \Gamma_i^\dagger \mathcal{O}. \quad (10)$$

The proof of the unifying theorem can be found in the Appendix.

Discussion. First we provide some background on the assumptions (the numbers of the notes correspond to the numbers of the assumptions in the theorem).

2. Assumption 2 is satisfied for open-loop operation of the system. In closed loop the noise would be fed back to the inputs through the controller.
3. Persistency of excitation means that u_k should at least contain i sinusoids with distinct frequencies. Any white or colored noise signal also satisfies this assumption.
4. Assumption 4 is needed for the asymptotic analysis. In practice it suffices to have a large amount of data.
5. Assumption 5 is made to avoid loss of rank of $\mathcal{O}$ when multiplying by the weighting matrices (see also the proof of the theorem in the Appendix).

The intuition of the previous section can now be related to Theorem 1.

Optimal prediction. From the proof in the Appendix, we indeed find that $\mathcal{O}$ is equal to the oblique projection of f along u on p. Equation (6) is just one way of writing this projection.

An important observation is that, independently of the weights W_1 or W_2 in the second step, the matrix that is rank-deficient and that determines the system order and matrices is the oblique projection $\mathcal{O}$. Since (see the proof of the theorem in the Appendix) $[p/u^\perp(p/u^\perp)^T]^{-1}p$ is of full rank, we see from (6) that the order and the extended observability matrix are fully determined by the rank and the column space of

$$[(f/u^\perp)(p/u^\perp)^T],$$

which is thus the matrix that lies at the heart of the theorem.

Complexity reduction. The optimal reduction $\mathcal{R}$ (the solution of the minimization problem (5)) can be taken equal to (Theorem 1)

$$\mathcal{R} = W_1^{-1} U_1 S_1 V_1^T W_2^\dagger.$$

Note that $\mathcal{R} = \mathcal{O}$ when all the assumptions of Theorem 1 are satisfied. In that case $\mathcal{R}$ is merely a basis for the row space of $\mathcal{O}$. However, when $j \neq \infty$ or when the data-generating system is not linear, the singular values of $W_1 \mathcal{O} W_2$, (7), are all different from zero. In that case the row space of $\mathcal{O}$ is of dimension li, and the order has to be chosen equal to the number of 'dominant' singular values. The complexity reduction step is then truly a reduction of the dimension of the row space of $\mathcal{O}$, and the weights W_1 and W_2 play an important role in determining which part of the original row space of $\mathcal{O}$ is retained. The error induced by this reduction is then

$$\|W_1(\mathcal{O} - \mathcal{R})W_2\|_F^2 = \sum_{k=n+1}^{li} \sigma_k^2,$$

where σ_k are the diagonal elements of S_1.

Section 5 presents a deeper analysis of the role of the weighting matrix W_1.

4. THREE ALGORITHMS AND THE UNIFYING THEOREM

In this section we show how the three subspace algorithms (N4SID, MOESP and CVA) determine the system order and the matrix Γ_i from the given input–output data as stated in Theorem 1, but with different weighting matrices W_1 and W_2 for each algorithm.

In each of the following subsections we first give an outline of the algorithm. Then we prove a theorem that connects the algorithm with the unifying Theorem 1. Finally, this connection is discussed in further detail. Table 1 gives an overview of the results. Note that for each algorithm, there are two different singular-value decompositions of importance.

SVD 'Theorem 1'. This indicates the singular-value decomposition suggested by the unifying Theorem. This refers to the singular-value decomposition of (7). The matrices of this decomposition will be denoted by capitals (and the necessary sub- and superscripts):

$$U, \quad S, \quad V.$$

SVD 'Algorithm'. This points to the singular-value decomposition used by the authors of the algorithm (which can be slightly different from that of Theorem 1). The matrices of this singular value decomposition will be denoted by script letters (and the necessary sub- and superscripts):

$$\mathcal{U}, \quad \mathcal{S}, \quad \mathcal{V}.$$

4.1. *N4SID*

Description of the algorithm. The acronym N4SID stands for 'numerical algorithms for subspace state space system identification', and is pronounced like a Californian license plate: *enforce it.*

The algorithm of Van Overschee and De Moor (1994) determines the order of the system and Γ_i directly from the singular-value decomposition of the oblique projection (the superscript n stands for 'n4sid'):

$$\mathcal{O} = (\mathcal{U}_1^n \quad \mathcal{U}_2^n)\begin{pmatrix} \mathcal{S}_1^n & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} (\mathcal{V}_1^n)^T \\ (\mathcal{V}_2^n)^T \end{pmatrix}. \tag{11}$$

The order is equal to the number of non-zero singular values in $\mathcal{S}_1^n$. The observability matrix is taken equal to

$$\Gamma_i = \mathcal{U}_1^n (\mathcal{S}_1^n)^{1/2}. \tag{12}$$

The algorithm of Viberg *et al.* (1993) calculates the singular-value decomposition of $\hat{G}$ (adjusted to our notation):

$$\hat{G} = f(p^T \quad u^T)\left[\begin{pmatrix} p \\ u \end{pmatrix}(p^T \quad u^T)\right]^{-1} \times \begin{pmatrix} I_{(m+l)i} & 0 \\ 0 & 0 \end{pmatrix}\left[\begin{pmatrix} p \\ u \end{pmatrix}(p^T \quad u^T)\right]^{1/2}. \tag{13}$$

Squaring (13) and using the results of part 1 of the proof of Theorem 1 in the Appendix leads to

$$\hat{G}\hat{G}^T = [(f/u^\perp)(p/u^\perp)^T] \cdot [(p/u^\perp)(p/u^\perp)^T]^{-1} \cdot [pp^T] \cdot [(p/u^\perp)(p/u^\perp)^T]^{-1} \cdot [(p/u^\perp)(f/u^\perp)^T],$$

which is exactly equal to $\mathcal{O}\mathcal{O}^T$ (see e.g. (6)). This implies that both the algorithm of Van Overschee and De Moor (1994) and that of Viberg *et al.* (1993) calculated the same singular values $\mathcal{S}_1^n$ and the same left singular space $\mathcal{U}_1^n$, and thus determine the order n and Γ_i in exactly the same way.

