Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

State Representations From Finite Time Series

Ivan Markovsky, Jan C. Willems, and Bart De Moor

***Abstract*— We present two algorithms for construction of a state sequence of a linear time-invariant system from a finite exact trajectory of that system. The first algorithm uses the classical in subspace identification splitting of the data into "past" and "future" and computes a bases for the past–future intersection. The second algorithm, based on the shift-and-cut map, operates on a half deeper Hankel matrix.**

***Index Terms*— Subspace identification, state construction, shift-and-cut map, persistency of excitation, behavioral systems.**

I. Introduction

The basic problem considered in this paper is to obtain the most powerful unfalsified linear time-invariant (LTI) model corresponding to a given finite time series $w_\mathrm{d} = \big(w_\mathrm{d}(1),\dots,w_\mathrm{d}(T)\big)$, $w_\mathrm{d}(t) \in \mathbb{R}^\mathtt{w}$. A discrete-time dynamical system $\mathscr{B}$ is viewed as a set of time series $(\mathbb{R}^\mathtt{w})^\mathbb{Z}$ and the notation $\mathscr{B}|_{[1,T]}$ is used for its restriction on the interval $1,\dots,T$. We say that a system $\mathscr{B}$ is unfalsified by the data w_d (or is an exact model of w_d) if $w_\mathrm{d} \in \mathscr{B}|_{[1,T]}$. The system $\mathscr{B}_1$ is more powerful than the system $\mathscr{B}_2$ if $\mathscr{B}_1 \subset \mathscr{B}_2$. For infinite time series, the most powerful unfalsified model (MPUM) of w_d always exists and is unique in the model class $\mathscr{L}^\mathtt{w}$ of LTI systems with $\mathtt{w}$ variables [1, Theorem 11]. We denote it by $\mathscr{B}_\mathrm{mpum}(w_\mathrm{d})$.

Our goal is to find algorithms that implement the mapping

$$w_\mathrm{d} \mapsto \mathscr{B}_\mathrm{mpum}(w_\mathrm{d}).$$

However, concrete representations of $\mathscr{B}_\mathrm{mpum}(w_\mathrm{d})$ are needed. In this paper, we consider the input/state/output one

$$\sigma x = Ax + Bu, \qquad y = Cx + Du, \qquad w = \Pi \operatorname{col}(u,y),$$

where σ is the forward shift operator $(\sigma x)(t) = x(t+1)$ and Π is a permutation matrix.

A related question of interest is the following one:

> Suppose that the data w_d is generated by an LTI model $\mathscr{B}$. Under what conditions can this model be recovered back from the data?

This identifiability question is answered in [2]. In order to state the result, we introduce notation for block-Hankel matrix constructed from a time series

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

Lemma 1 (Fundamental lemma [2]). *Let*

1) $w_\mathrm{d} = (u_\mathrm{d}, y_\mathrm{d})$ *be a T samples long trajectory of the LTI system $\mathscr{B}$ of order* $\mathtt{n}$, i.e.,
$$w_\mathrm{d} = \begin{bmatrix} u_\mathrm{d} \\ y_\mathrm{d} \end{bmatrix} = \left(\begin{bmatrix} u_\mathrm{d}(1) \\ y_\mathrm{d}(1) \end{bmatrix}, \dots, \begin{bmatrix} u_\mathrm{d}(T) \\ y_\mathrm{d}(T) \end{bmatrix} \right) \in \mathscr{B}|_{[1,T]};$$
2) *the system $\mathscr{B}$ be controllable; and*
3) *the input sequence u_d be persistently exciting of order* $\Delta + \mathtt{n}$, i.e., *the matrix $\mathscr{H}_{\Delta+\mathtt{n}}(u_\mathrm{d})$ is of full row rank.*

Then any Δ samples long trajectory $w = (u,y)$ of $\mathscr{B}$ can be written as a linear combination of the columns of $\mathscr{H}_\Delta(w_\mathrm{d})$ and any linear combination $\mathscr{H}_\Delta(w_\mathrm{d})g$, $g \in \mathbb{R}^{T-\Delta+1}$, is also a trajectory of $\mathscr{B}$, i.e.,

$$\operatorname{col\,span}\big(\mathscr{H}_\Delta(w_\mathrm{d})\big) = \mathscr{B}|_{[1,\Delta]}.$$

The fundamental lemma gives conditions under which the Hankel matrix $\mathscr{H}_\Delta(w_\mathrm{d})$ has the "correct" image (and as a consequence the "correct" left kernel). For sufficiently large Δ it answers the identifiability question.

