STATE CONSTRUCTION IN SUBSPACE IDENTIFICATION

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

ESAT, SCD-SISTA, K.U.Leuven,
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

{Jan.Willems,Ivan.Markovsky,Bart.DeMoor}@esat.kuleuven.be

Abstract: In this presentation, we consider the problem of obtaining the state trajectory directly from an observed vector time-series. We show how the Hankel structure of the data matrix can be exploited in this construction. Both the cases of infinite as well as finite time-series are considered, but only deterministic systems are discussed.

Keywords: Most powerful unfalsified model, subspace identification, state construction.

1. INTRODUCTION

It is pedagogically orthodox to deal with SYSID theory by going from the 'simple' to the 'complex'. We consider the case in which the observed data is assumed to be exact ('noise free') as simple, and the (approximate) identification of stochastic models, the central problem considered in the popular textbooks on the subject, as complex. In this article, we reexamine some aspects of exact SYSID.

The theory of exact SYSID centers around the notion of the most powerful unfalsified model (MPUM). This is the model that explains the observations exactly, but infers as little else as possible about the system that generated the data (in a sense that will be made precise later). We consider as the model class the deterministic linear time-invariant (LTI) systems. In this context it is convenient to assume in first instance that an infinite (vector) time-series is observed, since it is then easy to see that the MPUM exists.

A very effective class of SYSID algorithms goes under the name of *subspace algorithms*. They came to the foreground in the 1990's (Van Overschee and De Moor, 1994; Van Overschee and De Moor, 1996). The basic idea is to construct first the state trajectory that corresponds to the observed time-series, and subsequently deduce the system parameter matrices of a state space model from the input/state/output trajectory. These algorithms have very nice properties from a numerical and model simplification point of view. They rely on standard numerical linear algebra routines, allow for reduction of the state space dimension, and fitting of the model parameters comes down to a classical least squares problem. Usually in these algorithms, the state trajectory is obtained by setting up the Hankel matrix of the data, splitting it in the 'past' and the 'future', and deducing the 'present' state trajectory by considering the intersection or projection of the future and the past. Different algorithms vary in this aspect, but the past/future separation is central.

The purpose of this article is two-fold. The main goal is to show that by exploiting the Hankel structure of the data matrix, it is possible to avoid the past/future separation. The second goal is to develop the MPUM in the case that the observed time-series is finite.

We use standard notation as $\mathbb{R}, \mathbb{N}, \mathbb{Z}, \mathbb{Z}_+, \mathbb{R}^{\mathtt{n}}, \mathbb{R}^{\mathtt{n}_1 \times \mathtt{n}_2}$, etc. $\mathbb{R}[\xi]$ denotes the polynomials with real coefficients and $\mathbb{R}^{\mathtt{n}_1 \times \mathtt{n}_2}[\xi]$ the polynomial matrices with real $\mathtt{n}_1 \times \mathtt{n}_2$ matrices as coefficients. When the number of rows is immaterial and the number of columns is $\mathtt{n}$, we write $\mathbb{R}^{\bullet \times \mathtt{n}}[\xi]$. Similar for polynomial vectors, or when the number of rows and/or columns is irrelevant. The degree of $P \in \mathbb{R}^{\bullet \times \bullet}[\xi]$ equals the largest degree of its entries, and is denoted as $\text{degree}(P)$.

2. LTI SYSTEMS

We view a model as a subset, $\mathscr{B}$, of a universum, $\mathscr{U}$, of a priori possibilities. $\mathscr{B} \subseteq \mathscr{U}$ is called the *behavior* of the model. In *dynamical* systems, the model consists of maps from a set of relevant time instances to the *signal space*. We assume that the set of time instances is $\mathbb{N}$ and that the signal space is $\mathbb{R}^{\mathtt{w}}$. The behavior is then a set $\mathscr{B}$ of maps from $\mathbb{N}$ into $\mathbb{R}^{\mathtt{w}}$, $\mathscr{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$.

