Available online at www.sciencedirect.com
SCIENCE DIRECT®

ELSEVIER

Automatica 41 (2005) 755–766

automatica

www.elsevier.com/locate/automatica

Algorithms for deterministic balanced subspace identification[☆]

Ivan Markovsky[a,*], Jan C. Willems[a], Paolo Rapisarda[b], Bart L.M. De Moor[a]

[a]ESAT, SCD-SISTA, K.U.Leuven, Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
[b]Department of Mathematics, University of Maastricht, 6200 MD Maastricht, The Netherlands

Received 22 January 2004; received in revised form 7 October 2004; accepted 21 October 2004
Available online 13 January 2005

Abstract

New algorithms for identification of a balanced state space representation are proposed. They are based on a procedure for the estimation of impulse response and sequential zero input responses directly from data. The proposed algorithms are more efficient than the existing alternatives that compute the whole Hankel matrix of Markov parameters. It is shown that the computations can be performed on Hankel matrices of the input–output data of various dimensions. By choosing wider matrices, we need persistency of excitation of smaller order. Moreover, this leads to computational savings and improved statistical accuracy when the data is noisy. Using a finite amount of input–output data, the existing algorithms compute finite time balanced representation and the identified models have a lower bound on the distance to an exact balanced representation. The proposed algorithm can approximate arbitrarily closely an exact balanced representation. Moreover, the finite time balancing parameter can be selected automatically by monitoring the decay of the impulse response. We show what is the optimal in terms of minimal identifiability condition partition of the data into “past” and “future”.

Keywords: Exact deterministic subspace identification; Balanced model reduction; Approximate system identification; MPUM

1. Introduction

In this paper, we consider the following exact deterministic identification problem: given an input–output trajectory $\tilde{w} = (\tilde{u}, \tilde{y})$, $\tilde{w} = (\tilde{w}(1), \ldots, \tilde{w}(T))$, of an LTI system

$$\mathscr{S}: \quad \begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{1}$$

$u(t) \in \mathbb{R}^{\mathtt{m}}$, $y(t) \in \mathbb{R}^{\mathtt{p}}$, $x(t) \in \mathbb{R}^{\mathtt{n}}$, determine from $\tilde{w}$ an associated balanced state model (Moore, 1981; Pernebo & Silverman, 1982)

$$\mathscr{S}_{\text{bal}}: \quad \begin{aligned} x_{\text{bal}}(t+1) &= A_{\text{bal}}x_{\text{bal}}(t) + B_{\text{bal}}u(t) \\ y(t) &= C_{\text{bal}}x_{\text{bal}}(t) + D_{\text{bal}}u(t). \end{aligned}$$

The given trajectory $\tilde{w} = (\tilde{u}, \tilde{y})$ is an *exact* trajectory of $\mathscr{S}$. This means that there exists $\tilde{x}(1) \in \mathbb{R}^{\mathtt{n}}$, such that the response of $\mathscr{S}$ to the input $\tilde{u}$ and initial condition $\tilde{x}(1)$ is $\tilde{y}$. The problem is to find conditions and algorithms to construct $\mathscr{S}_{\text{bal}}$ directly from $\tilde{w}$. Although the assumption that $\tilde{w}$ is exact is mainly of theoretical importance, we believe that solving the exact identification problem is a prerequisite for the study of the realistic *approximate* identification problems.

Exact state-space identification has been considered earlier in Gopinath (1969), Budin (1971) and later on in the behavioral setting in Willems (1986). The modern approach is deterministic subspace identification (Van Overschee & De Moor, 1996, Chapter 2). The identified model need not be in a special state space basis. The problem considered in this paper is to derive a balanced model and the motivation is that in a balanced basis one can apply truncation as a very effective heuristic for model reduction, which yields a method for approximate identification. Thus, exact identification of a balanced model is a prelude to approximate identification. In the language of Willems (1986), we want

[☆] This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor B. Ninness under the direction of Editor T. Soderstrom.

[*] Corresponding author. Fax: +32-16-321970.

E-mail addresses: Ivan.markovsky@esat.kuleuven.ac.be (I. Markovsky), jan.willems@esat.kuleuven.ac.be (J.C. Willems), bart.demoor@esat.kuleuven.ac.be (P. Rapisarda), p.rapisarda@math.unimaas.nl (B.L.M. De Moor).

0005-1098/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2004.10.007

to find an input/state/output balanced representation of the *most powerful unfalsified model* (*MPUM*).

The balanced state-space identification problem is studied in Moonen and Ramos (1993) and Van Overschee and De Moor (1996, Chapter 5). The proposed algorithms fit in the outline given below, which will be called the *basic algorithm*. The following notation is used: with $f = (f(1), \dots, f(T))$,

$$\mathscr{H}_\Delta(f) := \begin{bmatrix} f(1) & f(2) & f(3) & \cdots & f(T-\Delta+1) \\ f(2) & f(3) & f(4) & \cdots & f(T-\Delta+2) \\ f(3) & f(4) & f(5) & \cdots & f(T-\Delta+3) \\ \vdots & \vdots & \vdots & & \vdots \\ f(\Delta) & f(\Delta+1) & f(\Delta+2) & \cdots & f(T) \end{bmatrix}$$

and σ is the shift operator $\sigma f(t) := f(t+1)$. Acting on a vector or matrix, σ removes the first block-row. By f we will denote both the time series $(f(1), \dots, f(T))$ and the vector $\mathrm{col}(f(1), \dots, f(T))$, where $\mathrm{col}(\cdot)$ denotes a (block) column vector.

Algorithm 1 (*Basic algorithm*). *Input*: a time series $\tilde{w} = (\tilde{u}, \tilde{y})$, an upper bound $\mathsf{n}_{\max}$ of the system order, and a finite time balancing parameter $\Delta > \mathsf{n}_{\max}$.

1. Find the first 2Δ samples $H(0), \dots, H(2\Delta-1)$ of the impulse response matrix of $\mathscr{S}$ and let $H := \mathrm{col}(H(0), \dots, H(2\Delta-1))$.
2. Find zero input responses $y_0^{(1)}, \dots, y_0^{(M)}$, $M := T - \Delta + 1$, of length Δ, generated from initial conditions $x_0^{(1)}, \dots, x_0^{(M)}$ that form a valid state sequence of $\mathscr{S}$ obtained by the input $\tilde{u}$. Let $Y_0 := [y_0^{(1)} \ \cdots \ y_0^{(M)}]$.
3. Compute the restricted SVD, $\mathfrak{H} = U\Sigma V^\top$, of the block Hankel matrix of Markov parameters $\mathfrak{H} = \mathscr{H}_\Delta(\sigma H) \in \mathbb{R}^{\Delta\mathtt{p}\times\Delta\mathtt{m}}$.
4. Compute the balanced state sequence $\tilde{X}_{\mathrm{bal}} := \sqrt{\Sigma^{-1}}U^\top Y_0$,

   $$\tilde{X}_{\mathrm{bal}} = [\tilde{x}_{\mathrm{bal}}(\mathsf{n}_{\max}+1) \quad \cdots \quad \tilde{x}_{\mathrm{bal}}(\mathsf{n}_{\max}+M)].$$

5. Compute the finite time Δ balanced realization A_{bal}, B_{bal}, C_{bal}, D_{bal} by solving the linear system of equations

   $$\begin{bmatrix} \tilde{x}_{\mathrm{bal}}(\mathsf{n}_{\max}+2)) & \cdots & \tilde{x}_{\mathrm{bal}}(\mathsf{n}_{\max}+M) \\ \tilde{y}(\mathsf{n}_{\max}+1) & \cdots & \tilde{y}(\mathsf{n}_{\max}+T-\Delta) \end{bmatrix} = \begin{bmatrix} A_{\mathrm{bal}} & B_{\mathrm{bal}} \\ C_{\mathrm{bal}} & D_{\mathrm{bal}} \end{bmatrix} \times \begin{bmatrix} \tilde{x}_{\mathrm{bal}}(\mathsf{n}_{\max}+1) & \cdots & \tilde{x}_{\mathrm{bal}}(\mathsf{n}_{\max}+T-\Delta) \\ \tilde{u}(\mathsf{n}_{\max}+1) & \cdots & \tilde{u}(\mathsf{n}_{\max}+T-\Delta) \end{bmatrix}. \tag{2}$$

Output: a finite time-Δ balanced representation (A_{bal}, B_{bal}, C_{bal}, D_{bal}) of $\mathscr{S}$.

Note 1 (*Finite time-Δ balancing*). The basic algorithm factors a finite $\Delta \times \Delta$ block Hankel matrix of Markov parameters $\mathfrak{H}$, so that the obtained representation (A_{bal}, B_{bal}, C_{bal}, D_{bal}) is *finite time-Δ balanced*. For $\Delta \gg \mathsf{n}_{\max}$, the representation obtained is close to an infinite time balanced one. Determining an appropriate value for the parameter Δ, however, is a problem in its own right, and will be addressed in the paper.

Note 2 (*Model reduction*). Identification of a state-space model in a balanced basis is motivated by the effective heuristic for model reduction by truncation in that basis. In principle, it is possible to identify the model in any basis and then apply standard algorithms for state transformation to a balanced basis. The direct algorithm discussed in this paper, however, has the advantage over the indirect approach that it allows to *identify* a reduced order model directly from data without ever computing a full order model.