Theorem 2. N4SID. The algorithms of Van Overschee and De Moor (1994) and Viberg *et al.* (1993) correspond to Theorem 1 with the weights

$$W_1 = I,$$

$$W_2 = I.$$

Proof. This is trivial (just replace W_1 and W_2 by I in the formulas of Theorem 1).

Table 1. Overview of the results for the three algorithms. The numbers refer to the corresponding formulas. W_1 and W_2 are the weighting matrices of Theorem 1, while SVD 'Theorem 1' indicates the singular-value decomposition suggested by the unifying theorem. SVD 'Algorithm' points to the singular-value decomposition used by the authors of the algorithm (which can be slightly different from that of Theorem 1). Γ_i, $\tilde{X}_i W_2$ and $\tilde{X}_i$ are also indicated for each algorithm

	Theorem 1	N4SID	MOESP	CVA
W_1	—	I	I	$[f/u^\perp(f/u^\perp)^T]^{-1/2}$
W_2	—	I	$\Pi_{u^\perp}$	$\Pi_{u^\perp}$
SVD 'Theorem 1'	(7)	(11)	(A.6)	(A.7)
SVD 'Algorithm'	—	(11)	(16)	(19)
Γ_i	(8)	(12)	(17)	(24)
$\tilde{X}_i W_2$	(9)	(14)	(18)	(20), (23)
$\tilde{X}_i$	(10)	(14)	—	(22), (25)

Discussion. A consequence of $W_2 = I$ is that we do not need (10) to determine the states $\tilde{X}_i$, but they can be determined simply from the singular-value decomposition (11) (using (9)) as

$$\tilde{X}_i = (\mathcal{S}_1^{\mathrm{n}})^{1/2}(\mathcal{V}_1^{\mathrm{n}})^{\mathrm{T}}. \tag{14}$$

In the following subsections it will become clear that for the other two algorithms, the determination of the state sequence $\tilde{X}_i$ requires the use of (10).

In Viberg *et al.*, (1993) an interpretation of this algorithm is given in an instrumental variable framework. In Van Overschee and De Moor (1994) the connection with non-steady-state Kalman filters is elaborated (see also Section 2).

Once the extended observability matrix and/or the states have been computed, the other system matrices can be computed in different ways (Fig. 1). In Viberg *et al.* (1993) the matrices A and C are calculated directly, while in Van Overschee and De Moor (1994) the states play a crucial role.

4.2. *MOESP*

Description of the algorithm. The acronym MOESP stands for 'multivariable output-error state space'. Verhaegen (1994) considered the following QR decomposition (adapted to our notation):

$$\begin{pmatrix} u \\ p \\ f \end{pmatrix} = \begin{pmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{pmatrix} \begin{pmatrix} Q_1 \\ Q_2 \\ Q_3 \end{pmatrix}. \tag{15}$$

With the singular-value decomposition (the superscript m stands for 'moesp')

$$L_{32} = (\mathcal{U}_1^{\mathrm{m}} \quad \mathcal{U}_2^{\mathrm{m}}) \begin{pmatrix} \mathcal{S}_1^{\mathrm{m}} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} (\mathcal{V}_1^{\mathrm{m}})^{\mathrm{T}} \\ (\mathcal{V}_2^{\mathrm{m}})^{\mathrm{T}} \end{pmatrix}, \tag{16}$$

Verhaegen proved that the system order can be retrieved from the number of singular values of $\mathcal{S}_1^{\mathrm{m}}$ different from zero. He also proved that a possible choice for Γ_i is

$$\Gamma_i = \mathcal{U}_1^{\mathrm{m}}(\mathcal{S}_1^{\mathrm{m}})^{1/2}. \tag{17}$$

Theorem 3. *MOESP.* The algorithm of Verhaegen (1994) corresponds to Theorem 1 with the following weights:

$$W_1 = I,$$

$$W_2 = \Pi_{u^\perp}.$$

The proof can be found in the Appendix.

Discussion. From the proof, we can conclude that applying the weights of Theorem 3 to the results of Theorem 1 leads to the MOESP algorithm of Verhaegen (1994). MOESP is thus a special case of the unifying theorem.

Note that since W_2 is not of full rank, we do not recover the full state from the singular value decomposition of $L_{33}Q_2$ (see (31)). According to (9), $L_{32}Q_2$ only recovers the projection of the state:

$$\tilde{X}_i W_2 = \tilde{X}_i \Pi_{u^\perp} = (S_1^{\mathrm{m}})^{1/2}(V_1^{\mathrm{m}})^{\mathrm{T}} = (\mathcal{S}_1^{\mathrm{m}})^{1/2}(\mathcal{V}_1^{\mathrm{m}})^{\mathrm{T}} Q_2. \tag{18}$$

The state could be determined through (10). However, since MOESP does not use state sequences, we shall not elaborate on this any further.

Once the extended observability matrix and/or the states have been computed, the other system matrices can be computed in different ways (Fig. 1).