A behavior $\mathscr{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ is said to be *linear* if $w \in \mathscr{B}, w' \in \mathscr{B}$, and $\alpha \in \mathbb{R}$ imply $w + w' \in \mathscr{B}$ and $\alpha w \in \mathscr{B}$, and *time-invariant* if $\sigma\mathscr{B} \subseteq \mathscr{B}$. The left *shift*, σ, is defined by $(\sigma f)(\mathtt{t}) := f(\mathtt{t}+1)$. The third property that enters into the specification of the model class is completeness. $\mathscr{B}$ is called *complete* if it has the following property:

$$\big[w : \mathbb{N} \to \mathbb{R}^{\mathtt{w}} \text{ belongs to } \mathscr{B}\big] \Leftrightarrow \big[w|_{[1,\mathtt{t}]} \in \mathscr{B}|_{[1,\mathtt{t}]} \text{ for all } \mathtt{t} \in \mathbb{N}\big].$$

In words: $\mathscr{B}$ is complete if $w : \mathbb{N} \to \mathbb{R}^{\mathtt{w}}$ is 'legal' (i.e. $\in \mathscr{B}$) iff everyone of its 'prefixes' $(w(1), \ldots, w(\mathtt{t}))$ is 'legal' (i.e. $\in \mathscr{B}|_{[1,\mathtt{t}]}$).

The combination of linearity, time-invariance, and completeness can be expressed in many equivalent ways. In particular, the following are equivalent:

(1) $\mathscr{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ is linear, time-invariant, and complete;
(2) $\mathscr{B}$ is a linear, shift invariant, closed (in the topology of pointwise convergence) subset of $(\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$;
(3) $\exists\, R \in \mathbb{R}^{\bullet \times \mathtt{w}}[\xi]$ such that $\mathscr{B}$ consists of the solutions $w : \mathbb{N} \to \mathbb{R}^{\mathtt{w}}$ of

$$R(\sigma)w = 0. \qquad (\mathfrak{Ker})$$

The set of behaviors $\mathscr{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ that satisfy the equivalent conditions 1. to 3. is denoted by $\mathscr{L}^{\mathtt{w}}$, or, when the number of variables is unspecified, by $\mathscr{L}^{\bullet}$. Since $\mathscr{B} = \ker(R(\sigma))$ in $(\mathfrak{Ker})$, we call $(\mathfrak{Ker})$ a *kernel representation* of the behavior $\mathscr{B}$.

We now introduce a characterization that is mathematically more abstract, but very pertinent in the context of SYSID. It identifies a behavior $\mathscr{B} \in \mathscr{L}^{\bullet}$ with an $\mathbb{R}[\xi]$-module.

Consider $\mathscr{B} \in \mathscr{L}^{\mathtt{w}}$. The polynomial vector $n \in \mathbb{R}^{1\times \mathtt{w}}[\xi]$ is called an *annihilator* of $\mathscr{B}$ if $n(\sigma)\mathscr{B} = 0$, i.e. if $n(\sigma)w = 0$ for all $w \in \mathscr{B}$. Denote by $\mathscr{N}_{\mathscr{B}}$ the set of annihilators for $\mathscr{B}$. Observe that $\mathscr{N}_{\mathscr{B}}$ is an $\mathbb{R}[\xi]$-module. Hence the map $\mathscr{B} \mapsto \mathscr{N}_{\mathscr{B}}$ associates with each $\mathscr{B} \in \mathscr{L}^{\mathtt{w}}$ a submodule of $\mathbb{R}^{1\times \mathtt{w}}[\xi]$. It is possible to prove that this map is a bijection, i.e. each submodule of $\mathbb{R}^{1\times \mathtt{w}}[\xi]$ is the set of annihilators of exactly one element of $\mathscr{L}^{\mathtt{w}}$. This characterization of $\mathscr{B}$ in terms of its module of annihilators is very useful in the context of deterministic SYSID. It shows precisely what we are looking for in order to identify a system in the model class $\mathscr{L}^{\bullet}$: (a set of generators of) the submodule $\mathscr{N}_{\mathscr{B}}$.