Theorem 1 (Identifiability condition). *The system $\mathscr{B} \in \mathscr{L}^\mathtt{w}$ is identifiable from the exact data $w_\mathrm{d} = (u_\mathrm{d}, y_\mathrm{d}) \in \mathscr{B}$ if $\mathscr{B}$ is controllable and an input component u_d is persistently exciting of order $\mathtt{L}+1+\mathtt{n}$, where $\mathtt{n}$ is the order and $\mathtt{L}$ is the lag (the observability index) of $\mathscr{B}$.*

The identifiability conditions stated above actually guarantee that $\mathscr{B}_\mathrm{mpum}(w_\mathrm{d})$ coincides with the data generating system. Therefore under these conditions, by computing the MPUM, we recover the data generating system from data.

The exact identification problem has been studied in the literature under the name deterministic system identification, see [3, Section 2]. Two main classes of algorithms have been proposed in the literature:

- MOESP: $w_\mathrm{d} \mapsto$ an observability matrix $\mapsto (A,B,C,D)$, and
- N4SID: $w_\mathrm{d} \mapsto$ a state sequence $\mapsto (A,B,C,D)$.

The four mappings indicated above are tractable in the sense that their algorithmic implementations are solutions of linear systems of equations. In this paper, we consider the N4SID-type algorithms. Let x_d be a minimal state sequence of the data generating system corresponding to $w_\mathrm{d} = (u_\mathrm{d}, y_\mathrm{d})$. The mapping $x_\mathrm{d} \mapsto (A,B,C,D)$ is implemented by solving the linear system of equations

$$\begin{bmatrix} \sigma x_\mathrm{d} \\ y_\mathrm{d} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x_\mathrm{d} \\ u_\mathrm{d} \end{bmatrix}.$$

In the rest of the paper, we discuss the mapping from data to a state sequence $w_\mathrm{d} \mapsto x_\mathrm{d}$. Our contribution is a new algorithm for computation of a state sequence of the MPUM from data. It is an application of the shift-and-cut map proposed in [4]. The existing algorithms for computation of a state sequence from data, *e.g.*, the oblique projection, split the data into "past" and "future" parts. The proposed algorithm does not use such a splitting and because of this has computational advantages.

II. State construction via intersection of past and future

The first deterministic subspace identification algorithms [5], [6] are based on [1, Proposition 20], which gives a formula for the minimal state dimension of the MPUM. The result is for infinite time series and states that if the four times infinite Hankel matrix of the data is partitioned into "past" and "future" submatrices as follows:

$$\left[\frac{\mathscr{H}_{\mathrm{p}}}{\mathscr{H}_{\mathrm{f}}}\right] := \left[\begin{array}{ccccc} & \vdots & \vdots & \vdots & \\ \cdots & w_{\mathrm{d}}(-2) & w_{\mathrm{d}}(-1) & w_{\mathrm{d}}(0) & \cdots \\ \cdots & w_{\mathrm{d}}(-1) & w_{\mathrm{d}}(0) & w_{\mathrm{d}}(1) & \cdots \\ \hline \cdots & w_{\mathrm{d}}(0) & w_{\mathrm{d}}(1) & w_{\mathrm{d}}(2) & \cdots \\ \cdots & w_{\mathrm{d}}(1) & w_{\mathrm{d}}(2) & w_{\mathrm{d}}(3) & \cdots \\ & \vdots & \vdots & \vdots & \end{array}\right],$$

the minimal state dimension of $\mathscr{B}_{\mathrm{mpum}}(w_{\mathrm{d}})$ is equal to

$$\mathtt{n} = \operatorname{rank}(\mathscr{H}_{\mathrm{p}}) + \operatorname{rank}(\mathscr{H}_{\mathrm{f}}) - \operatorname{rank}(\mathscr{H}).$$

Moreover, a basis for the intersection of the row spans of $\mathscr{H}_{\mathrm{p}}$ and $\mathscr{H}_{\mathrm{f}}$ is a minimal state sequence x_{d} of the MPUM, corresponding to the time series w_{d}.

The past–future intersection property is utilized in [6], where the two stage SVD procedure, outlined in Algorithm 1, is proposed.

Algorithm 1 Computation of a state sequence from data by past–future intersection [6].