4.3. *CVA*

Description of the algorithm. Larimore (1990) considered the canonical correlation between, on the one hand, the past (p) conditional to the future inputs (u) and on the other hand the future outputs (f) conditional to the future inputs (u). In formulas, this means that he considered the principal angles and directions between $p/u^\perp$ and $f/u^\perp$. He denoted his class of algorithms by CVA, which stands for 'canonical variate analysis'.

Consider the singular-value decomposition of $\mathcal{M}^{\mathrm{c}}$ (the superscript c stands for 'cva'):

$$\begin{aligned} \mathcal{M}^{\mathrm{c}} &\stackrel{\mathrm{def}}{=} [(f/u^\perp)(f/u^\perp)^{\mathrm{T}}]^{-1/2}[(f/u^\perp)(p/u^\perp)^{\mathrm{T}}] \\ &\quad \times [(p/u^\perp)(p/u^\perp)^{\mathrm{T}}]^{-1/2} \\ &= (\mathcal{U}_1^{\mathrm{c}} \quad \mathcal{U}_2^{\mathrm{c}}) \begin{pmatrix} \mathcal{S}_1^{\mathrm{c}} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} (\mathcal{V}_1^{\mathrm{c}})^{\mathrm{T}} \\ (\mathcal{V}_2^{\mathrm{c}})^{\mathrm{T}} \end{pmatrix}. \end{aligned} \tag{19}$$

From Pal (1982), we then find that the cosines of the principal angles between $p/u^\perp$ and $f/u^\perp$ are given by the elements of $\mathcal{S}_1^{\mathrm{c}}$ and that the principal directions α in $p/u^\perp$ and β in $f/u^\perp$ are given by

$$\alpha = L_\alpha(p/u^\perp), \tag{20}$$

$$\beta = L_\beta(f/u^\perp), \tag{21}$$

with

$$L_\alpha = (\mathcal{S}_1^{\mathrm{c}})^{1/2}(\mathcal{V}_1^{\mathrm{c}})^{\mathrm{T}}[(p/u^\perp)(p/u^\perp)^{\mathrm{T}}]^{-1/2},$$

$$L_\beta = (\mathcal{S}_1^{\mathrm{c}})^{1/2}(\mathcal{U}_1^{\mathrm{c}})^{\mathrm{T}}[(f/u^\perp)(f/u^\perp)^{\mathrm{T}}]^{-1/2}.$$

Larimore (1990) determined the order from the number of principal angles different from $\frac{1}{2}\pi$. He did not use the extended observability

matrix explicitly (as in N4SID and MOESP), but instead he worked with the states (which he called 'memory'). The way we understand it, he defined this 'memory' ($\tilde{\mathscr{X}}_i^c$) by applying the linear combination L_α to the past p:

$$\tilde{\mathscr{X}}_i^c \stackrel{\text{def}}{=} L_\alpha p. \tag{22}$$

Note that this 'memory' is not equal to α (the principal directions in the projected past (20)). We shall show in the sequel that these principal directions α are *projected* states, and thus cannot be used as states.

Theorem 4. *CVA*. The algorithm of Larimore (1990) corresponds to Theorem 1 with the following weights:

$$W_1 = [(f/u^\perp)(f/u^\perp)^T]^{-1/2},$$
$$W_2 = \Pi_{u^\perp}.$$

The proof can be found in the Appendix.

Discussion. From Theorem 4, we conclude that applying the weighting matrices of Theorem 4 to the results of Theorem 1 leads to a principal direction analysis between the past inputs and outputs orthogonalized to the future inputs ($p/u^\perp$) and the future outputs orthogonalized to the future inputs ($f/u^\perp$). The same principal angles are calculated as in Larimore (1990).

We shall now investigate further similarities. The formula (9) shows how we can partially reconstruct the states. In part 2 of the proof of the theorem in the Appendix it is shown that the (partial) states recovered from (9) are equal to the principal directions α. This proves formally that the principal directions are no states, but projected states:

$$\alpha = \tilde{X}_i \Pi_{u^\perp}. \tag{23}$$

The 'full' state sequence is given by (10) of Theorem 1. First Γ_i is determined (for the weights of Theorem 4) as

$$\Gamma_i = [(f/u^\perp)(f/u^\perp)^T]^{1/2} U_1^c (S_1^c)^{1/2}. \tag{24}$$

Then we find, with (10),

$$\tilde{X}_i^c = \Gamma_i^\dagger \mathscr{O}. \tag{25}$$

In the Appendix (proof of the equivalent states) it is shown that the state used by Larimore ($\tilde{\mathscr{X}}_i^c$ as defined in (22)) is exactly the same as $\tilde{X}_i^c$ of (25), and is thus indeed a valid state sequence.

An additional point concerning the CVA method is that the weighting used in this method is optimal for state order determination from finite sample sizes. This has been shown by example by Larimore (1994).

A last remark concerns the sensitivity to scaling. While the two algorithms N4SID and MOESP are sensitive to scaling of the inputs and/or outputs, the CVA algorithm is insensitive. This is because only angles and normalized directions are considered in the CVA algorithm.

Note that the general framework proposed by Larimore (1990) corresponds to a generalization of principal directions and angles. In this framework, the matrix

$$W_1 = [(f/u^\perp)(f/u^\perp)^T]^{-1/2}$$

is replaced by another weighting matrix

$$W_1 = \Lambda^{-1/2},$$

which still falls into the unifying approach of Theorem 1. For a further interpretation of W_1, see Section 5.

Once the extended observability matrix and/or the states have been computed, the other system matrices can be computed in different ways (Fig. 1).