Behaviors in $\mathscr{L}^{\bullet}$ admit many other representations. The following is a very familiar one:

4) $\mathscr{B} \in \mathscr{L}^{\mathtt{w}} \Leftrightarrow \exists$ integers $\mathtt{m}, \mathtt{p}, \mathtt{n} \in \mathbb{Z}_+$ with $\mathtt{m} + \mathtt{p} = \mathtt{w}$, matrices $A \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}, B \in \mathbb{R}^{\mathtt{n}\times\mathtt{m}}, C \in \mathbb{R}^{\mathtt{p}\times\mathtt{n}}, D \in \mathbb{R}^{\mathtt{p}\times\mathtt{m}}$, and a permutation matrix $\Pi \in \mathbb{R}^{\mathtt{w}\times\mathtt{w}}$ such that $\mathscr{B}$ consists of all $w : \mathbb{N} \to \mathbb{R}^{\mathtt{w}}$ for which there exist $u : \mathbb{N} \to \mathbb{R}^{\mathtt{m}}$, $x : \mathbb{N} \to \mathbb{R}^{\mathtt{n}}$, and $y : \mathbb{N} \to \mathbb{R}^{\mathtt{p}}$ such that

$$\sigma x = Ax + Bu, \;\; y = Cx + Du, \;\; w = \Pi \begin{bmatrix} u \\ y \end{bmatrix}. \quad (\mathfrak{S})$$

The representation $(\mathfrak{S})$ is called an *input/state/output* (i/s/o) representation of the corresponding behavior $\mathscr{B} \in \mathscr{L}^{\mathtt{w}}$. It is said to be *minimal* if the dimension of the state space is minimal over all i/s/o representations $(\mathfrak{S})$ of $\mathscr{B}$. Minimality is equivalent to state observability (defined in the classical way), but does not require state controllability.

Certain integer 'invariants' (meaning maps from $\mathscr{L}^{\bullet}$ to $\mathbb{Z}_+$) are important in the sequel. One is the *lag*, $\mathtt{L}(\mathscr{B})$, defined as the smallest $\mathtt{L} \in \mathbb{Z}_+$ such that $\big[w|_{[\mathtt{t},\mathtt{t}+\mathtt{L}]} \in \mathscr{B}|_{[1,\mathtt{L}+1]} \text{ for all } \mathtt{t} \in \mathbb{N}\big] \Rightarrow \big[w \in \mathscr{B}\big]$. Equivalently, the smallest degree over the polynomial matrices R such that $\mathscr{B} = \ker(R(\sigma))$. A second integer invariant is the *input cardinality*, denoted by $\mathtt{m}(\mathscr{B})$, defined as $\mathtt{m}$, the number of input variables in any $(\mathfrak{S})$ representation of $\mathscr{B}$. The number of output variables, $\mathtt{p}$, yields the *output cardinality* $\mathtt{p}(\mathscr{B})$. A third important integer invariant is the *state cardinality*, $\mathtt{n}(\mathscr{B})$, defined as the smallest number $\mathtt{n}$ of state variables over all i/s/o representations $(\mathfrak{S})$ of $\mathscr{B}$.

In many areas of system theory, controllability enters as a regularizing assumption. In the behavioral theory, an appealing notion of controllability has been put forward. It expresses what is needed intuitively, it applies to any dynamical system, regardless of its representation, and it has the classical state transfer definition as a special case. Adapted to the case at hand, it reads as follows. The time-invariant behavior $\mathscr{B} \subseteq (\mathbb{R}^{\bullet})^{\mathbb{N}}$ is said to be *controllable* if for any $w_1 \in \mathscr{B}$, $w_2 \in \mathscr{B}$, and $\mathtt{t}_1 \in \mathbb{N}$, there exists a $\mathtt{t}_2 \in \mathbb{N}$ and a $w \in \mathscr{B}$ such that $w(\mathtt{t}) = w_1(\mathtt{t})$ for $1 \le \mathtt{t} \le \mathtt{t}_1$, and $w(\mathtt{t}) = w_2(\mathtt{t} - \mathtt{t}_1 - \mathtt{t}_2)$ for $\mathtt{t} > \mathtt{t}_1 + \mathtt{t}_2$. It can be shown that $\mathscr{B} \in \mathscr{L}^{\bullet}$ is controllable iff it has a state space representation $(\mathfrak{S})$ that is controllable (more precisely, reachable) in the classical sense.