The model reduction can be done by Step 5 of the basic algorithm. Let r be the desired order of the reduced model and let $\tilde{X}_{\mathrm{red}}$ be the truncated to the first r rows balanced state sequence $\tilde{X}_{\mathrm{bal}}$. As a heuristic model reduction procedure, we derive the reduced model parameters by solving the *least-squares problem*

$$\begin{bmatrix} \tilde{x}_{\mathrm{red}}(\mathsf{n}_{\max}+2) & \cdots & \tilde{x}_{\mathrm{red}}(\mathsf{n}_{\max}+M) \\ \tilde{y}(\mathsf{n}_{\max}+1) & \cdots & \tilde{y}(\mathsf{n}_{\max}+T-\Delta) \end{bmatrix} = \begin{bmatrix} A_{\mathrm{red}} & B_{\mathrm{red}} \\ C_{\mathrm{red}} & D_{\mathrm{red}} \end{bmatrix} \times \begin{bmatrix} \tilde{x}_{\mathrm{red}}(\mathsf{n}_{\max}+1) & \cdots & \tilde{x}_{\mathrm{red}}(\mathsf{n}_{\max}+T-\Delta) \\ \tilde{u}(\mathsf{n}_{\max}+1) & \cdots & \tilde{u}(\mathsf{n}_{\max}+T-\Delta) \end{bmatrix}$$

in place of the exact system of equations (2). The obtained model (A_{red}, B_{red}, C_{red}, D_{red}) is *not* the same as the model obtained by truncation of the (finite time-Δ) balanced model. In particular, we do not know about error bounds similar to the ones available for the (infinite time) balanced model reduction. The model reduction question is not further discussed in this paper and will be treated elsewhere.

Note 3. In Moonen and Ramos (1993); Van Overschee and De Moor (1996), it is not mentioned that the Hankel matrix of Markov parameters $\mathfrak{H}$ is computed. Also in Moonen and Ramos (1993), it is not mentioned that the matrix Y_0 of sequential zero input responses is computed. In the present paper, we interpret these algorithms as implementations of the above basic algorithm and reveal their structure. The important difference among the algorithms of Moonen–Ramos, Van Overschee–De Moor, and Algorithm 7 proposed in this paper is namely the method of computation of the matrix Y_0 and the impulse response H.

Step 1, the computation of the impulse response, is the crucial one. In fact, once H is computed, a balanced model can be obtained directly via Kung's algorithm. This gives the alternative deterministic balanced model identification algorithm outlined in Algorithm 2.

Algorithm 2. *Input*: a time series $\tilde{w}=(\tilde{u}, \tilde{y})$, an upper bound $\mathtt{n}_{\max}$ of the system order, and a finite time balancing parameter $\Delta > \mathtt{n}_{\max}$.

1. Find the first 2Δ samples $H(0), \ldots, H(2\Delta-1)$ of the impulse response matrix of $\mathscr{S}$ and let $H := \operatorname{col}(H(0), \ldots, H(2\Delta-1))$.
2. Compute the restricted SVD, $\mathfrak{H}=U\Sigma V^{\top}$, of the block Hankel matrix of Markov parameters $\mathfrak{H}=\mathscr{H}_{\Delta}(\sigma H) \in \mathbb{R}^{\Delta\mathtt{p}\times\Delta\mathtt{m}}$.
3. Define $\mathscr{O}_{\mathrm{bal}} := U\sqrt{\Sigma}$ and $\mathscr{C}_{\mathrm{bal}} := \sqrt{\Sigma}V^{\top}$.
4. Let $D_{\mathrm{bal}}=H(0)$, B_{bal} be equal to the first $\mathtt{m}$ columns of $\mathscr{C}_{\mathrm{bal}}$ (the first block column), C_{bal} be equal to the first $\mathtt{p}$ rows of $\mathscr{O}_{\mathrm{bal}}$ (the first block row), and A_{bal} be the solution of the equation $(\sigma^*\mathscr{O}_{\mathrm{bal}})A_{\mathrm{bal}}=\sigma\mathscr{O}_{\mathrm{bal}}$, where σ and σ^*, acting on a block matrix, remove, respectively, the first and the last block rows.

Output: a finite time-Δ balanced representation $(A_{\mathrm{bal}}, B_{\mathrm{bal}}, C_{\mathrm{bal}}, D_{\mathrm{bal}})$ of $\mathscr{S}$.

In Algorithm 2, once the impulse response is computed, the parameters A_{bal}, B_{bal}, C_{bal}, D_{bal} are obtained without returning to the original observed data. Yet another alternative for computing a balanced representation directly from data is to obtain the parameters A_{bal} and C_{bal} as in Algorithm 2 from $\mathscr{O}_{\mathrm{bal}}$ and the parameters B_{bal} and D_{bal} (as well as the initial condition $x_{\mathrm{bal}}(1)$, under which $\tilde{w}$ is obtained) from the linear system of equations

$$\tilde{y}(t) = C_{\mathrm{bal}}A_{\mathrm{bal}}^{t}x_{\mathrm{bal}}(1) + \sum_{\tau=1}^{t-1} C_{\mathrm{bal}}A_{\mathrm{bal}}^{t-1-\tau}B_{\mathrm{bal}}\tilde{u}(\tau) + D_{\mathrm{bal}}\delta(t+1), \quad \text{for } t=1,\ldots,T, \tag{3}$$

using the original data. (By using Kronecker products (3) can be solved explicitly.) The resulting Algorithm 3 is in the spirit of the MOESP-type algorithms, see Verhaegen and Dewilde (1992).

Algorithm 3. *Input*: a time series $\tilde{w}=(\tilde{u}, \tilde{y})$, an upper bound $\mathtt{n}_{\max}$ of the system order, and a finite time balancing parameter $\Delta > \mathtt{n}_{\max}$.

1. Find the first 2Δ samples $H(0), \ldots, H(2\Delta-1)$ of the impulse response matrix of $\mathscr{S}$ and let $H := \operatorname{col}(H(0), \ldots, H(2\Delta-1))$.
2. Compute the restricted SVD, $\mathfrak{H}=U\Sigma V^{\top}$, of the block Hankel matrix of Markov parameters $\mathfrak{H}=\mathscr{H}_{\Delta}(\sigma H) \in \mathbb{R}^{\Delta\mathtt{p}\times\Delta\mathtt{m}}$.
3. Define $\mathscr{O}_{\mathrm{bal}} := U\sqrt{\Sigma}$.
4. Let C_{bal} be equal to the first p rows of $\mathscr{O}_{\mathrm{bal}}$ (the first block row), and A_{bal} be the solution of the equation

   $$(\sigma^*\mathscr{O}_{\mathrm{bal}})A_{\mathrm{bal}}=\sigma\mathscr{O}_{\mathrm{bal}}.$$

5. Solve (3) for B_{bal}, D_{bal}, and $x_{\mathrm{bal}}(1)$.

Output: a finite time-Δ balanced representation $(A_{\mathrm{bal}}, B_{\mathrm{bal}}, C_{\mathrm{bal}}, D_{\mathrm{bal}})$ of $\mathscr{S}$.

Simulation results show that in the presence of noise "going back to the data", as done in the basic algorithm and in Algorithm 3, leads to more accurate results. This gives an indication that the basic algorithm and Algorithms 3 might be superior to Algorithm 2 in the noisy case.

The outline of the paper is as follows. In Section 2 we present two lemmas that are instrumental for the derivation of the algorithm. The first one, which we call the *fundamental lemma*, gives conditions on a trajectory $\tilde{w}$, under which any response of $\mathscr{S}$ with length L belongs to the image of the Hankel matrix $\mathscr{H}_L(\tilde{w})$. As a consequence, any response of length L can be found as $\mathscr{H}_L(\tilde{w})g$ for a suitable $g \in \mathbb{R}^{T-L+1}$. The second lemma, which we call the *weaving lemma*, shows how an arbitrary long response of $\mathscr{S}$ can be obtained from a finite amount of data $\tilde{w}$ by weaving together segments of the desired response.

In Section 3, the fundamental lemma is applied for construction of the impulse response H. Theorem 4 gives an algorithm for the computation of H, based on the construction $H=\mathscr{H}_{2\Delta}(\tilde{w})G$, for a suitable G. This approach gives a limited length response. Using the weaving lemma, an algorithm is derived that computes arbitrary many samples of the impulse response. By monitoring the decay of the impulse response while computing it, the parameter Δ of the basic algorithm can be chosen adaptively.

In Section 4, an algorithm for the computation of the matrix Y_0 that appears in the basic algorithm is described. It is also based on the fundamental lemma and in analogy with the impulse response computation has a block version and an iterative version. We show that the block version of the algorithm is actually equivalent to the famous oblique projection from the classical subspace algorithms, which gives a system theoretic interpretation of the oblique projection. (In the subspace identification literature the oblique projection is defined and interpreted as a geometric operation and its system theoretic meaning remains hidden.)

2. Fundamental lemmas

Denote by $\mathscr{B}|_{[1,L]}$ the set of all trajectories of the system $\mathscr{S}$ over the time interval $[1, L]$, i.e.,

$$\mathscr{B}|_{[1,L]} := \left\{ w = \begin{bmatrix} u \\ y \end{bmatrix} := \left(\begin{bmatrix} u(1) \\ y(1) \end{bmatrix}, \ldots, \begin{bmatrix} u(L) \\ y(L) \end{bmatrix} \right) \right| \exists x(1), \ldots, x(L+1) \text{ such that (1) holds} \Big\}.$$

The notion of persistency of excitation is defined next, cf. (Van Overschee & De Moor, 1996, Definition 5).

Definition 1 (*Persistency of excitation*). The sequence $\tilde{u}=(\tilde{u}(1), \ldots, \tilde{u}(T))$ is persistently exciting of order L if the Hankel matrix $\mathscr{H}_L(\tilde{u})$ is of full row rank.