4.4. *Other algorithms*

It should be clear by now that the three published algorithms discussed so far do not exclude other variations on the same theme. Different weighting functions will allow for a different determination of Γ_i and thus a different algorithm. As a simple example of such an extension, we mention a possible introduction of W_1 in the MOESP and/or the N4SID scheme. Another important note is that different algorithms calculate the same result (up to within a similarity transform) whenever the exact system order n is chosen and the number of data points goes to infinity (since all algorithms are asymptotically unbiased). However, the differences between the algorithms (and between the different weightings) become clear under the following conditions.

1. When the order that is chosen is different from the exact order, this implies model reduction (note also that a lot of real life processes are infinite-dimensional, which implies that the order is always underestimated). The distribution of this 'bias error' will be different for each algorithm (for each weighting). The characterization of this error as a function of the weighting matrices W_1 and W_2 is discussed in Van Overschee and De Moor (1995).
2. The number of data points is finite. This raises the question of the large sample (asymptotic) variance on the result. This question was already partially solved by Viberg *et al.* (1991, 1993). Continuing efforts are underway by these authors to get more insight into the variance distribution.

5. HOW TO CHOOSE THE WEIGHT W_1

In this section we shall outline an intuitive way to choose the weighting matrix W_1. The matrix is constructed from the impulse response of a shaping filter.

The weight W_1 was introduced in the complexity reduction optimization problem (5):

$$\min_{\mathcal{R}} \|W_1(\mathcal{O} - \mathcal{R})W_2\|_F^2$$

$$\text{constrained by rank } (\mathcal{R}) = n.$$

To get a better understanding of the meaning of this optimization problem, consider the first column of $\mathcal{O}$. Since we know (see the proof of Theorem 1 in the Appendix) that $\mathcal{O} = \Gamma_i \tilde{X}_i$, we find that this first column (τ_1) is equal to:

$$\tau_1 = \Gamma_i \tilde{x}_i = \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{i-1} \end{pmatrix} \tilde{x}_i.$$

τ_1 is thus equal to the autonomous response of the system (without noise) to an initial state $\tilde{x}_i$. In this way, each column τ_k of $\mathcal{O}$ can be considered as the autonomous response to the state $\tilde{x}_{k+i+1}$. The states $\tilde{x}_k$, on the other hand, are the states that are reached when an input u_k is applied to the system. $\mathcal{O}$ thus contains the autonomous responses to initial states that are typically reached when the input u_k is applied to the system. It seems meaningful to try to keep these responses as intact as possible. Thus the minimization problem is

$$\min_{\mathcal{R}} \|\mathcal{O} - \mathcal{R}\|_F^2,$$

$$\text{constrained by rank } (\mathcal{R}) = n.$$

We are now ready to interpret the weight W_1. Suppose that a shaping filter $W_1(z)$ is given. This filter has a large amplitude at frequency ranges where the system has to be modeled well, and vice versa:

$$W_1(z) = C_w(zI - A_w)^{-1}B_w + D_w.$$

The weighting matrix W_1 is now formed as

$$W_1 = \begin{pmatrix} D_w & 0 & 0 & \dots & 0 \\ C_wB_w & D_w & 0 & \dots & 0 \\ C_wA_wB_w & C_wB_w & D_w & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ C_wA_w^{i-2}B_w & C_wA_w^{i-3}B_w & C_wA_w^{i-4}B_w & \dots & D_w \end{pmatrix}.$$

Consider the first column of $W_1\mathcal{O}$:

$$W_1\tau_1 = W_1\Gamma_i\tilde{x}_i =$$

$$\begin{pmatrix} D_w & 0 & 0 & \dots & 0 \\ C_wB_w & D_w & 0 & \dots & 0 \\ C_wA_wB_w & C_wB_w & D_w & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ C_wA_w^{i-2}B_w & C_wA_w^{i-3}B_w & C_wA_w^{i-4}B_w & \dots & D_w \end{pmatrix} \times \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{i-1} \end{pmatrix} \tilde{x}_i.$$

It is clear that the elements of $W_1\tau_1$ are the first i components of the filtered sequence τ_1 (filter $W_1(z)$). This implies that the minimization problem

$$\min_{\mathcal{R}} \|W_1[\mathcal{O} - \mathcal{R}]\|_F^2,$$

$$\text{constrained by rank } (\mathcal{R}) = n,$$

will approximate the *filtered* autonomous responses of the states $\tilde{x}_k$ as well as possible. Thus if $W_1(z)$ is a low-pass filter then the approximation will be good at low frequencies and bad at high frequencies. This specific form of the weighting matrix W_1 thus introduces the freedom to shape the error in the frequency domain.

Van Overschee and De Moor (1995) show that this weight W_1 can be interpreted in the framework of the weighted model reduction of Enns (1984).