$\mathscr{B} \subseteq (\mathbb{R}^{\bullet})^{\mathbb{N}}$ is said to be *homogeneous* if $\sigma\mathscr{B} = \mathscr{B}$. This implies time-invariance. A (time-invariant) controllable behavior is automatically homogeneous.

3. THE MPUM FOR INFINITE TIME-SERIES

We discuss only SYSID algorithms that associate a deterministic LTI model *exactly* to an observed time-series. The idea is to obtain the least complex model that explains the data. The concept which implements this is the Most Powerful Unfalsified Model (MPUM). This notion was introduced in (Willems, 1986, 1987, Part II) for infinite time-series. We discuss the infinite case first. Later we will adapt it to finite time-series.

Assume that the time-series

$$\tilde{w}_\infty = (\tilde{w}(1), \tilde{w}(2), \ldots, \tilde{w}(\mathtt{t}), \ldots),\ \tilde{w}(\mathtt{t}) \in \mathbb{R}^{\mathtt{w}}$$

is observed. The behavior $\mathscr{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ is *unfalsified* by $\tilde{w}_\infty$ if $\tilde{w}_\infty \in \mathscr{B}$. $\mathscr{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ is *more powerful* than $\mathscr{B}' \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ if $\mathscr{B} \subseteq \mathscr{B}'$. Hence, *the more a model forbids, the better it is*. $\mathscr{B} \in \mathscr{L}^{\mathtt{w}}$ is the *MPUM in* $\mathscr{L}^{\mathtt{w}}$ *for* $\tilde{w}_\infty$ if it is unfalsified by $\tilde{w}_\infty$ and more powerful than any other behavior in $\mathscr{L}^{\mathtt{w}}$ that is also unfalsified by $\tilde{w}_\infty$.

For an *infinite* observed time-series, it is easy to prove that this MPUM always exists. In fact, it is equal to the closure (in the topology of pointwise convergence) of $\mathrm{span}\{\sigma^{\mathtt{t}}\tilde{w}_\infty, \mathtt{t} \in \mathbb{Z}_+\}$. Indeed, this closure is obviously linear, time-invariant and closed, and hence, by the characterization 2) of the previous section, a behavior in $\mathscr{L}^{\mathtt{w}}$. Moreover, $\tilde{w}_\infty$ is unfalsified by it, and, since any unfalsified behavior in $\mathscr{L}^{\mathtt{w}}$ must necessarily contain all the $\sigma^{\mathtt{t}}\tilde{w}_\infty$ and be closed, it is more powerful than any other unfalsified behavior. Denote this MPUM by $\mathscr{B}^\star(\tilde{w}_\infty)$.

The algorithms that compute $\mathscr{B}^\star(\tilde{w}_\infty)$ from $\tilde{w}_\infty$ invariably consider the Hankel matrix of the data, defined as the doubly infinite matrix

$$\mathscr{H}(\tilde{w}_\infty) := \begin{bmatrix} \tilde{w}_\infty(1) & \tilde{w}_\infty(2) & \cdots & \tilde{w}_\infty(\mathtt{t}'') & \cdots \\ \tilde{w}_\infty(2) & \tilde{w}_\infty(3) & \cdots & \tilde{w}_\infty(\mathtt{t}''+1) & \cdots \\ \vdots & \vdots & \vdots\vdots & \vdots & \vdots\vdots \\ \tilde{w}_\infty(\mathtt{t}') & \tilde{w}_\infty(\mathtt{t}'+1) & \cdots & \tilde{w}_\infty(\mathtt{t}'+\mathtt{t}''-1) & \cdots \\ \vdots & \vdots & \vdots\vdots & \vdots & \ddots \end{bmatrix}$$