Lemma 2 (*Fundamental lemma Willems, Rapisarda, Markovsky, and DeMoor (2004)*). *Let*

1. $\tilde{w} = (\tilde{u}, \tilde{y})$ *be a trajectory of the LTI system* $\mathscr{S}$, *i.e.,*

$$\tilde{w} = \begin{bmatrix} \tilde{u} \\ \tilde{y} \end{bmatrix} = \left(\begin{bmatrix} \tilde{u}(1) \\ \tilde{y}(1) \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{u}(T) \\ \tilde{y}(T) \end{bmatrix} \right) \in \mathscr{B}|_{[1,T]};$$

2. *the system* $\mathscr{S}$ *be controllable*; *and*
3. *the input sequence* $\tilde{u}$ *be persistently exciting of order* $L + \mathtt{n}$, *where* $\mathtt{n}$ *is the order* (*the dimension of the state space*) *of* $\mathscr{S}$.

Then any L samples long trajectory $w = (u, y)$ *of* $\mathscr{S}$ *can be written as a linear combination of the columns of* $\mathscr{H}_L(\tilde{w})$ *and any linear combination* $\mathscr{H}_L(\tilde{w})g$, $g \in \mathbb{R}^{T-L+1}$, *is also a trajectory of* $\mathscr{S}$, *i.e.,* $\operatorname{col\,span}(\mathscr{H}_L(\tilde{w})) = \mathscr{B}|_{[1,L]}$.

Proof. The proof of these results as well as interesting corollaries are given in Willems et al. (2004). □

The fundamental lemma states conditions under which the Hankel matrix $\mathscr{H}_L(\tilde{w})$ has the "correct" image (and as a consequence the "correct" left kernel). The conditions are not verifiable from the data $\tilde{w}$ alone, so that in identification problems, where only $\tilde{w}$ is given, they should be assumed. In addition, for the derivation of the algorithm, we assume that an upper bound $\mathtt{n}_{\max}$ on the system order $\mathtt{n}$ and an upper bound $\mathtt{l}_{\max}$ on the system lag $\mathtt{l}$ are a priori known. The system lag $\mathtt{l}$ is defined as the observability index of $\mathscr{S}$. Note that $\mathtt{n}_{\max}$ can be used as a loose upper bound on $\mathtt{l}$. Generically $\mathtt{p}\mathtt{l} = \mathtt{n}$ and $\mathtt{p}\mathtt{l}_{\max} = \mathtt{n}_{\max}$. Assumptions 1–3 of the fundamental lemma and the assumption that $\mathtt{n}_{\max}$ is given are the standard assumptions for deterministic subspace identification, see, e.g., Van Overschee and De Moor (1996, Chapter 2).

The next lemma shows how a long response can be constructed by weaving together short ones.

Lemma 3 (*Weaving responses*). *Let*

1. $\tilde{w}^{(1)}$ *be a* T_1 *samples long trajectory of* $\mathscr{S}$, *i.e.,* $\tilde{w}^{(1)} \in \mathscr{B}|_{[1,T_1]}$;
2. $\tilde{w}^{(2)}$ *be a* T_2 *samples long trajectory of* $\mathscr{S}$, *i.e.,* $\tilde{w}^{(2)} \in \mathscr{B}|_{[1,T_2]}$;
3. *the last l samples of* $\tilde{w}^{(1)}$ *coincide with the first l samples of* $\tilde{w}^{(2)}$, *i.e.,*

$$\begin{aligned}&(\tilde{w}^{(1)}(T_1 - l + 1), \ldots, \tilde{w}^{(1)}(T_1)) \\ &\quad = (\tilde{w}^{(2)}(1), \ldots, \tilde{w}^{(2)}(l));\end{aligned}$$

4. *l is larger than or equal to the lag* $\mathtt{l}$ *of the system* $\mathscr{S}$.

Then the trajectory

$$w := (\tilde{w}^{(1)}(1), \ldots, \tilde{w}^{(1)}(T_1), \tilde{w}^{(2)}(l+1), \ldots, \tilde{w}^{(2)}(T_2)), \tag{4}$$

obtained by weaving together $\tilde{w}^{(1)}$ *and* $\tilde{w}^{(2)}$ *is a trajectory of* $\mathscr{S}$, *i.e.,* $w \in \mathscr{B}|_{[1, T_1+T_2-l]}$.

Proof. Let $\tilde{x}^{(1)} := (\tilde{x}^{(1)}(1), \ldots, \tilde{x}^{(1)}(T_1 + 1))$ and $\tilde{x}^{(2)} := (\tilde{x}^{(2)}(1), \ldots, \tilde{x}^{(2)}(T_2 + 1))$ be the state sequences of $\mathscr{S}$ associated with $w^{(1)}$ and $w^{(2)}$, respectively. Assumptions 3 and 4 imply that $\tilde{x}^{(1)}(T_1 + 1) = \tilde{x}^{(2)}(l + 1)$. Therefore, (4) is a trajectory of $\mathscr{S}$. □

3. Computation of the impulse response

In this section, we consider Step 1 of the basic algorithm: given a trajectory $\tilde{w} = (\tilde{u}, \tilde{y})$, find the first 2Δ samples H of the impulse response of $\mathscr{S}$. We need the first 2Δ samples of the impulse response in order to construct the $\Delta \times \Delta$ block Hankel matrix $\mathscr{H}_\Delta(\sigma H)$, whose factorization in turn gives the finite time-Δ balancing transformation.

From the fundamental lemma, we know that, under suitable conditions, $\operatorname{col\,span}(\mathscr{H}_{2\Delta}(\tilde{w})) = \mathscr{B}|_{[1,2\Delta]}$. This implies that there exists a matrix G, such that $\mathscr{H}_{2\Delta}(\tilde{y})G = H$. Thus, the problem reduces to the one of finding a particular G.

Let row dim denote the number of *block* rows of a matrix or vector and define U_{p}, U_{f}, Y_{p}, Y_{f} as follows

$$\mathscr{H}_{\mathtt{l}_{\max}+2\Delta}(\tilde{u}) =: \begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \end{bmatrix}, \quad \mathscr{H}_{\mathtt{l}_{\max}+2\Delta}(\tilde{y}) =: \begin{bmatrix} Y_{\mathrm{p}} \\ Y_{\mathrm{f}} \end{bmatrix},$$

where $\operatorname{row\,dim}(\mathrm{U_p}) = \operatorname{row\,dim}(\mathrm{Y_p}) = \mathtt{l}_{\max}$ and $\operatorname{row\,dim}(\mathrm{U_f}) = \operatorname{row\,dim}(\mathrm{Y_f}) = 2\Delta$.

Theorem 4 (*Impulse response from data*). *Let* $\tilde{w} = (\tilde{u}, \tilde{y})$ *be a trajectory of a controllable LTI system* $\mathscr{S}$ *of order* $\mathtt{n} \leqslant \mathtt{n}_{\max}$ *and lag* $\mathtt{l} \leqslant \mathtt{l}_{\max}$ *and let* $\tilde{u}$ *be persistently exciting of order* $2\Delta + \mathtt{l}_{\max} + \mathtt{n}_{\max}$. *Then the system of equations*

$$\begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{p}} \end{bmatrix} G = \begin{bmatrix} 0_{\mathtt{m}\mathtt{l}_{\max} \times \mathtt{m}} \\ \begin{bmatrix} I_{\mathtt{m}} \\ 0_{\mathtt{m}(2\Delta-1) \times \mathtt{m}} \end{bmatrix} \\ 0_{\mathtt{p}\mathtt{l}_{\max} \times \mathtt{m}} \end{bmatrix}, \tag{5}$$

is solvable for $G \in \mathbb{R}^{\bullet \times \mathtt{m}}$. *Moreover, for any particular solution* $\bar{G}$, *the matrix* $Y_f \bar{G}$ *contains the first* 2Δ *samples of the impulse response of* $\mathscr{S}$, *i.e.,* $Y_f \bar{G} = H$.

Proof. Under the assumptions of the theorem, we can apply the fundamental lemma with $L = \mathtt{l}_{\max} + 2\Delta$, thus

$$\operatorname{col\,span}(\mathscr{H}_{\mathtt{l}_{\max}+2\Delta}(\tilde{w})) = \mathscr{B}|_{[1,\, \mathtt{l}_{\max}+2\Delta]}.$$

First we show that (5) is solvable. The impulse response $\left(\begin{bmatrix} I_{\mathtt{m}} \\ 0_{\mathtt{m}(2\Delta-1)\times\mathtt{m}} \end{bmatrix}, H \right)$ is a (matrix valued) response of $\mathscr{S}$ obtained under zero initial conditions. Because of the zero initial conditions, $\left(\begin{bmatrix} I_{\mathtt{m}} \\ 0_{\mathtt{m}(2\Delta-1)\times\mathtt{m}} \end{bmatrix}, H \right)$ preceded by any number of zeros remains a response of $\mathscr{S}$. Therefore, there exists a matrix $\bar{G}$, such that

$$\begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{p}} \\ Y_{\mathrm{f}} \end{bmatrix} \bar{G} = \begin{bmatrix} 0_{\mathtt{m}\mathtt{l}_{\max}\times\mathtt{m}} \\ \begin{bmatrix} I_{\mathtt{m}} \\ 0_{\mathtt{m}(2\Delta-1)\times\mathtt{m}} \end{bmatrix} \\ 0_{\mathtt{p}\mathtt{l}_{\max}\times\mathtt{m}} \\ H \end{bmatrix}.$$

This shows that there exists a solution $\bar{G}$ of (5) and therefore $Y_f\bar{G}$ is the impulse response.