Input: $w_{\mathrm{d}} = (u_{\mathrm{d}}, y_{\mathrm{d}})$ — an exact trajectory of a system $\mathscr{B} \in \mathscr{L}^{\mathtt{w}}$, and $\Delta \in \mathbb{Z}$, $\Delta \geq \mathtt{L}$ (the lag of $\mathscr{B}$).

1: Compute the SVD

$$U\Sigma V^{\top} = \mathscr{H}_{2\Delta}(w_{\mathrm{d}}) =: \begin{bmatrix} W_{\mathrm{p}} \\ W_{\mathrm{f}} \end{bmatrix} \begin{matrix} \mathtt{w}\Delta \\ \mathtt{w}\Delta \end{matrix}$$

and define the partitionings

$$\begin{array}{c} \begin{matrix} \mathtt{w}\Delta & \mathtt{w}\Delta \end{matrix} \\ \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \end{array} \begin{matrix} \mathtt{w}\Delta \\ \mathtt{w}\Delta \end{matrix} := \begin{array}{c} \begin{matrix} \mathtt{w}\Delta & \mathtt{w}\Delta \end{matrix} \\ \begin{bmatrix} U_1 & U_2 \end{bmatrix} \end{array} := U.$$

2: Compute the matrix $\tilde{X} := U_{12}^{\top} W_{\mathrm{p}}$.

3: Compute the SVD

$$\bar{U}\bar{\Sigma}\bar{V}^{\top} = \tilde{X},$$

define $\mathtt{n} := \operatorname{rank}(U_{12}^{\top} W_{\mathrm{p}})$, and let

$$\begin{bmatrix} \bar{U}_1 \\ \bar{U}_2 \end{bmatrix} \begin{matrix} \mathtt{n} \\ \mathtt{w}\Delta - \mathtt{n} \end{matrix} := \bar{U}.$$

Output: $\bar{U}_1$ — a minimal state sequence of $\mathscr{B}$.

Let $\mathtt{m}$ be the number of inputs and $\mathtt{p} = \mathtt{w} - \mathtt{m}$ be the number of outputs of $\mathscr{B}$. In Algorithm 1 (as well as in the fundamental lemma) the input/output partitioning $w_{\mathrm{d}} = (u_{\mathrm{d}}, y_{\mathrm{d}})$ of the given time series is assumed. In particular $\mathtt{m} = \dim\big(u_{\mathrm{d}}(t)\big)$ is known.

Define the (finite) input and output, "past" and "future" matrices

$$\begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \end{bmatrix} \begin{matrix} \mathtt{m}\Delta \\ \mathtt{m}\Delta \end{matrix} := \mathscr{H}_{2\Delta}(u_{\mathrm{d}}), \qquad \begin{bmatrix} Y_{\mathrm{p}} \\ Y_{\mathrm{f}} \end{bmatrix} \begin{matrix} \mathtt{p}\Delta \\ \mathtt{p}\Delta \end{matrix} := \mathscr{H}_{2\Delta}(y_{\mathrm{d}}),$$

the matrix of the "past" state sequence

$$X_{\mathrm{p}} := \begin{bmatrix} x_{\mathrm{d}}(1) & \cdots & x_{\mathrm{d}}(T - 2\Delta + 1) \end{bmatrix}.$$

and the matrix of the "future" state sequence

$$X_{\mathrm{f}} := \begin{bmatrix} x_{\mathrm{d}}(\Delta + 1) & \cdots & x_{\mathrm{d}}(T - \Delta + 1) \end{bmatrix}.$$

In [6], it is shown that the intersection property

$$\operatorname{row\,span}(X_{\mathrm{f}}) = \operatorname{row\,span}(W_{\mathrm{p}}) \cap \operatorname{row\,span}(W_{\mathrm{f}}) \tag{1}$$

holds under the rank condition

$$\operatorname{rank}\left(\begin{bmatrix} X_{\mathrm{p}} \\ U_{\mathrm{p}} \end{bmatrix}\right) = \mathtt{n} + \Delta\mathtt{m}. \tag{2}$$