A kernel representation of $\mathscr{B}^\star(\tilde{w}_\infty)$ can be obtained by computing the (compact support elements in the) left kernel of $\mathscr{H}(\tilde{w}_\infty)$. The *left kernel* of a matrix $M \in \mathbb{R}^{\mathtt{n}_1 \times \mathtt{n}_2}$, denoted $\mathsf{leftker}(M)$, consists of the row vectors $n \in \mathbb{R}^{1 \times \mathtt{n}_1}$ that are annihilated by postmultiplication by M: $nM = 0$. In the case of $\mathscr{H}(\tilde{w}_\infty)$, it is useful to view the elements in the left kernel as vector polynomials, as follows. Assume that

$$n = \begin{bmatrix} n_0 & n_1 & \cdots & n_{\mathrm{degree}(n)} & 0 & \cdots & 0 & \cdots \end{bmatrix} \in \mathsf{leftker}(\mathscr{H}(\tilde{w}_\infty))$$

with the $n_{\mathtt{k}}$'s $\in \mathbb{R}^{1 \times \mathtt{w}}$. Associate with n the polynomial vector $n_0 + n_1\xi + \cdots + n_{\mathrm{degree}(n)}\xi^{\mathrm{degree}(n)} \in \mathbb{R}^{1 \times \mathtt{w}}[\xi]$, and denote it, with slight abuse of notation, also as $n(\xi)$. Then it is easily seen that the left kernel of $\mathscr{H}(\tilde{w}_\infty)$ is an $\mathbb{R}[\xi]$-module.

A kernel representation of $\mathscr{B}^\star(\tilde{w}_\infty)$ is obtained by computing a set of generators of this module. By more closely examining the Hankel structure of $\mathscr{H}(\tilde{w}_\infty)$, a set of generators may be obtained from a finite truncation of $\mathscr{H}(\tilde{w}_\infty)$ (see (Willems, 1986, 1987, Part II)). Here we only discuss truncations in the rows. In order to compute a set of generators, it suffices to consider the truncated Hankel matrix

$$\mathscr{H}(\tilde{w}_\infty)_{[1:\mathtt{d}]} := \begin{bmatrix} \tilde{w}_\infty(1) & \tilde{w}_\infty(2) & \cdots & \tilde{w}_\infty(\mathtt{t}'') & \cdots \\ \tilde{w}_\infty(2) & \tilde{w}_\infty(3) & \cdots & \tilde{w}_\infty(\mathtt{t}''+1) & \cdots \\ \vdots & \vdots & \vdots\vdots & \vdots & \vdots\vdots \\ \tilde{w}_\infty(\mathtt{d}) & \tilde{w}_\infty(\mathtt{d}+1) & \cdots & \tilde{w}_\infty(\mathtt{d}+\mathtt{t}''-1) & \cdots \end{bmatrix}$$

with $\mathtt{d} \geq \mathtt{L}(\mathscr{B}^\star(\tilde{w}_\infty))$, and compute a set of generators of the left kernel of $\mathscr{H}(\tilde{w}_\infty)_{[1:\mathtt{d}]}$. Since the cardinality of the set of generators of this left kernel equals $\mathtt{p}(\mathscr{B}^\star(\tilde{w}_\infty))$, this computation may be done more efficiently than by computing (a basis of) the whole left kernel of $\mathscr{H}(\tilde{w}_\infty)_{[1:\mathtt{d}]}$. We refer to (Willems, 1986, 1987, Part II), (**?**), (Van Overschee and De Moor, 1994; Van Overschee and De Moor, 1996), and the companion article (Markovsky *et al.*, 2005) for various (exact, approximate, recursive, etc.) algorithms that construct a kernel representation of $\mathscr{B}^\star(\tilde{w}_\infty)$.