Conversely, let G be a solution of (5). We have

$$\begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{p}} \\ Y_{\mathrm{f}} \end{bmatrix} G = \begin{bmatrix} 0_{\mathtt{m}\mathtt{l}_{\max}\times\mathtt{m}} \\ \begin{bmatrix} I_{\mathtt{m}} \\ 0_{\mathtt{m}(2\Delta-1)\times\mathtt{m}} \end{bmatrix} \\ 0_{\mathtt{p}\mathtt{l}_{\max}\times\mathtt{m}} \\ Y_{\mathrm{f}}G \end{bmatrix} \tag{6}$$

and the fundamental lemma guarantees that the right-hand side of (6) is a response of $\mathscr{S}$. The response is identically zero during the first $\mathtt{l}_{\max}$ samples, which (using the assumption $\mathtt{l}_{\max}\geqslant l$) guarantees that the initial conditions are set to zero. The input $\begin{bmatrix} I_{\mathtt{m}} \\ 0_{\mathtt{m}(2\Delta-1)\times\mathtt{m}} \end{bmatrix}$ is a matrix valued impulse, so that the corresponding output $Y_{\mathrm{f}}G$ is indeed the impulse response H. □

Theorem 1 gives the following block algorithm for the computation of H.

Algorithm 4 (*Block computation of the impulse response*). *Input*: $\tilde{u}$, $\tilde{y}$, $\mathtt{l}_{\max}$, and Δ.

1. Solve the system of Eq. (5). Let $\bar{G}$ be the computed solution.
2. Compute $H = Y_f\bar{G}$.

Output: the first 2Δ samples of the impulse response H.

Note 4 (*Efficient implementation via QR factorization*). The system of equations in Step 1 of Algorithm 4 can be solved efficiently by first "compressing the data" via the QR decomposition

$$\begin{bmatrix} U_{\mathrm{p}}^\top & U_{\mathrm{f}}^\top & Y_{\mathrm{p}}^\top & | & Y_{\mathrm{f}}^\top \end{bmatrix} = QR, \quad R^\top =: \left[\begin{array}{c|c} R_{11} & 0 \\ \hline R_{21} & R_{22} \end{array}\right],$$

where $R_{11} \in \mathbb{R}^{(\mathtt{m}(\mathtt{l}_{\max}+2\Delta)+\mathtt{p}\mathtt{l}_{\max})\times(\mathtt{m}(\mathtt{l}_{\max}+2\Delta)+\mathtt{p}\mathtt{l}_{\max})}$, and then computing the pseudo-inverse of the R_{11} block. We have

$$H = Y_{\mathrm{f}} \begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{p}} \end{bmatrix}^+ \begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix} = R_{21}R_{11}^+ \begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix}.$$

We proceed to point at an inherent limitation of Algorithm 4 when dealing with finite amount of data.

Let T samples of the input and the output be given. The persistency of excitation assumption in Theorem 4 requires that $\mathscr{H}_{2\Delta+\mathtt{l}_{\max}+\mathtt{n}_{\max}}(\tilde{u})$ is full row rank, which implies that

$$\mathtt{m}(2\Delta + \mathtt{l}_{\max} + \mathtt{n}_{\max}) \leqslant T - (2\Delta + \mathtt{l}_{\max} + \mathtt{n}_{\max}) + 1$$
$$\Rightarrow \Delta \leqslant \frac{1}{2}\left(\frac{T+1}{\mathtt{m}+1} - \mathtt{l}_{\max} - \mathtt{n}_{\max}\right).$$

Thus using Algorithm 4, we are limited in the number of samples of the impulse response that can be computed. Moreover, for efficiency and accuracy (in the presence of noise), we want to have Hankel matrices U_{p}, U_{f}, etc., with many more columns than rows, which implies small Δ.

According to Lemma 3, however, it is possible to find arbitrary many samples of the impulse response. Algorithm 5 does this by computing iteratively blocks of L consecutive samples, where

$$1 \leqslant L \leqslant \frac{1}{2}\left(\frac{T+1}{\mathtt{m}+1} - \mathtt{l}_{\max} - \mathtt{n}_{\max}\right). \tag{7}$$

Moreover, monitoring the decay of H (provided the system is stable) while computing it, gives a heuristic way to determine the parameter Δ.

In the recursive algorithm the matrices U_{p}, U_{f}, Y_{p}, Y_{f} defined above are redefined as follows:

$$\mathscr{H}_{\mathtt{l}_{\max}+L}(\tilde{u}) =: \begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \end{bmatrix}, \quad \mathscr{H}_{\mathtt{l}_{\max}+L}(\tilde{y}) =: \begin{bmatrix} Y_{\mathrm{p}} \\ Y_{\mathrm{f}} \end{bmatrix},$$

where row $\dim(U_{\mathrm{p}}) = $ row $\dim(Y_{\mathrm{p}}) = \mathtt{l}_{\max}$ and row dim $(U_{\mathrm{f}}) = $ row $\dim(Y_{\mathrm{f}}) = L$.

Algorithm 5 (*Iterative computation of the impulse response*). *Input*: $\tilde{u}$, $\tilde{y}$, $\mathtt{n}_{\max}$, $\mathtt{l}_{\max}$, and either Δ or a convergence tolerance ε.

1. Choose the number of samples L computed in one iteration step according to the persistency of excitation of $\tilde{u}$. In particular (7) should be satisfied.
2. *Initialization:* $k := 0$, $F_{\mathrm{u}}^{(0)} := \begin{bmatrix} 0_{\mathtt{m}\mathtt{l}_{\max}\times\mathtt{m}} \\ \begin{bmatrix} I_{\mathtt{m}} \\ 0_{\mathtt{m}(L-1)\times\mathtt{m}} \end{bmatrix} \end{bmatrix}$ and $F_{\mathrm{y,p}}^{(0)} := 0_{\mathtt{p}\mathtt{l}_{\max}}$.
3. *Repeat*
    - 3.1. Solve the system $\begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{p}} \end{bmatrix} G^{(k)} = \begin{bmatrix} F_{\mathrm{u}}^{(k)} \\ F_{\mathrm{y,p}}^{(k)} \end{bmatrix}$.
    - 3.2. Compute the response $H^{(k)} := F_{\mathrm{y,f}}^{(k)} := Y_{\mathrm{f}}G^{(k)}$.
    - 3.3. Define $F_{\mathrm{y}}^{(k)} := \begin{bmatrix} F_{\mathrm{y,p}}^{(k)} \\ F_{\mathrm{y,f}}^{(k)} \end{bmatrix}$.
    - 3.4. Shift F_{u} and F_{y}: $F_{\mathrm{u}}^{(k+1)} := \begin{bmatrix} \sigma^L F_{\mathrm{u}}^{(k)} \\ 0_{\mathtt{m}L\times\mathtt{m}} \end{bmatrix}$, $F_{\mathrm{y,p}}^{(k+1)} := \sigma^L F_{\mathrm{y}}^{(k)}$.
    - 3.5. Increment the iteration counter $k := k+1$.
4. *Until* $\begin{cases} kL \geqslant 2\Delta & \text{if } \Delta \text{ is given,} \\ \|H^{(k-1)}\|_F \leqslant \varepsilon \text{ and } kL \text{ is even} & \text{otherwise.} \end{cases}$
5. If Δ is not given as an input, define $\Delta := kL/2$.

Output: $H = \mathrm{col}(H^{(0)}, \dots, H^{(k-1)})$ and Δ if Δ is not given as an input.

Proposition 5. *Let $(\tilde u, \tilde y)$ be a trajectory of a controllable LTI system $\mathscr{S}$ of order $\mathtt{n} \leqslant \mathtt{n}_{\max}$ and lag $\mathtt{l} \leqslant \mathtt{l}_{\max}$, and let $\tilde u$ be persistently exciting of order $L + \mathtt{l}_{\max} + \mathtt{n}_{\max}$. Then Algorithm* 5 *computes the first 2Δ samples of the impulse response of $\mathscr{S}$.*

Proof. Under the assumptions of the proposition, we can apply Theorem 1, with the parameter Δ in the theorem, replaced by the parameter L, selected on Step 1 of the algorithm. Steps 3.1 and 3.1 of the recursive algorithm correspond to the steps of the block algorithm. The right-hand side $\begin{bmatrix} F_{\mathrm{u}}^{(k)} \\ F_{\mathrm{y,p}}^{(k)} \end{bmatrix}$ of the system of equations, solved in Step 3.1, is initialized so that $H^{(0)}$ is indeed the matrix of the first L samples of the impulse response.

The response computed on the $(k+1)$th iteration step, $k \geqslant 1$, is a response due to zero input and its first $\mathtt{l}_{\max}$ samples overlap the last $\mathtt{l}_{\max}$ samples of the response computed on the kth iteration step. By the weaving lemma, their concatenation is a valid response. Applying this argument recursively, we have that H computed by the algorithm is the impulse response of the system. □

Note 5 (*Computation of an arbitrary response from data*). For the purpose of balanced subspace identification, we need construction of the impulse response from data. In Markovsky, Willems, Rapisarda, and De Moor (2004), Theorem 4 and Algorithm 5 are modified to compute an arbitrary response directly from data.

Note 6 (*Efficient implementation via QR factorization*). The most expensive part of Algorithm 4 is solving the system of equations on Step 3.1. It can be solved efficiently via the QR factorization as described in Note 4. Moreover, since the matrix on the left-hand side of the system is fixed, the pseudo-inverse can be computed out of the iteration loop and used for all iterations.