6. EXAMPLE

We illustrate the differences between the different algorithms and choices of weights with a simulation example. Consider the fourth-order system (in forward innovation form)

$$x_{k+1} = \begin{pmatrix} 0.67 & 0.67 & 0 & 0 \\ -0.67 & 0.67 & 0 & 0 \\ 0 & 0 & -0.67 & -0.67 \\ 0 & 0 & 0.67 & -0.67 \end{pmatrix} x_k + \begin{pmatrix} 0.6598 \\ 1.9698 \\ 4.3171 \\ -2.6436 \end{pmatrix} u_k + \begin{pmatrix} -0.1027 \\ 0.5501 \\ 0.3545 \\ -0.5133 \end{pmatrix} e_k,$$

$$y_k = (-0.5749 \quad 1.0751 \quad -0.5225 \quad 0.1830)x_k - (0.7139)u_k + (0.9706)e_k.$$

The input u_k is a white Gaussian noise process

with variance 1. The innovation e_k is also a white Gaussian noise sequence with variance 9. This leads to a signal-to-noise ratio (of variances) at the output of 2.2. We consider experiments with 100 samples. The number of block rows i is chosen equal to 10. Two different series of Monte Carlo experiments are performed. Each series consists of 100 experiments. For each experiment, a new innovation sequence e_k is applied (the input u_k is the same over the 100 experiments). For every experiment, the extended observability matrix Γ_i is determined. The estimated matrix A is determined as:

$$A = (\underline{\Gamma}_i)^\dagger \cdot \overline{\Gamma}_i,$$

where $\underline{\Gamma}_i$ is equal to Γ_i without the last l (number of outputs) rows. Similarly, $\overline{\Gamma}_i$ is equal to Γ_i without the first l rows. The eigenvalues of the matrix A are calculated and plotted.

Different algorithms. In this first Monte Carlo experiment, we have tested the three different algorithms considered in this paper. Figure 3 shows the results.

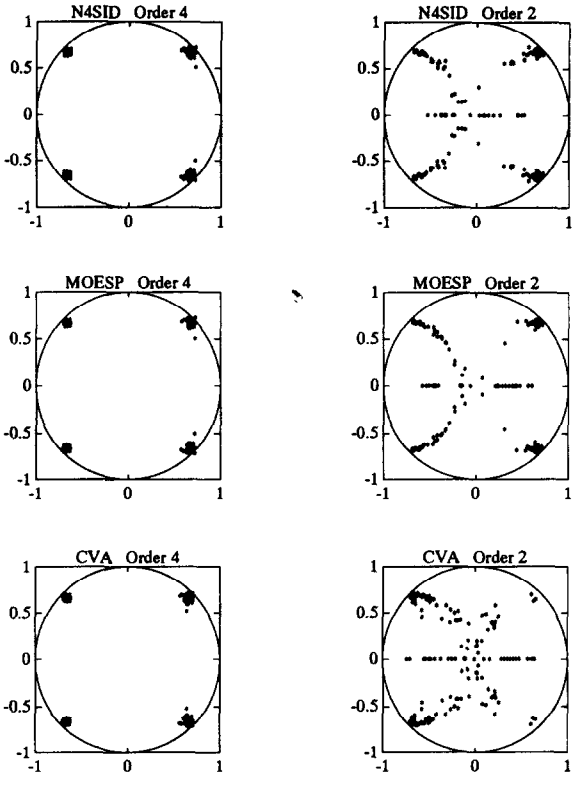

Fig. 3. Plot of the eigenvalues of the estimates of A. The rows correspond to the different algorithms (N4SID, MOESP and CVA), while the columns correspond to different orders of the identified system (4 and 2). The fourth-order system (first column) is identified equally well for the three algorithms. The second-order approximation (second column) turns out to have the smallest variance for the CVA algorithm. This observation has been made in other simulation studies, but could not be proved.

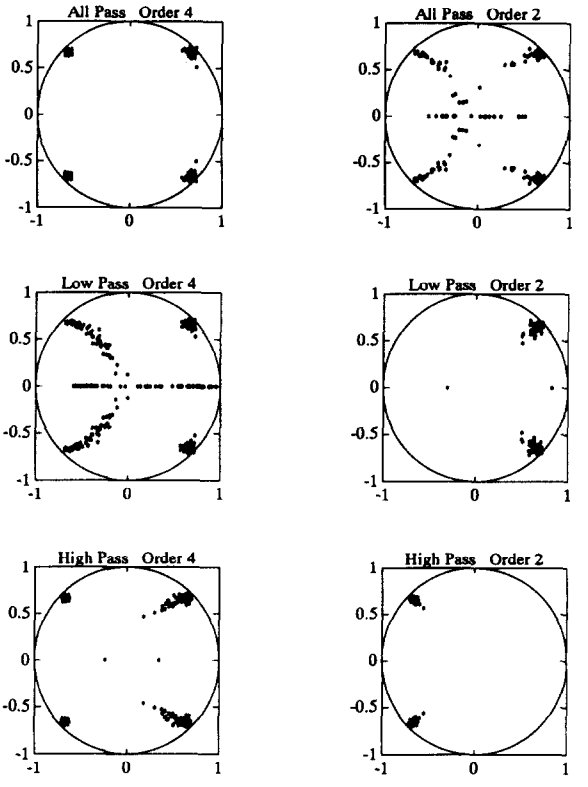

Fig. 4. Plot of the eigenvalues of the estimates of A. The rows correspond to different filters $W_1(z)$ (all-pass, low-pass and high-pass), while the columns correspond to different orders of the identified system (4 and 2). The weight $W_2 = I$. The fourth-order system (first column) identifies the four poles, but the variance of the poles is the smallest where the amplitude of the filter is the largest (large variance for the high-frequency poles when the low-pass filter is used, and vice versa). The second-order approximation is clearly the best where the amplitude of the shaping filter is the largest, as was predicted in Section 5.

Different weights W_1. In this second Monte Carlo experiment we have tested the use of three different weights $W_1(z)$: an all-pass weight, a low-pass weight (a sixth-order Butterworth filter, with cut-off frequency equal to half the Nyquist frequency) and a high-pass weight (a sixth-order Butterworth filter, with cut-off frequency half the Nyquist frequency). W_2 was for each of the cases equal to the identity. Figure 4 shows the results.