4. SUBSPACE ID

Subspace identification algorithms use i/s/o models ($\mathfrak{S}$) in an essential way. Assume that from the observations $\tilde{w}_\infty$, we have identified the MPUM $\mathscr{B}^\star(\tilde{w}_\infty)$. We could then compute an input/state/output representation ($\mathfrak{S}$) for $\mathscr{B}^\star(\tilde{w}_\infty)$, and obtain the state trajectory corresponding to $\tilde{w}_\infty = \left[\begin{smallmatrix} u_\infty \\ y_\infty \end{smallmatrix}\right]$

$$\tilde{x}_\infty = (\tilde{x}(1), \tilde{x}(2), \ldots, \tilde{x}(\mathtt{t}), \ldots), \quad \tilde{x}(\mathtt{t}) \in \mathbb{R}^{\mathtt{n}(\mathscr{B}^\star(\tilde{w}_\infty))}.$$

This lays out the path to go from $\tilde{w}_\infty$ to (A,B,C,D), and from there, to $\tilde{x}_\infty$. These are related by

$$\begin{bmatrix} \tilde{x}(2) & \tilde{x}(3) & \cdots & \tilde{x}(\mathtt{t}+1) & \cdots \\ \tilde{y}(1) & \tilde{y}(2) & \cdots & \tilde{y}(\mathtt{t}) & \cdots \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \tilde{x}(1) & \tilde{x}(2) & \cdots & \tilde{x}(\mathtt{t}) & \cdots \\ \tilde{u}(1) & \tilde{u}(2) & \cdots & \tilde{u}(\mathtt{t}) & \cdots \end{bmatrix}. \quad (\mathscr{L})$$

The basic idea of subspace ID is the following. If we could somehow obtain *first* the state trajectory $\tilde{x}_\infty$, *directly from* $\tilde{w}_\infty$, we could subsequently solve the linear system of equations ($\mathscr{L}$) for the unknown parameter matrices $\left[\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right]$. This would yield a SYSID algorithm that identifies a state representation for $\mathscr{B}^\star(\tilde{w}_\infty)$. The major advantage of dealing with the equations ($\mathscr{L}$) is that, from a numerical linear algebra point of view, they lend themselves very well to approximations. Indeed, one can first reduce (using e.g. SVD) the state dimension by rank reduction of the matrix

$$\mathscr{X} = \begin{bmatrix} \tilde{x}(1) & \tilde{x}(2) & \cdots & \tilde{x}(\mathtt{t}) & \cdots \end{bmatrix}$$

and subsequently solve the (truncated) equations ($\mathscr{L}$) approximately for $\left[\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right]$ in a least squares (LS) sense. These numerical features are very well explained and efficiently exploited in (Van Overschee and De Moor, 1994; Van Overschee and De Moor, 1996).

The following question emerges:

How do we pass directly from $\tilde{w}_\infty$ to the corresponding state trajectory $\tilde{x}_\infty$ of $\mathscr{B}^\star(\tilde{w}_\infty)$?

4.1 State construction from past/future partition

An algorithm which achieves this state construction, under the assumption that $\mathscr{B}^\star(\tilde{w}_\infty)$ is homogeneous (e.g. controllable), is based on partitioning the Hankel matrix into the 'past' and the 'future', as follows

$$\mathscr{H}(\tilde{w}_\infty)_{[1:\mathtt{d}+\mathtt{d}']} = \begin{bmatrix} \mathscr{H}(\tilde{w}_\infty)_{[1:\mathtt{d}]} \\ ------- \\ \mathscr{H}(\tilde{w}_\infty)_{[\mathtt{d}+1:\mathtt{d}+\mathtt{d}']} \end{bmatrix} \begin{matrix} \leftarrow \text{`past'} \quad \mathscr{P} \\ \leftarrow \text{`present'} \\ \leftarrow \text{`future'} \quad \mathscr{F} \end{matrix}$$