4. Computation of sequential zero input responses

In this section, we consider step 2 of the basic algorithm: given $\tilde w = (\tilde u, \tilde y)$, find a sequential zero input responses Y_0 of $\mathscr{S}$. By "sequential", we mean that the initial conditions corresponding to the columns of Y_0 form a valid state sequence of $\mathscr{S}$.

Using the fundamental lemma, a set of zero input trajectories can be computed from data by solving the system of equations

$$\begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{p}} \end{bmatrix} G = \begin{bmatrix} U_{\mathrm{p}} \\ 0 \\ Y_{\mathrm{p}} \end{bmatrix} \tag{8}$$

and setting $Y_0 = Y_{\mathrm{f}} G$. Moreover, the Hankel structure of U_{p} and Y_{p} imply that Y_0 is a matrix of sequential responses. System (8) and $Y_0 = Y_{\mathrm{f}} G$ give a block algorithm for the computation of sequential zero input responses. It is analogous to the block algorithm for the computation of the impulse response and again the computation can be performed efficiently via the QR factorization.

Note 7 (*Connection with the oblique projection*). For any particular solution $\bar G$ of (8), $Y_0 = Y_f \bar G$ is equal to the oblique projection $Y_{\mathrm{f}}/_{U_{\mathrm{f}}} W_{\mathrm{p}}$, $W_{\mathrm{p}} := \begin{bmatrix} U_{\mathrm{p}} \\ Y_{\mathrm{p}} \end{bmatrix}$ of the classical subspace algorithms. To show this, let

$$\bar G = [W_{\mathrm{p}}^\top \quad U_{\mathrm{f}}^\top] \begin{bmatrix} W_{\mathrm{p}} W_{\mathrm{p}}^\top & W_{\mathrm{p}} U_{\mathrm{f}}^\top \\ U_{\mathrm{f}} W_{\mathrm{p}}^\top & U_{\mathrm{f}} U_{\mathrm{f}}^\top \end{bmatrix}^+ \begin{bmatrix} W_{\mathrm{p}} \\ 0 \end{bmatrix}$$

be the least-squares least norm solution of $\begin{bmatrix} U_{\mathrm{p}} \\ Y_{\mathrm{p}} \\ U_{\mathrm{f}} \end{bmatrix} G = \begin{bmatrix} U_{\mathrm{p}} \\ Y_{\mathrm{p}} \\ 0 \end{bmatrix}$, which is (8) with permuted rows, so that the solution is not changed. Then

$$Y_f \bar G = Y_{\mathrm{f}} [W_{\mathrm{p}}^\top \quad U_{\mathrm{f}}^\top] \begin{bmatrix} W_{\mathrm{p}} W_{\mathrm{p}}^\top & W_{\mathrm{p}} U_{\mathrm{f}}^\top \\ U_{\mathrm{f}} W_{\mathrm{p}}^\top & U_{\mathrm{f}} U_{\mathrm{f}}^\top \end{bmatrix}^+ \begin{bmatrix} W_{\mathrm{p}} \\ 0 \end{bmatrix}, \tag{9}$$

which is the definition of the oblique projection $Y_{\mathrm{f}}/_{U_{\mathrm{f}}} W_{\mathrm{p}}$, see Van Overschee and De Moor (1996, p. 21, Eq. (1.4). Thus, the oblique projection is an implementation of the block algorithm for the computation of the sequential zero input responses Y_0.

We proceed to present a recursive algorithm for the computation of Y_0, analogous to Algorithm 5 for the computation of the impulse response. An advantage of the recursive algorithm over the block one is that one is not restricted by the finite amount of data $\tilde w$ to a finite length responses Y_0.

Algorithm 6 (*Sequential zero input responses*). *Input*: u, y, $\mathtt{n}_{\max}$, $\mathtt{l}_{\max}$, and either the desired number of samples Δ or a convergence tolerance ε.

1. Choose the number of samples L computed in one iteration step according to the persistency of excitation of $\tilde u$. In particular (7) should be satisfied.
2. *Initialization*: $k := 0$, $F_{\mathrm{u}}^{(0)} := \begin{bmatrix} U_{\mathrm{p}} \\ 0 \end{bmatrix}$, and $F_{\mathrm{y,p}}^{(0)} := Y_{\mathrm{p}}$.
3. *Repeat*
    - 3.1. Solve the system $\begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{p}} \end{bmatrix} G^{(k)} = \begin{bmatrix} F_{\mathrm{u}}^{(k)} \\ F_{\mathrm{y,p}}^{(k)} \end{bmatrix}$.
    - 3.2. Compute the response $Y_0^{(k)} := F_{\mathrm{y,f}}^{(k)} := Y_{\mathrm{f}} G^{(k)}$.
    - 3.3. Define $F_{\mathrm{y}}^{(k)} := \begin{bmatrix} F_{\mathrm{y,p}}^{(k)} \\ F_{\mathrm{y,f}}^{(k)} \end{bmatrix}$.

3.4. Shift F_{u} and F_{y}: $F_{\mathrm{u}}^{(k+1)} := \begin{bmatrix} \sigma^L F_{\mathrm{u}}^{(k)} \\ 0_{\mathtt{m}L\times \mathtt{m}} \end{bmatrix}$, and $F_{\mathrm{y,p}}^{(k+1)} := \sigma^L F_{\mathrm{y}}^{(k)}$.
3.5. Increment the iteration counter $k := k+1$.
4. *Until* $\begin{cases} kL \geqslant \Delta & \text{if } \Delta \text{ is given,} \\ \|Y_0^{(k-1)}\|_F \leqslant \varepsilon & \text{otherwise.} \end{cases}$
5. If Δ is not given as an input, define $\Delta := kL$.

Output: $Y_0 = \mathrm{col}(Y_0^{(0)}, \ldots, Y_0^{(k-1)})$ and Δ, if Δ is not given as an input.

Proposition 6. *Under the assumptions of Proposition* 5, *Algorithm* 6 *computes a matrix of sequential zero input responses of* $\mathscr{S}$ *with* Δ *block rows.*

Proof. Similar to the proof of Proposition 5. □

Note 8 (*Efficient implementation via QR factorization*). Note 6is valid also for Algorithm 6. Moreover, in the basic algorithm, where both Algorithms 5 and 6 are applied, the pseudo-inverse needs to be computed only once.

5. An algorithm for deterministic balanced subspace identification

In the previous sections, we have specified Steps 1 and 2 of the basic algorithm. Steps 3–5 follow from standard derivations, which we now repeat for completeness. Let $\mathfrak{H}$ be the Hankel matrix of the Markov parameters $\mathfrak{H} := \mathscr{H}_\Delta(\sigma H)$. By factoring $\mathfrak{H}$ into $\mathscr{O}_{\mathrm{bal}}$ and $\mathscr{C}_{\mathrm{bal}}$ via the *restricted* SVD

$$\mathfrak{H} = U\Sigma V^\top = \underbrace{U\sqrt{\Sigma}}_{\mathscr{O}_{\mathrm{bal}}}\,\underbrace{\sqrt{\Sigma}V^\top}_{\mathscr{C}_{\mathrm{bal}}}, \quad \Sigma \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}$$

we obtain an extended observability matrix

$$\mathscr{O}_{\mathrm{bal}} = \mathrm{col}(C_{\mathrm{bal}}, C_{\mathrm{bal}}A_{\mathrm{bal}}, \ldots, C_{\mathrm{bal}}A_{\mathrm{bal}}^{\Delta-1})$$

and a corresponding extended controllability matrix

$$\mathscr{C}_{\mathrm{bal}} = [B_{\mathrm{bal}} \quad A_{\mathrm{bal}}B_{\mathrm{bal}} \quad \cdots \quad A_{\mathrm{bal}}^{\Delta-1}B_{\mathrm{bal}}]$$

in a finite time balanced basis. The basis is finite time-Δ balanced, because the finite time-Δ observability gramian $\mathscr{O}_{\mathrm{bal}}^\top\mathscr{O}_{\mathrm{bal}} = \Sigma$ and the finite time-Δ controllability gramian $\mathscr{C}_{\mathrm{bal}}\mathscr{C}_{\mathrm{bal}}^\top = \Sigma$ are equal.

The matrix of sequential zero input responses Y_0 can be written as $Y_0 = \mathscr{O}X$ for a certain extended observability matrix $\mathscr{O}$ and a state sequence X in the same basis. We find the balanced state sequence

$$\tilde{X}_{\mathrm{bal}} := [\tilde{x}_{\mathrm{bal}}(\mathtt{l}_{\max}+1) \quad \cdots \quad \tilde{x}_{\mathrm{bal}}(\mathtt{l}_{\max}+T+1-L)]$$

corresponding to $\mathscr{O}_{\mathrm{bal}} = U\sqrt{\Sigma}$ from

$$Y_0 = \mathscr{O}_{\mathrm{bal}}\tilde{X}_{\mathrm{bal}} \Rightarrow \tilde{X}_{\mathrm{bal}} = \sqrt{\Sigma^{-1}}U^\top Y_0.$$

The corresponding balanced representation $(A_{\mathrm{bal}}, B_{\mathrm{bal}}, C_{\mathrm{bal}}, D_{\mathrm{bal}})$ is computed from the system of equations

$$\begin{bmatrix} \tilde{x}_{\mathrm{bal}}(\mathtt{l}_{\max}+2)2) & \cdots & \tilde{x}_{\mathrm{bal}}(\mathtt{l}_{\max}+T+1-L) \\ \tilde{y}(\mathtt{l}_{\max}+1) & \cdots & \tilde{y}(\mathtt{l}_{\max}+T-L) \end{bmatrix} = \begin{bmatrix} A_{\mathrm{bal}} & B_{\mathrm{bal}} \\ C_{\mathrm{bal}} & D_{\mathrm{bal}} \end{bmatrix} \times \begin{bmatrix} \tilde{x}_{\mathrm{bal}}(\mathtt{l}_{\max}+1) & \cdots & \tilde{x}_{\mathrm{bal}}(\mathtt{l}_{\max}+T-L) \\ \tilde{u}(\mathtt{l}_{\max}+1) & \cdots & \tilde{u}(\mathtt{l}_{\max}+T-L) \end{bmatrix}. \tag{10}$$

This yields the following procedure.