Algorithm 1 computes a basis for the intersection space (1) as follows. The first SVD is used for the computation of a basis for the left null space of the data matrix $\mathscr{H}_{2\Delta}(w_{\mathrm{d}})$—the columns of U_2 provide such a basis. From

$$\tilde{X} := U_{12}^{\top} W_{\mathrm{p}} = -U_{22}^{\top} W_{\mathrm{f}},$$

it follows that

$$\operatorname{row\,span}(\tilde{X}) = \operatorname{row\,span}(W_{\mathrm{p}}) \cap \operatorname{row\,span}(W_{\mathrm{f}}),$$

so that the columns of $\tilde{X}$ form a state sequence. In general, however, it is nonminimal. The second SVD gives a minimal state sequence by computing a basis for $\operatorname{row\,span}(\tilde{X})$, *i.e.*, there is a nonsingular matrix $S \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}$, such that

$$\bar{U}_1 = S X_{\mathrm{f}}, \qquad \det(S) \neq 0.$$

Comments

Proposition 20 of [1] assumes an infinite time series w_{d}. The results of [6] apply for finite time series but relay on the unverifiable from the data w_{d} assumption (2). Based on the fundamental lemma, we give verifiable from the data conditions under which (2) holds.

Corollary 1. *Assume that $\sigma x = Ax + Bu$ is controllable and consider a trajectory $u_{\mathrm{d}}(1), \ldots, u_{\mathrm{d}}(T)$, $x_{\mathrm{d}}(1), \ldots, x_{\mathrm{d}}(T)$ of this system. Then if u_{d} is persistently exciting of order $\mathtt{n} + \Delta$, condition (2) holds.*

Proof: See [2, Corollary 1, item iii)]. □

Note that by choosing $\Delta = \mathtt{L}$, *i.e.*, when the lag of the data generating system is a priori known, the condition of Corollary 1 is equivalent to the identifiability condition of Theorem 1.

III. State construction via the shift-and-cut map

As shown in [7], the kernel of a block-Hankel matrix is highly structured. Let the columns of N_1 be in the left kernel of $\mathscr{H}_{\Delta_1}(w_\mathrm{d})$, *i.e.*,

$$N_1^\top \mathscr{H}_{\Delta_1}(w_\mathrm{d}) = 0$$

and let $N_{1,i} \in \mathbb{R}^{\mathtt{w}\times \operatorname{coldim}(N_1)}$ be the ith block element of N_1. Then for $\Delta_2 > \Delta_1$, the columns of

$$N_2 := \begin{bmatrix} N_{1,0} & 0 & \cdots & 0 \\ N_{1,1} & N_{1,0} & \ddots & \vdots \\ \vdots & N_{1,1} & \ddots & 0 \\ N_{1,\Delta_1-1} & \vdots & \ddots & N_{1,0} \\ 0 & N_{1,\Delta_1-1} & & N_{1,1} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & N_{1,\Delta_1-1} \end{bmatrix}, \tag{3}$$

are in the left kernel of $\mathscr{H}_{\Delta_2}(w_\mathrm{d})$, *i.e.*,

$$N_2^\top \mathscr{H}_{\Delta_2}(w_\mathrm{d}) = 0.$$

However, extra conditions are needed for the stronger statement

$$\begin{aligned}\operatorname{col\,span}(N_1) &= \operatorname{left\,ker}\big(\mathscr{H}_{\Delta_1}(w_\mathrm{d})\big) \\ &\implies \operatorname{col\,span}(N_2) = \operatorname{left\,ker}\big(\mathscr{H}_{\Delta_2}(w_\mathrm{d})\big).\end{aligned} \tag{4}$$

Lemma 2. *Under the conditions of Theorem 1, if $\Delta_1 > \mathtt{L}$, then (4) holds.*

Proof: Clearly

$$\operatorname{col\,span}(N_2) \subseteq \operatorname{left\,ker}\big(\mathscr{H}_{\Delta_2}(w_\mathrm{d})\big).$$

In order to show that under the extra conditions equality holds, define the matrix polynomial

$$N_1(z) := \sum_{i=0}^{\Delta_1-1} N_1 z^i.$$

By Theorem 1 and with $\Delta_1 > \mathtt{L}$,

$$\mathscr{B} = \ker\big(N_1(\sigma)\big) := \{\, w \in (\mathbb{R}^{\mathtt{w}})^{\mathbb{Z}} \mid N_1(\sigma)w = 0 \,\}. \tag{5}$$