where

$$\mathscr{H}(\tilde{w}_\infty)_{[\mathtt{k}':\mathtt{k}'']} := \begin{bmatrix} \tilde{w}_\infty(\mathtt{k}') & \tilde{w}_\infty(\mathtt{k}'+1) & \cdots & \tilde{w}_\infty(\mathtt{k}'+\mathtt{t}) & \cdots \\ \tilde{w}_\infty(\mathtt{k}'+1) & \tilde{w}_\infty(\mathtt{k}'+2) & \cdots & \tilde{w}_\infty(\mathtt{k}'+\mathtt{t}+1) & \cdots \\ \vdots & \vdots & \vdots\vdots & \vdots & \vdots\vdots \\ \tilde{w}_\infty(\mathtt{k}'') & \tilde{w}_\infty(\mathtt{k}''+1) & \cdots & \tilde{w}_\infty(\mathtt{k}''+\mathtt{t}-1) & \cdots \end{bmatrix}.$$

Consider the span of the rows of the past $\mathscr{P}$ and of the future $\mathscr{F}$. It turns out that for $\mathtt{d},\mathtt{d}'$ sufficiently large, $\mathtt{d},\mathtt{d}' \geq \mathtt{L}(\mathscr{B}^\star(\tilde{w}_\infty)) - 1$, the intersection of these spans is equal to the state space of $\mathscr{B}^\star(\tilde{w}_\infty)$, more precisely, the linear combinations of the rows of $\mathscr{P}$ (or $\mathscr{F}$) contained in this intersection give the 'present' state trajectory! These linear combinations thus yield the state trajectory

$$\tilde{x}_\infty = (\tilde{x}(\mathtt{d}), \tilde{x}(\mathtt{d}+1), \ldots, \tilde{x}(\mathtt{t}), \cdots).$$

This fact, noticed in (Willems, 1986, 1987, sections 15–17), and generalized in many directions in (Van Overschee and De Moor, 1994; Van Overschee and De Moor, 1996), allows to identify the system parameters (A,B,C,D) of $\mathscr{B}^\star(\tilde{w}_\infty)$ by solving $(\mathscr{L})$.

How do we compute the common linear combinations spanned by the rows of M_1 and M_2 in the partitioned matrix $\begin{bmatrix} M_1 \\ M_2 \end{bmatrix}$? This can be done by computing a basis for the left kernel of $\begin{bmatrix} M_1 \\ M_2 \end{bmatrix}$. Indeed, if $\begin{bmatrix} n_1 & n_2 \end{bmatrix}$ is in the left kernel, then $n_1 M_1 = -n_2 M_2$ yields one of these linear combinations. This can be applied to $\left[\begin{array}{c} \mathscr{P} \\ \hline \mathscr{F} \end{array}\right]$ and allows to compute the desired state trajectory, leading to the following conceptual algorithm for SYSID:

Data: $\tilde{w}_\infty : \mathbb{N} \to \mathbb{R}^{\mathtt{w}}$, (known) i/o partitioning $\begin{bmatrix} \tilde{u}_\infty \\ \tilde{y}_\infty \end{bmatrix}$. Assume that $\mathscr{B}^\star(\tilde{w}_\infty)$ is homogeneous.

1: Obtain (an estimate of) an upper bound of $\mathtt{L}(\mathscr{B}^\star(\tilde{w}_\infty))$. Choose $\mathtt{d},\mathtt{d}' \geq \mathtt{L}(\mathscr{B}^\star(\tilde{w}_\infty))$-1.

2: Construct the Hankel matrices $\mathscr{H}(\tilde{w}_\infty)_{[1:\mathtt{d}+\mathtt{d}']}$, $\mathscr{H}(\tilde{w}_\infty)_{[1:\mathtt{d}]}$ and, $\mathscr{H}(\tilde{w}_\infty)_{[\mathtt{d}+1:\mathtt{d}+\mathtt{d}']}$. Compute the left kernel N of $\mathscr{H}(\tilde{w}_\infty)_{[1:\mathtt{d}+\mathtt{d}']}$ and partition it conformably as $N = \begin{bmatrix} N_{\mathscr{P}} & N_{\mathscr{F}} \end{bmatrix}$

3: Compute $\mathscr{X} = N_{\mathscr{P}} \; \mathscr{H}(\tilde{w}_\infty)_{[1:\mathtt{d}]}$.