Algorithm 7. *Input*: $\tilde{u}$, $\tilde{y}$, $\mathtt{n}_{\max}$, $\mathtt{l}_{\max}$, and either $\Delta > \mathtt{n}_{\max}$ or a convergence tolerance ε.

1. Apply Algorithm 5 with inputs $\tilde{u}$, $\tilde{y}$, $\mathtt{n}_{\max}$, $\mathtt{l}_{\max}$, and either Δ or ε, in order to compute the impulse response H and, if not given, the parameter Δ. In the latter case, let $\Delta := \max(\mathtt{n}_{\max}+1, \Delta)$
2. Apply Algorithm 6 with inputs $\tilde{u}$, $\tilde{y}$, $\mathtt{n}_{\max}$, $\mathtt{l}_{\max}$, L, and Δ, in order to compute the sequential zero input responses Y_0.
3. Form the Hankel matrix $\mathfrak{H} := \mathscr{H}_\Delta(\sigma H)$ and compute the restricted SVD $\mathfrak{H} = U\Sigma V^\top$.
4. Compute a balanced state sequence $\tilde{X}_{\mathrm{bal}} = \sqrt{\Sigma^{-1}}U^\top Y_0$.
5. Compute a balanced representation by solving (10).

Output: A_{bal}, B_{bal}, C_{bal}, D_{bal}, and Δ, if Δ is not given as an input.

The discussion in this section and Propositions 5 and 6 prove the following main result.

Theorem 7. *Under the assumptions of Proposition* 5, $(A_{\mathrm{bal}}, B_{\mathrm{bal}}, C_{\mathrm{bal}}, D_{\mathrm{bal}})$ *computed by Algorithm* 7 *is a finite time-*Δ *balanced representation of* $\mathscr{S}$.

6. Alternative algorithms

We outline the algorithms of Van Overschee–De Moor and Moonen–Ramos.

Algorithm 8 (Van Overschee & De Moor, 1996). *Input*: $\tilde{u}$, $\tilde{y}$, and a parameter i.

Define: $\begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \end{bmatrix} := \mathscr{H}_{2i}(\tilde{u})$, where row $\dim(\mathrm{U_p}) = \mathrm{i}$, and $\begin{bmatrix} Y_{\mathrm{p}} \\ Y_{\mathrm{f}} \end{bmatrix} := \mathscr{H}_{2i}(\tilde{y})$, where row $\dim(\mathrm{Y_p}) = \mathrm{i}$.

1. Compute the weight matrix $W := U_{\mathrm{p}}^\top(U_{\mathrm{p}}U_{\mathrm{p}}^\top)^{-1}J$, where J is the left–right flipped identity matrix.
2. Compute the oblique projection $Y_0 := Y_{\mathrm{f}}/_{U_{\mathrm{f}}}\begin{bmatrix} U_{\mathrm{p}} \\ Y_{\mathrm{p}} \end{bmatrix}$.
3. Compute the matrix $\hat{\mathfrak{H}} := Y_0W$ and the restricted SVD, $\hat{\mathfrak{H}} = U\Sigma V^\top$.

4. Compute a balanced state sequence $X_{\mathrm{bal}} = \sqrt{\Sigma^{-1}} U^{\top} Y_0$.
5. Compute a balanced representation by solving (10).

Output: $A_{\mathrm{bal}}, B_{\mathrm{bal}}, C_{\mathrm{bal}}, D_{\mathrm{bal}}$.

Note 9. The weight matrix W is different from the one in Van Overschee & De Moor (1996). In terms of the final result $\hat{\mathfrak{H}}$, however, it is equivalent. Another difference between Algorithm 8 and the deterministic balanced subspace algorithm of Van Overschee & De Moor (1996) is that the shifted state sequence appearing on the left-hand side of (10) is recomputed in Van Overschee & De Moor (1996) by another oblique projection.

Algorithm 9 (*Moonen & Ramos, 1993*). *Input*: $\tilde{u}$, $\tilde{y}$, and a parameter i.

Define: $\begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \end{bmatrix} := \mathscr{H}_{2i}(\tilde{u})$, where $\mathrm{row\ dim}(\mathrm{U_p}) = \mathrm{i}$, and $\begin{bmatrix} Y_{\mathrm{p}} \\ Y_{\mathrm{f}} \end{bmatrix} := \mathscr{H}_{2i}(\tilde{y})$, where $\mathrm{row\ dim}(\mathrm{Y_p}) = \mathrm{i}$.

0. Compute a matrix $[T_1\ T_2\ T_3\ T_4]$, whose rows form a basis for the left kernel of $\begin{bmatrix} U_{\mathrm{p}} \\ Y_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{f}} \end{bmatrix}$.
1. Compute the Hankel matrix of Markov parameters $\mathfrak{H} = T_4^{+}(T_2 T_4^{+} T_3 - T_1)J$.
2. Compute a matrix of zero input responses $Y_0 = T_4^{+}[T_1\ T_2] \begin{bmatrix} U_{\mathrm{p}} \\ Y_{\mathrm{p}} \end{bmatrix}$.
3. Compute the restricted SVD, $\mathfrak{H} = U \Sigma V^{\top}$.
4. Compute a balanced state sequence $X_{\mathrm{bal}} = \sqrt{\Sigma^{-1}} U^{\top} Y_0$.
5. Compute a balanced representation by solving (10).

Output: $A_{\mathrm{bal}}, B_{\mathrm{bal}}, C_{\mathrm{bal}}, D_{\mathrm{bal}}$.

Note 10. In the algorithms of Van Overschee–De Moor and Moonen–Ramos, the parameter i plays the row of the finite time balancing parameter Δ from the previous sections. Note that i is given and the "past" and the "future" are taken with equal length i.

Both Algorithms 8 and 9 fit in the outline of the basic algorithm but Steps 1 and 2 are implemented in rather different ways. As shown in Note 7, the oblique projection $Y_{\mathrm{f}} /_{U_{\mathrm{f}}} \begin{bmatrix} U_{\mathrm{p}} \\ Y_{\mathrm{p}} \end{bmatrix}$ is a matrix of sequential zero input responses. The weight matrix W, in the algorithm of Van Overschee–De Moor, is constructed so that $\hat{\mathfrak{H}} = Y_0 W$ is an approximation of the Hankel matrix of Markov parameters $\mathfrak{H}$; it is the sum of $\mathfrak{H}$ and a matrix of zero input responses.

The most expensive computation in the algorithm of Moonen–Ramos is Step 0, the computation of the annihilators $[T_1 \cdots T_4]$. The matrix $[T_1\ T_2] \begin{bmatrix} U_{\mathrm{p}} \\ Y_{\mathrm{p}} \end{bmatrix}$ is a non-minimal state sequence (the shift-and-cut operator (Rapisarda & Willems, 1997)) and T_4^{+} is a corresponding extended observability matrix. Thus $T_4^{+}[T_1\ T_2] \begin{bmatrix} U_{\mathrm{p}} \\ Y_{\mathrm{p}} \end{bmatrix}$ is a matrix of sequential zero input responses. It turns out that $(T_2 T_4^{+} T_3 - T_1)J$ is an extended controllability matrix (in the same basis), so that $T_4^{+}(T_2 T_4^{+} T_3 - T_1)J$ is the Hankel matrix of Markov parameters $\mathfrak{H}$.

A major difference between the proposed Algorithm 7, from one side, and the algorithms of Van Overschee–De Moor and Moonen–Ramos, on the other side, is that in Algorithm 7 the Hankel matrix $\mathfrak{H}$ is not computed but *constructed* from the impulse response that parameterizes it. This is a big computational saving because recomputing the same elements of $\mathfrak{H}$ is avoided. In addition, in approximate identification, where $\tilde{w}$ is not a trajectory of $\mathscr{S}$, the matrices $\hat{\mathfrak{H}}$ and $\mathfrak{H}$ computed by the algorithms of Van Overschee–De Moor and Moonen–Ramos are in general no longer Hankel, while the matrix $\mathfrak{H}$ in Algorithm 7 is by construction Hankel.

7. On the splitting of the data into "past" and "future"

In the algorithms of Moonen–Ramos and Van Overschee–De Moor the block-Hankel matrices $\begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \end{bmatrix}$ and $\begin{bmatrix} Y_{\mathrm{p}} \\ Y_{\mathrm{f}} \end{bmatrix}$ are split into "past" and "future" of an equal length. A natural question is why this is necessary and furthermore what is "optimal" according to certain relevant criteria partitionings. These questions are open for a long time, in particular in the context of the stochastic identification problem, see De Moor (2003).