Assume by contradiction that there is a vector $n \in \mathbb{R}^{\Delta_2 \mathtt{w}}$ such that $n^\top \mathscr{H}_{\Delta_2}(w_\mathrm{d}) = 0$, but $n \notin \operatorname{col\,span}(N_2)$. From the Sylvester resultant matrix $\begin{bmatrix} n & N_2 \end{bmatrix}$, it follows that the polynomial $n(z) := \sum_{i=0}^{\Delta_2-1} N_i z^i \in \mathbb{R}^{\mathtt{w}}[z]$ associated with n is coprime with $N_1(z)$. This contradicts to (5). $\square$

Lemma 2 shows that the knowledge of $\operatorname{left\,ker}\big(\mathscr{H}_{\Delta}(w_\mathrm{d})\big)$, $\Delta > \mathtt{L}$, suffices to construct $\operatorname{left\,ker}\big(\mathscr{H}_{2\Delta}(w_\mathrm{d})\big)$, used in the intersection algorithms. This offers computational savings since the kernel of a half deeper matrix needs to be computed. In analogy with Algorithm 1, using this approach, we obtain Algorithm 2. Note that for the smallest possible value $\mathtt{L}+1$ of the parameter Δ, the assumptions of Lemma 2 are consistent with the identifiability conditions of Corollary 1.

The operation in step 2 of Algorithm 1 is the shift-and-cut map introduced in [4].

Algorithm 2 Computation of a state sequence from data without past–future intersection.

Input: $w_\mathrm{d} = (u_\mathrm{d}, y_\mathrm{d})$ — an exact trajectory of a system $\mathscr{B} \in \mathscr{L}^{\mathtt{w}}$, and $\Delta \in \mathbb{Z}$, $\Delta \geq \mathtt{L}$ (the lag of $\mathscr{B}$).

1: Compute a basis N for the left kernel of $\mathscr{H}_\Delta(w_\mathrm{d})$ and let $N_i \in \mathbb{R}^{\mathtt{w}}$ be the ith block element of N.

2: Compute the matrix

$$\tilde{X} := \begin{bmatrix} N_0^\top & N_1^\top & \cdots & N_{\Delta-2}^\top \\ 0 & N_0^\top & \cdots & N_{\Delta-3}^\top \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & N_0^\top \end{bmatrix} \mathscr{H}_{\Delta-1}(\sigma w_\mathrm{d})$$

3: Compute the SVD

$$\bar{U}\bar{\Sigma}\bar{V}^\top = \tilde{X},$$

define $\mathtt{n} := \operatorname{rank}(U_{12}^\top W_\mathrm{p})$, and let

$$\begin{array}{cc} \begin{bmatrix} \bar{U}_1 \\ \bar{U}_2 \end{bmatrix} & \begin{array}{c} \mathtt{n} \\ \mathtt{w}\Delta - \mathtt{n} \end{array} \end{array} := \bar{U}.$$

Output: $\bar{U}_1$ — a minimal state sequence of $\mathscr{B}$.

For concreteness one can assume that the kernel computation on step 1 of Algorithm 2 is carried out by the SVD

$$U\Sigma V^\top = \mathscr{H}_\Delta(w_\mathrm{d}).$$

This assumption makes Algorithm 2 match Algorithm 1 (2 SVDs plus one matrix–matrix product) with the important difference that Algorithm 2 operates on matrices of depth Δ, while Algorithm 1 operates on matrices of depth 2Δ. This is the reason for the improved numerical efficiency of Algorithm 2.

IV. Numerical efficiency

The cost for the restricted SVD of an $m \times n$, $m \leq n$ matrix, assuming that only the V factor (left null space) is needed is proportional to $2m^2 n + 11m^3$, see [8, Section 5.4.5]. In system identification, typically $T \gg \mathtt{w}\Delta$, so that

$$j := T - 2\Delta + 1 \approx T - \Delta + 1.$$

With this approximation the cost per step for Algorithm 1 is:

1: $8(\mathtt{w}\Delta)^2 j + 88(\mathtt{w}\Delta)^3$
2: $(\mathtt{w}\Delta)^2 j$
3: $2(\mathtt{w}\Delta)^2 j + 11(\mathtt{w}\Delta)^3$

while the cost per step for Algorithm 2 is:

1: $2(\mathtt{w}\Delta)^2 j + 11(\mathtt{w}\Delta)^3$
2: $\big(\mathtt{w}(\Delta-1)\big)^2 j$
3: $2\big(\mathtt{w}(\Delta-1)\big)^2 j + 11\big(\mathtt{w}(\Delta-1)\big)^3$

We have in total $11(\mathtt{w}\Delta)^2 j + 99(\mathtt{w}\Delta)^3$ flops for Algorithm 1 and less than $5(\mathtt{w}\Delta)^2 j + 22(\mathtt{w}\Delta)^3$ flops for Algorithm 2. In the following section, we show the empirical number of floating point operations for a simulation example in which $\mathtt{w}\Delta = 10$ and $T = 100$, so that $T \not\gg \mathtt{w}\Delta$.