7. CONCLUSIONS

We have introduced a unifying theorem that allows the extraction of the system order and the extended observability matrix from given input–output data. The theorem succeeds in unifying three algorithms described in the literature. By the introduction of weighting matrices, the approximation of the system can be shaped in the frequency domain.

Acknowledgments—We should like to thank Mats Viberg for many interesting discussions on the topics treated in this paper. Peter Van Overschee is a research assistant and Bart De Moor a senior research associate of the Belgian National

Fund for Scientific Research. This paper presents research results of the Belgian Programme on Interuniversity Poles of Attraction (IUAP 17 and 50), initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture. The scientific responsibility rests with its authors.

REFERENCES

Akaike, H. (1975). Markovian representation of stochastic processes by canonical variables. *SIAM J. Control,* **13,** 162–173.

Enns, D. (1984). Model reduction for control system design. PhD dissertation, Depart of Aeronautics and Astronautics, Stanford University.

Kailath, T. (1980). *Linear Systems.* Prentice-Hall, Englewood Cliffs, NJ.

Kung, S. Y. (1978). A new identification and model reduction algorithm via singular value decomposition. In *Proc. 12th Asilomar Conf. on Circuits, Systems and Computers,* Pacific Grove, CA, pp. 705–714.

Larimore, W. E. (1990). Canonical variate analysis in identification, filtering, and adaptive control. In *Proc. 29th IEEE Conf. on Decision and Control,* Honolulu, Hawaii, pp. 596–604.

Larimore, W. E. (1994). The optimality of canonical variate identification by example. In *Proc. 10th IFAC Symp. on System Identification,* Copenhagen, 4–6 July, Vol. 2, pp. 151–156.

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

Pal, D. (1982). Balanced stochastic realization and model reduction. Master's thesis, Washington State University, Electrical Engineering.

Paulraj, A., R. Roy and T. Kailath (1986). A subspace rotation approach to signal parameter estimation. *Proc. IEEE,* **74,** 1044–1045.

Söderström, T. and P. Stoica (1989). *System Identification.* Prentice-Hall, Englewood Cliffs, NJ.

Van Overschee, P. and B. De Moor (1993). Subspace algorithms for the stochastic identification problem. *Automatica,* **29,** 649–660.

Van Overschee, P. and B. De Moor (1994). N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica,* **30,** 75–93.

Van Overschee, P. and B. De Moor (1995). Choice of state-space basis in combined deterministic–stochastic subspace identification. *Automatica,* **31,** 1877–1883.

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., B. Ottersten, B. Wahlberg and L. Ljung (1991). A statistical perspective on state-space modeling using subspace methods. In *Proc. 30th IEEE Conf. on Decision and Control,* Brighton, England, Vol. 2, pp. 1337–1342.

Viberg, M., B. Ottersten, B. Wahlberg and L. Ljung (1993). Performance of subspace based state space system identification methods. In *Proc. 12th IFAC World Congress,* Sydney, Australia, 18–23 July, Vol. 7, pp. 369–372.

APPENDIX—PROOFS

Proof of the unifying Theorem 1

Part 1. In this first part we prove (6). First note that the matrix $(p^T\ u^T)^T$ is of full rank. This is because the input u_k is persistently exciting of order $2i$ (assumption 3 of the theorem) and because the noise sequences v_k and w_k are not identically zero (assumption 1 of the theorem).

The solution to the minimization problem (4),

$$\min_{L_p, L_u} \left\| f - (L_p \quad L_u)\begin{pmatrix} p \\ u \end{pmatrix} \right\|_F^2,$$

can thus be written as

$$(L_p \quad L_u) = f(p^T \quad u^T)\left[\begin{pmatrix} p \\ u \end{pmatrix}(p^T \quad u^T)\right]^{-1}. \qquad (A.1)$$

Through the matrix inversion lemma (Kailath, 1980), (A.1) can be rewritten as

$$(L_p \quad L_u) = f(p^T \quad u^T)\begin{pmatrix} \Delta^{-1} & -\Delta^{-1}pu^T(uu^T)^{-1} \\ -(uu^T)^{-1}up^T\Delta^{-1} & [uu^T - up^T(pp^T)^{-1}pu^T]^{-1} \end{pmatrix},$$

with

$$\Delta = p[I_j - u^T(uu^T)^{-1}u]p^T.$$

This implies that

$$L_p = f(I_j - u^T(uu^T)^{-1}u)p^T\Delta^{-1}$$
$$= f[I_j - u^T(uu^T)^{-1}u]p^T\{p[I_j - u^T(uu^T)^{-1}u]p^T\}^{-1}. \qquad (A.2)$$

We also know that

$$\Pi_{u^\perp} = I_j - u^T(uu^T)^{-1}u,$$

and that this projection operator is idempotent ($\Pi_{u^\perp}\Pi_{u^\perp} = \Pi_{u^\perp}$) and symmetric. We can thus transform (A.2) to

$$L_p = f\Pi_{u^\perp}p^T \cdot (p\Pi_{u^\perp}p^T)^{-1}$$
$$= f\Pi_{u^\perp}\Pi_{u^\perp}^T p^T \cdot (p\Pi_{u^\perp}\Pi_{u^\perp}^T p^T)^{-1}$$
$$= (f\Pi_{u^\perp})(p\Pi_{u^\perp})^T \cdot [(p\Pi_{u^\perp})(p\Pi_{u^\perp})^T]^{-1}$$
$$= (f/u^\perp)(p/u^\perp)^T \cdot [(p/u^\perp)(p/u^\perp)^T]^{-1},$$

which proves (6):

$$\mathcal{O} = L_p p$$
$$= (f/u^\perp)(p/u^\perp)^T \cdot [(p/u^\perp)(p/u^\perp)^T]^{-1}p.$$

It should also be clear that $L_p p$ is the oblique projection of the row space of f along the row space of u on the row space of p, since it is the part of the row space of f that lies along the row space of p when the row space of f is projected (in a least-squares sense) on the row space of u and p.