In Sections 3 and 4 we showed that the past U_{p}, Y_{p} is used to assign the initial conditions and the future U_{f}, Y_{f} is used to compute the response. By weaving consecutive segments of the response, as done in Algorithms 5 and 6, the number of block rows in the future does not need to be equal to the required length of the response. Thus from the perspective of deterministic identification, the answer to the above question is:

> $\mathrm{row\ dim}(\mathrm{U_p}) = \mathrm{row\ dim}(\mathrm{Y_p}) = \mathtt{l}_{\max}$, i.e., the given least upper bound on the system lag $\mathtt{l}$, and $\mathrm{row\ dim}(\mathrm{U_f}) = \mathrm{row\ dim}(\mathrm{Y_f}) \in \{1, \ldots, \gamma - \mathtt{l}_{\max} + \mathtt{n}_{\max}\}$, where γ is the order of persistency of excitation of the input $\tilde{u}$.

As shown in Willems et al. (2004, Section 4, Comment 5)

$$\tilde{u} \text{ persistently exciting of order } \mathtt{l}_{\max} + 1 + \mathtt{n}_{\max} \tag{11}$$

is a sharp sufficient condition for identifiability of the system $\mathscr{S}$. By using the iterative algorithms for computation of the impulse response and sequential free responses with parameter $L = 1$, Algorithms 2, 3, and 7 are applicable under the same assumption, so that the partitioning "past $= \mathtt{l}_{\max}$ and future $= 1$" is consistent with (11).

Using the fundamental lemma, we can prove the following result.

Proposition 8. *Let $(\tilde{u}, \tilde{y})$ be a trajectory of a controllable LTI system $\mathscr{S}$ of order $\mathtt{n} \leqslant \mathtt{n}_{\max}$ and lag $\mathtt{l} \leqslant i$, and let $\tilde{u}$ be persistently exciting of order $2i + \mathtt{n}_{\max}$. Then the representations computed by Algorithms* 8 *and* 9 *are equivalent to $\mathscr{S}$. Moreover, the representation computed by Algorithm* 9 *is in a finite time-i balanced basis.*

Proposition 8 shows that Algorithms 8 and 9 are not parsimonious with respect to the available data: the system $\mathscr{S}$ could be identifiable with algorithms Algorithms 2, 3, and 7 but not with Algorithms 8 and 9.

Note that the persistency of excitation required by Algorithms 8 and 9 is a function of the finite time balancing parameter. This implies that with a finite amount of data, Algorithms 8 and 9 are limited in the ability to identify a balanced representation. In fact,

$$i \leqslant \left\lfloor \frac{T+1}{2(\max(\mathtt{m}, \mathtt{p})+1)} \right\rfloor.$$

In contrast, the persistency of excitation required by Algorithms 2, 3, and 7 depends only on the upper bounds on the system order and the lag and thus these algorithms can compute an infinite time balanced representation if (11) holds.

8. Simulations

In this section, we show some examples illustrate some advantages of the proposed algorithms. In all the experiments the system $\mathscr{S}$ is minimal with transfer function

$$\begin{aligned} &C(Iz-A)^{-1}B + D \\ &\quad = \frac{0.89172(z-0.5193)(z+0.5595)}{(z-0.4314)(z+0.4987)(z+0.6154)}. \end{aligned}$$

The input is a unit variance white noise and the data available for identification is the corresponding trajectory $\tilde{w}$ of $\mathscr{S}$, corrupted by independent white noises with standard deviation s. Although, our main concern is the correct work of the algorithms for exact data, i.e., with $s=0$, by varying the noise variance s, we can investigate empirically the performance under noise. The simulation time is $T=100$. In all experiments the upper bounds $\mathtt{n}_{\max}$ and $\mathtt{l}_{\max}$ are taken equal to the system order $\mathtt{n}=3$ and the parameter L is taken equal to 3.

Consider first the estimation of the impulse response. Fig. 1 shows the exact impulse response H of $\mathscr{S}$ and the estimate $\hat{H}$ computed by Algorithm 5. With exact data, $||H - \hat{H}||_F = 10^{-15}$, so that up to the numerical precision the match is exact. The plots on Fig. 1 show the deterioration of the estimates when the data is corrupted by noise.

Consider next the computation of the zero input response. Table 1 shows the error of estimation $e := ||Y_0 - \hat{Y}_0||_F$ and the corresponding amount of operations, where Y_0 is a matrix of exact sequential zero input responses with length $\Delta = 10$ and $\hat{Y}_0$ is its estimate computed from data. The estimate is computed in three ways: by Algorithm 6, implemented with the QR factorization, see Note 8; by the oblique projection, computed directly from (9); and by the oblique projection, computed via the QR factorization, see Note 7.

Algorithm 6 needs less computations and gives more accurate results than the alternatives. As already emphasized, the reason for this is that selecting the parameter $L = \mathtt{n}_{\max} = 3$ instead of $L = \Delta = 10$, as in a block computation, results in a more overdetermined system of equations in Step 3.1 of Algorithm 6 compared with system (8) used in the block algorithm. (For example, the difference is 95 vs. 88 columns.) As a result the noise is averaged over more samples, which leads to statistically better estimate. Solving several more overdetermined systems of equations instead of one more rectangular system can be more efficient, as it is in the example. In fact, the freedom in the choice of L makes it possible to optimize efficiency or another criterion.

All algorithms return a finite time balanced model. The next experiment illustrates the effect of the parameter Δ on the balancing. Let $W_{\mathrm{c}}/W_{\mathrm{o}}$ be the controllability/observability gramians of an infinite time balanced model and $\hat{W}_{\mathrm{c}}/\hat{W}_{\mathrm{o}}$ be the controllability/observability gramians of an identified model. Define closeness to balancing by

$$e_{\mathrm{bal}}^2 := \frac{||W_{\mathrm{c}} - \hat{W}_{\mathrm{c}}||_F^2 + ||W_{\mathrm{o}} - \hat{W}_{\mathrm{o}}||_F^2}{||W_{\mathrm{c}}||_F^2 + ||W_{\mathrm{o}}||_F^2}.$$

Fig. 2 shows e_{bal} as a function of Δ for the three algorithms presented in the paper. The estimates obtained by Algorithm 7 and the algorithm of Moonen–Ramos are identical estimate obtained by the algorithms of Van Overschee–De Moor is asymptotically equivalent, but for small Δ, is worse. This is a consequence of the fact that this algorithm uses an approximation of the Hankel matrix of Markov parameters. Fig. 2 also shows e_{bal} as a function of Δ for noisy data with $s = 0.001$ and the total number of floating point operations (flops) required by the three algorithms.

9. Conclusions

The impulse response and the sequence of zero input responses are the main tools for balanced subspace identification. Classically they are computed with the oblique projection. We gave a system theoretic interpretation of the oblique projection and a new algorithm for computation of a response directly from data. The new algorithm allows computation of an arbitrary long response from a finite data set and has the following advantages over the existing alternatives.

- The algorithms of Moonen–Ramos and Van Overschee–De Moor compute the whole Hankel matrix of Markov parameters $\mathfrak{H}$, while the proposed algorithms computes only the elements that uniquely specify $\mathfrak{H}$ and then *constructs* $\mathfrak{H}$ from them. Because of the Hankel structure, the algorithms of Moonen–Ramos and Van Overschee–De Moor recompute most elements

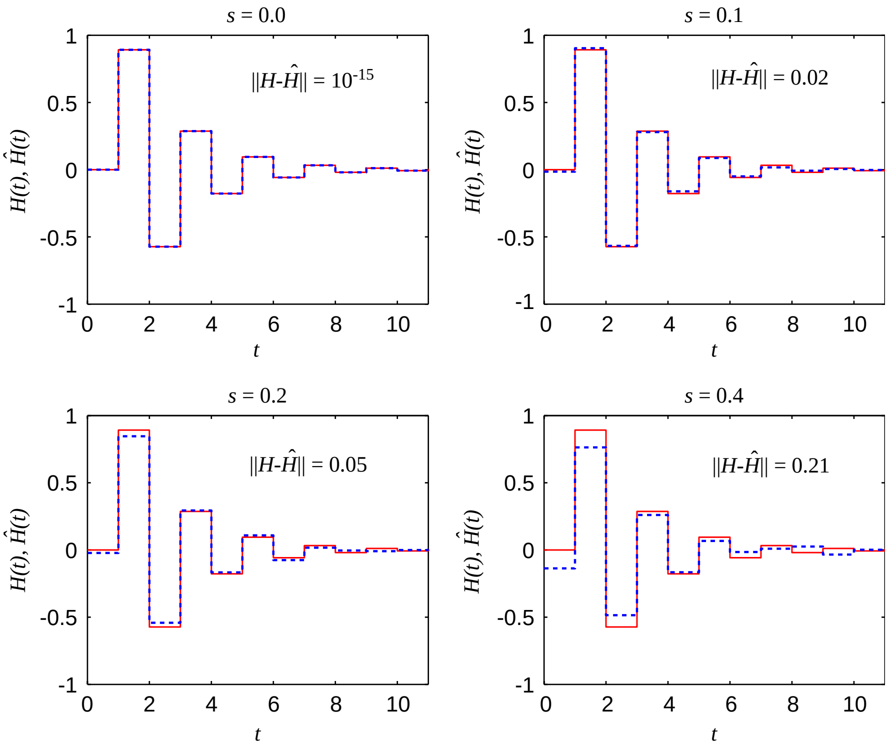

Fig. 1. Impulse response estimation. Solid line—exact impulse response H, dashed line—impulse response $\hat{H}$ computed from data via Algorithm 5.