V. Simulation example

Consider a random stable system $\mathscr{B}$ of order $\mathtt{n} = 6$, with $\mathtt{m} = 3$ inputs, and $\mathtt{p} = 2$ outputs. The data available for identification is a $T = 100$ samples long random trajectory w_{d} of $\mathscr{B}$. Both the classical intersection algorithm and the new algorithm, proposed in Section III recover the system $\mathscr{B}$ back from the data w_{d} exactly up to the numerical errors due to the finite precision arithmetic. In the example:

$$\|\mathscr{B} - \hat{\mathscr{B}}_1\|_\infty = 2.4 \times 10^{-15} \text{ and } \|\mathscr{B} - \hat{\mathscr{B}}_2\|_\infty = 4.3 \times 10^{-15},$$

where $\hat{\mathscr{B}}_1$ and $\hat{\mathscr{B}}_2$ are respectively the systems obtained via Algorithms 1 and 2.

The computational requirements of the two algorithms, however, are different. For the considered example, the floating point operations counts given by MATLAB 5.0 are:

$$f_1 = 1.9 \times 10^6 \quad \text{and} \quad f_2 = 1.3 \times 10^6$$

for respectively Algorithms 1 and 2.

Note 1. Both algorithms are implemented as given in the paper and are not optimized in efficiency. In particular, the left null space is computed directly from the SVD of the Hankel matrix. Efficient implementation would first compute the R factor of the QR decomposition of the Hankel matrix and then the SVD of a sub-block of R.

VI. Conclusions

Two algorithms for state construction of the MPUM were presented. The first one is a classical intersection algorithm from the literature. However, using the fundamental lemma, we completed the presentation of [6] with a verifiable from the data condition for identifiability. The second algorithm (partially) uses the structure of the Hankel matrix in order to compute the past–future intersection more efficiently. The algorithm has a strong relation to the shift-and-cut map which is used for state construction.

The result of the paper shows that the classical subspace identification algorithms do not exploit the rich structure of the kernel of a block-Hankel matrix. Algorithm 2 can be improved by proceeding recursively over the block rows of the Hankel matrix $\mathscr{H}_\Delta(w_{\mathrm{d}})$ in constructing its left kernel recursively.

VII. Acknowledgments

This research is supported by the Belgian Federal Government under the DWTC program Interuniversity Attraction Poles, Phase V, 2002–2006, Dynamical Systems and Control: Computation, Identification and Modelling, by the KUL Concerted Research Action (GOA) MEFISTO–666, and by several grants en projects from IWT-Flanders and the Flemish Fund for Scientific Research; HPC-EUROPA (RII3-CT-2003-506079), with the support of the European Community — Research Infrastructure Action under the FP6 "Structuring the European Research Area" Program.

References

[1] J. C. Willems, "From time series to linear system—Part II. Exact modelling," *Automatica*, vol. 22, no. 6, pp. 675–694, 1986.

[2] J. C. Willems, P. Rapisarda, I. Markovsky, and B. D. Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.

[3] P. Van Overschee and B. De Moor, *Subspace Identification for Linear Systems: Theory, Implementation, Applications*. Kluwer, 1996.

[4] P. Rapisarda and J. C. Willems, "State maps for linear systems," *SIAM J. Contr. Optim.*, vol. 35, no. 3, pp. 1053–1091, 1997.

[5] B. De Moor, "Mathematical concepts for modeling of static and dynamic systems," Ph.D. dissertation, Dept. EE, K.U.Leuven, Belgium, 1988.

[6] M. Moonen, B. D. Moor, L. Vandenberghe, and J. Vandewalle, "On- and off-line identification of linear state-space models," *Int.J. Control*, vol. 49, pp. 219–232, 1989.

[7] G. Heinig and P. Jankowski, "Kernel structure of block Hankel and Toeplitz matricies and partial realization," *Lin. Alg. and Its Appl.*, vol. 175, pp. 1–30, 1992.

[8] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed. Johns Hopkins University Press, 1996.