Part 2. In Van Overschee and De Moor (1994) it is proved that, under conditions 1–4 of the theorem, we have

$$\mathcal{O} = \Gamma_i \tilde{X}_i.$$

This implies that the column space of $\mathcal{O}$ is equal to the column space of Γ_i.

Part 3. With (7), we easily find that the best 'unstructured' rank-n approximation (in the Frobenius norm) of $W_1\mathcal{O}W_2$ is equal to $U_1S_1V_1^T$ (problem (5)). We thus have

$$W_1\mathcal{O}W_2 = W_1\Gamma_i\tilde{X}_iW_2 = U_1S_1V_1^T = W_1\mathcal{R}W_2. \qquad (A.3)$$

(a) The rank of the second part of (A.3) is equal to n, since Γ_i has only n columns and $\tilde{X}_i$ has only n rows (and since W_1 is of full rank and rank (p) = rank (pW_2)—assumption 5 of the theorem 1). This proves claim (a) of the theorem.

(b) From (A.3), we know that

$$W_1\mathcal{R}W_2 = U_1S_1V_1^T.$$

From assumption 5 of the theorem, we know that W_1 is of full rank. Assume also that rank $(W_2) = \rho \le j$. We then find that the general solution $\mathcal{R}$ of the optimization problem (5) can be written as

$$\mathcal{R} = W_1^{-1}U_1S_1V_1^T(W_2^\dagger + ZW_2^\perp),$$

where $Z \in \mathbb{R}^{j\times(j-\rho)}$ is an arbitrary matrix and $W_2^\perp \in \mathbb{R}^{(j-\rho)\times j}$ is the orthogonal complement of the row space of W_2: $W_2^\perp W_2 = 0$ and rank $(W_2^\perp) = j - \rho$. The solution $\mathcal{R}$ with minimal Frobenius norm can be found by putting $Z = 0$:

$$\mathcal{R} = W_1^{-1}U_1S_1V_1^TW_2^\dagger.$$

This proves claim (b) of the theorem.

(c, d) We can split (A.3) into two parts:

$$W_1\Gamma_i = U_1 S_1^{1/2}, \tag{A.4}$$

$$\tilde{X}_i W_2 = S_1^{1/2} V_1^T. \tag{A.5}$$

Equation (A.4) implies that ((8) or claim (c) of the theorem)

$$\Gamma_i = W_1^{-1} U_1 S_1^{1/2}.$$

Equation (A.5) is claim (d) of the theorem. Note that, when splitting (A.3) into (A.4) and (A.5), a similarity transformation T is easily introduced, which explains the parenthetical notes in the theorem.

(e) Since $\mathcal{O} = \Gamma_i \tilde{X}_i$, we can recover the 'full' state sequence (not projected on the column space of W_2 as in (A.5)) as

$$\tilde{X}_i = \Gamma_i^{\dagger} \mathcal{O},$$

which is exactly claim (e) of the theorem.

Proof of Theorem 3 (*MOESP*)

With the QR decomposition of (15), we find that

$$p/u^{\perp} = L_{22} Q_2$$

$$f/u^{\perp} = (L_{32} \quad L_{33}) \begin{pmatrix} Q_2 \\ Q_3 \end{pmatrix}.$$

Note also that

$$Q_2 Q_2^T = I_{(l+m)i},$$

$$Q_2 Q_3^T = 0,$$

$$\text{rank}\,(L_{22}) = (l+m)i.$$

With the weights of Theorem 3 and (7), we find that the matrix from which we should take the singular-value decomposition according to Theorem 1 is

$$\begin{aligned}
&[(f/u^{\perp})(p/u^{\perp})^T][(p/u^{\perp})(p/u^{\perp})^T]^{-1} p\Pi_{u^{\perp}} \\
&= [(f/u^{\perp})(p/u^{\perp})^T][(p/u^{\perp})(p/u^{\perp})^T]^{-1}(p/u^{\perp}) \\
&= \left[(L_{32} \quad L_{33}) \begin{pmatrix} Q_2 \\ Q_3 \end{pmatrix} Q_2^T L_{22}^T\right](L_{22} Q_2 Q_2^T L_{22}^T)^{-1} L_{22} Q_2 \\
&= (L_{32} L_{22}^T)(L_{22} L_{22}^T)^{-1} L_{22} Q_2 \\
&= L_{32} L_{22}^T (L_{22}^T)^{-1} L_{22}^{-1} L_{22} Q_2 \\
&= L_{32} Q_2 \\
&= (U_1^m \quad U_2^m) \begin{pmatrix} S_1^m & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} (V_1^m)^T \\ (V_2^m)^T \end{pmatrix}.
\end{aligned} \tag{A.6}$$

Equation (16) takes the singular-value decomposition of L_{32}, while (A.6) takes the singular-value decomposition of $L_{32} Q_2$. Since Q_2 is orthonormal ($L_{32} L_{32}^T = L_{32} Q_2 Q_2^T L_{32}^T$), we find that the left singular vectors and the singular values of both matrices are equal:

$$\mathcal{U}_1^m = U_1^m,$$

$$\mathcal{S}_1^m = S_1^m.$$

Proof of Theorem 4 (*CVA*)