Table 1
Error of estimation $e = ||Y_0 - \hat{Y}_0||_F$ and the corresponding amount of operations f in mega flops, where Y_0 is an exact sequence of zero input responses and $\hat{Y}_0$ is the estimate computed from data

Method	$s = 0.0$		$s = 0.1$		$s = 0.2$		$s = 0.4$	
	e	f	e	f	e	f	e	f
Alg. 6 with QR	10^{-14}	130	1.2990	131	2.5257	132	4.7498	132
formula (9)	10^{-10}	182	1.6497	186	3.2063	187	6.0915	189
(9) with QR	10^{-14}	251	1.6497	251	3.2063	251	6.0915	252

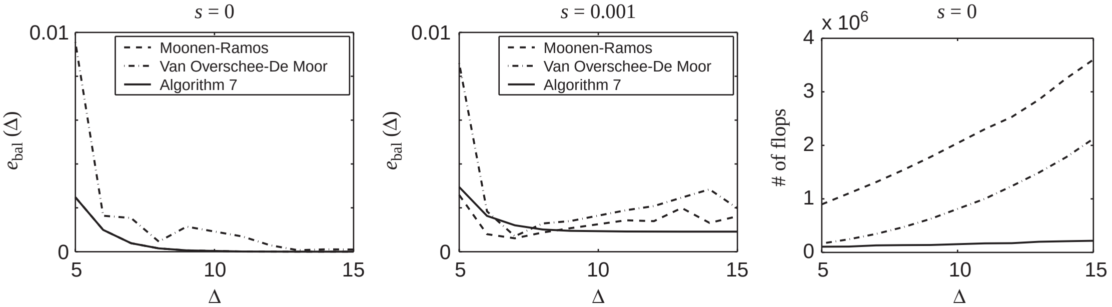

Fig. 2. Closeness to balancing e_{bal} and computational cost as functions of the finite time balancing parameter Δ.

of $\mathfrak{H}$ many times. This is an inefficient step in these algorithms.

- In the algorithms of Moonen–Ramos and Van Overschee–De Moor, the parameter Δ is supplied by the user. In the proposed algorithms, it can be determined automatically on the basis of a desired convergence tolerance of the impulse response, which is directly related to the closeness of the obtained representation to a balanced one.
- The algorithms of Moonen–Ramos and Van Overschee–De Moor compute finite time-Δ balanced representation with $\Delta \leqslant \lfloor \frac{1}{2}(T+1)/(\max(\mathtt{m},\mathtt{p})+1)\rfloor$, where T is the

length of the given time series $\tilde{w}$. By choosing the parameter Δ large enough, the proposed algorithms have no such limitation and can thus compute a representation that is arbitrary close to an infinite time balanced one.
- The proposed algorithms have weaker persistency of excitation conditions than the one needed for the algorithms of Moonen–Ramos and Van Overschee–De Moor. As a result, in certain cases, the proposed algorithms are applicable, while the algorithms of Moonen–Ramos and Van Overschee–De Moor are not.
- In the proposed algorithms, the computations can be organized to use more overdetermined system of equations, which result in more accurate estimates when $\tilde{w}$ is noisy.

We foresee other advantages on the level of the numerical implementation. Numerical issues, however, will be presented elsewhere.

Acknowledgements

Our research is supported by: Research Council KUL: GOA-Mefisto 666, several PhD/postdoc & fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects, G.0240.99 (multilinear algebra), G.0407.02 (support vector machines), G.0197.02 (power islands), G.0141.03 (Identification and cryptography), G.0491.03 (control for intensive care glycemia), G.0120.03 (QIT), G.0800.01 (collective intelligence), research communities (ICCoS, ANMMM); AWI: Bil. Int. Collaboration Hungary/ Poland; IWT: PhD Grants, Soft4s (softsensors), Belgian Federal Government: DWTC (IUAP IV-02 (1996-2001) and IUAP V-22 (2002-2006), PODO-II (CP/40: TMS and Sustainability); EU: CAGE; ERNSI; Eureka 2063-IMPACT; Eureka 2419-FliTE; Contract Research/agreements: Data4s, Electrabel, Elia, LMS, IPCOS, VIB.

References

Budin, M. A. (1971). Minimal realization of discrete linear systems from input–output observations. *IEEE Transactions on Automatic Control*, *16*(5), 395–401.

De Moor, B. (2003). On the number of rows and columns in subspace identification methods. *13th IFAC Symposium on Systems Identification* (pp. 1796–1801).

Gopinath, B. (1969). On the identification of linear time-invariant systems from input–output data. *Bell Systems Technical Journal*, *48*(5), 1101–1113.

Markovsky, I., Willems, J.C., Rapisarda, P., De Moor, B. (2004). *Data Driven Simulation with Applications to System Identification.* Technical report 04–53. Department of Electornics Engineering, K.U.Leuven. Accepted for publication in the proceedings of the 16th IFAC World Congress to be held in Prague, Czech Republic.

Moonen, M., & Ramos, J. (1993). A subspace algorithm for balanced state space system identification. *IEEE Transactions on Automactic Control*, *38*, 1727–1729.

Moore, B. C. (1981). Principal component analysis in linear systems: controllability, observability and model reduction. *IEEE Transactions on Automatic Control*, *26*(1), 17–31.

Pernebo, L., & Silverman, L. M. (1982). Model reduction via balanced state space representation. *IEEE Transactions on Automatic Control*, *27*, 382–387.

Rapisarda, P., & Willems, J. C. (1997). State maps for linear systems. *SIAM J. Control Optim.*, *35*(3), 1053–1091.

Van Overschee, P., & De Moor, B. (1996). *Subspace identification for linear systems: theory, implementation, applications*. Dortrecht: Kluwer.

Verhaegen, M., & Dewilde, P. (1992). Subspace model identification, Part 1: The output-error state-space model identification class of algorithms. *International Journal of Control*, *56*, 1187–1210.

Willems, J. C. (1986). From time series to linear system—Part II. Exact modelling. *Automatica*, *22*(6), 675–694.

Willems, J.C., Rapisarda, P., Markovsky, I., DeMoor, B., (2004). A note on persistency of excitation. *Systems & Control Letters*.

Ivan Markovsky was born in Sofia, Bulgaria in 1974. In 1998 he obtained MS degree in Control and Systems Engineering from the Technical University of Sofia. From 1998 to 2000 he was a research and teaching assistant at the Electrical Engineering Department of the University of Notre Dame, working on stability analysis of hybrid dynamical system. Since 2000 he is a doctoral student at the Department of Electrical Engineering of K.U.Leuven, with the research group SISTA (Signals, Identification, System Theory and Automation). His current research work is focused on identification methods in the behavioral setting.

Jan C. Willems was born in Bruges in Flanders, Belgium. He studied engineering at the University of Ghent. After his graduation in 1963, he went to the US, and obtained the M.Sc. degree from the University of Rhode Island in 1965, and the Ph.D. degree from the Massachusetts Institute of Technology in 1968, both in Electrical Engineering.

He was an assistant professor in the Department of electrical engineering at MIT from 1968 to 1973, with a 1 year leave of absence at the Department of Applied Mathematics and Theoretical physics of Cambridge University in the UK. In 1973, he was appointed Professor of Systems and Control in the Mathematics Department of the University of Groningen. In 2003, he became emeritus professor from the University of Groningen. Presently he is a full-time visiting professor at the Department of Electrical Engineering, with the research group on Signals, Identification, System Theory and Automation (SISTA), at the K.U.Leuven, Belgium. During the academic year 2003–2004, he held the Francqui Chair at the Faculty of Applied Sciences of the Université Catholique de Louvain.

His research interests involve various aspects of systems theory and control, especially the development of the behavioral appraoch.

Professor Willems is a fellow of the IEEE. He served terms as chairperson of the European Union Control Association and of the Dutch Mathematical Society. He has been on the editorial board of a number of journals, in particular, as managing editor of the SIAM Journal of Control and Optimization as and founding and managing editor of Systems & Control Letters.

In 1998, he received the IEEE Control Systems award.

Paolo Rapisarda got his “Laurea” degree in Computer Science at the University of Udine, Italy, and his Ph. D. in Mathematics at the University of Groningen, The Netherlands, working under the supervision of Prof. Jan C. Willems. He has worked at the Department of Electrical, Electronic and Computer Engineering of the University of Trieste, Italy. He is currently employed at the Department of Mathematics of the University of Maastricht, The Netherlands. His research interests are in control, identification, and in the behavioral approach to System and Control Theory.

Bart De Moor obtained a Master Degree (1983) and a Ph.D. (1988) in Electrical Engineering at the Katholieke Universiteit Leuven, Belgium and was a Visiting Research Associate at Stanford University (1988–1990). Currently, he is a full-time Professor at the Department of Electrical Engineering of the K.U.Leuven. His research interests are in numerical linear algebra and optimization, system theory, control and identification, quantum information theory, data-mining, information retrieval and bio-informatics, in which he (co-)authored more than 400 papers and 3 books. His work has won him several scientific awards (Leybold-Heraeus Prize (1986), Leslie Fox Prize (1989), Guillemin-Cauer best paper Award of the IEEE Transaction on Circuits and Systems (1990), Laureate of the Belgian Royal Academy of Sciences (1992), bi-annual Siemens Award (1994), best paper award of Automatica (IFAC, 1996), IEEE Signal Processing Society Best Paper Award (1999)). Since 2004 he is a fellow of the IEEE. From 1991–1999 he was the chief advisor on Science and Technology of several ministers of the Belgian Federal and the Flanders Regional Governments. He is in the board of three spin-off companies, of the Flemish Interuniversity Institute for Biotechnology, the Study Center for Nuclear Energy and several other scientific and cultural organizations. Since 2002 he also makes regular television appearances in the Science Show ‘Hoe?Zo!’ on national television in Belgium. Full biographical details can be found at www.esat.kuleuven.ac.be/~demoor.