Part 1. With the weights of Theorem 4, we find from Theorem 1 ((7), which we denote be M^c)

$$\begin{aligned}
M^c &= [(f/u^{\perp})(f/u^{\perp})^T]^{-1/2}[(f/u^{\perp})(p/u^{\perp})^T] \\
&\quad \times [(p/u^{\perp})(p/u^{\perp})^T]^{-1} p\Pi_{u^{\perp}} \\
&= [(f/u^{\perp})(f/u^{\perp})^T]^{-1/2}[(f/u^{\perp})(p/u^{\perp})^T] \\
&\quad \times [(p/u^{\perp})(p/u^{\perp})^T]^{-1} p/u^{\perp} \\
&= (U_1^c \quad U_2^c) \begin{pmatrix} S_1^c & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} (V_1^c)^T \\ (V_2^c)^T \end{pmatrix}.
\end{aligned} \tag{A.7}$$

Comparison of $\mathcal{M}^c$ of (19) (first line) and M^c of (A.7) (second line) leads to the conclusion

$$\mathcal{M}^c (\mathcal{M}^c)^T = M^c (M^c)^T.$$

This implies that the left singular vectors and singular values of both matrices are equal:

$$\mathcal{U}_1^c = U_1^c, \tag{A.8}$$

$$\mathcal{S}_1^c = S_1^c. \tag{A.9}$$

We conclude that, with the weights of Theorem 4 applied to Theorem 1, we can calculate the principal angles between $p/u^{\perp}$ and $f/u^{\perp}$.

Part 2. A second observation is that the principal directions α in $p/u^{\perp}$ are exactly equal to $(S_1^c)^{1/2}(V_1^c)^T$. This is because (with (A.8) and (A.9))

$$\begin{aligned}
&M^c = \mathcal{M}^c[(p/u^{\perp})(p/u^{\perp})^T]^{-1/2}(p/u^{\perp}) \\
&\Downarrow \\
&U_1^c (S_1^c)^{1/2} \cdot [(S_1^c)^{1/2}(V_1^c)^T] = \mathcal{U}_1^c (\mathcal{S}_1^c)^{1/2} \cdot [(\mathcal{S}_1^c)^{1/2}(\mathcal{V}_1^c)^T \\
&\qquad \times [(p/u^{\perp})(p/u^{\perp})^T]^{-1/2}(p/u^{\perp})] \\
&\Downarrow \\
&(S_1^c)^{1/2}(V_1^c)^T = (\mathcal{S}_1^c)^{1/2}(\mathcal{V}_1^c)^T [(p/u^{\perp})(p/u^{\perp})^T]^{-1/2}(p/u^{\perp}) \\
&\Downarrow \\
&(S_1^c)^{1/2}(V_1^c)^T = \alpha.
\end{aligned} \tag{A.10}$$

Proof of the equivalent states

We prove that the states of (22) are exactly equal to the states of (25):

$$\begin{aligned}
&(\mathcal{S}_1^c)^{1/2}(\mathcal{V}_1^c)^T = (\mathcal{S}_1^{c\,1/2}(\mathcal{V}_1^c)^T \\
&\Downarrow \\
&(\mathcal{S}_1^c)^{1/2}(\mathcal{V}_1^c)^T = (\Gamma_i^{\dagger})(\Gamma_i)(\mathcal{S}_1^c)^{1/2}(\mathcal{V}_1^c)^T \\
&\Downarrow \\
&(\mathcal{S}_1^c)^{1/2}(\mathcal{V}_1^c)^T = \Gamma_i^{\dagger}[(f/u^{\perp})(f/u^{\perp})^T]^{1/2} \\
&\qquad \times U_1^c (S_1^c)^{1/2}(\mathcal{S}_1^c)^{1/2}(\mathcal{V}_1^c)^T \\
&\Downarrow \\
&(\mathcal{S}_1^c)^{1/2}(\mathcal{V}_1^c)^T = \Gamma_i^{\dagger}[(f/u^{\perp})(f/u^{\perp})^T]^{1/2} \\
&\qquad \times \mathcal{U}_1^c \mathcal{S}_1^c (\mathcal{V}_1^c)^T \\
&\Downarrow \\
&(\mathcal{S}_1^c)^{1/2}(\mathcal{V}_1^c)^T = \Gamma_i^{\dagger}[(f/u^{\perp})(f/u^{\perp})^T]^{1/2} \\
&\qquad \times [(f/u^{\perp})(f/u^{\perp})^T]^{-1/2} \\
&\qquad \cdot [(f/u^{\perp})(p/u^{\perp})^T] \\
&\qquad \times [(p/u^{\perp})(p/u^{\perp})^T]^{-1/2} \\
&\Downarrow \\
&(\mathcal{S}_1^c)^{1/2}(\mathcal{V}_1^c)^T [(p/u^{\perp})(p/u^{\perp})^T]^{-1/2} p = \Gamma_i^{\dagger}[(f/u^{\perp})(p/u^{\perp})^T] \\
&\qquad \times [(p/u^{\perp})(p/u^{\perp})^T]^{-1} p \\
&\Downarrow \\
&(\mathcal{S}_1^c)^{1/2}(\mathcal{V}_1^c)^T [(p/u^{\perp})(p/u^{\perp})^T]^{-1/2} p = \Gamma_i^{\dagger} \mathcal{O} \\
&\Downarrow \\
&\tilde{\mathcal{X}}_i^c = \tilde{X}_i^c.
\end{aligned}$$