4: Row reduce $\mathscr{X} \rightsquigarrow \mathscr{X}'$ by deleting the (approximately) dependent rows. Columnwise:

$$\mathscr{X}' = \begin{bmatrix} \tilde{x}(\mathtt{d}) & \tilde{x}(\mathtt{d}+1) & \cdots & \tilde{x}(\mathtt{t}) & \cdots \end{bmatrix}.$$

Solve (by LS) the following equations for $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$:

$$\begin{bmatrix} \tilde{x}(\mathtt{d}+1) & \tilde{x}(\mathtt{d}+2) & \cdots & \tilde{x}(\mathtt{t}+1) & \cdots \\ \tilde{y}(\mathtt{d}) & \tilde{y}(\mathtt{d}+1) & \cdots & \tilde{y}(\mathtt{t}) & \cdots \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \tilde{x}(\mathtt{d}) & \tilde{x}(\mathtt{d}+1) & \cdots & \tilde{x}(\mathtt{t}) & \cdots \\ \tilde{u}(\mathtt{d}) & \tilde{u}(\mathtt{d}+1) & \cdots & \tilde{u}(\mathtt{t}) & \cdots \end{bmatrix}.$$

Output: (A,B,C,D), (an estimate of) $\mathscr{B}^\star(\tilde{w}_\infty)$ in i/s/o form.

This algorithm does not make use of the Hankel structure of the matrices involved, a common drawback of subspace algorithms. Equivalently, it does not make use of the module structure of the left kernel. For $\mathtt{d},\mathtt{d}'$ large (say, much larger than $\mathtt{L}(\mathscr{B}^\star(\tilde{w}_\infty))$), this algorithm risks at being inefficient.

4.2 State construction without past/future partition

But, there is more. As we have seen in the previous section, the left kernel of $\mathscr{H}(\tilde{w}_\infty) = \left[\begin{array}{c} \mathscr{P} \\ \hline \mathscr{F} \end{array}\right]$ can, for $d \geq \mathtt{L}(\mathscr{B}^\star(\tilde{w}_\infty))$, be deduced from the left kernel of $\mathscr{P}$ by simple module operations, and therefore there is no need to consider the separation into 'past' and 'future'. We state the resulting algorithm as a variation of the previous one:

Data: $\tilde{w}_\infty : \mathbb{N} \to \mathbb{R}^{\mathtt{w}}$, (known) i/o partitioning $\begin{bmatrix} \tilde{u}_\infty \\ \tilde{y}_\infty \end{bmatrix}$.

1: Obtain (an estimate of) an upper bound of $\mathtt{L}(\mathscr{B}^\star(\tilde{w}_\infty))$. Choose $\mathtt{d} \geq \mathtt{L}(\mathscr{B}^\star(\tilde{w}_\infty))$.

2: Choose $\mathtt{d} \geq \mathtt{L}(\mathscr{B}^\star(\tilde{w}_\infty))$. Construct the Hankel matrix $\mathscr{H}(\tilde{w}_\infty)_{[1:\mathtt{d}]}$. Compute the left kernel N of $\mathscr{H}(\tilde{w}_\infty)_{[1:\mathtt{d}]}$ and partition it conformably as

$$N = \begin{bmatrix} N_1 & N_2 & \cdots & N_{\mathtt{d}-1} & N_{\mathtt{d}} \end{bmatrix}.$$

3: Construct the following extension of N,

$$N_{\text{ext}} = \begin{bmatrix} N_2 & N_3 & \cdots & N_{\mathtt{d}-1} & N_{\mathtt{d}} \\ N_3 & N_4 & \cdots & N_{\mathtt{d}} & 0 \\ \vdots & \vdots & \vdots\vdots & \vdots & \vdots \\ N_{\mathtt{d}-1} & N_{\mathtt{d}} & \cdots & 0 & 0 \\ N_{\mathtt{d}} & 0 & \cdots & 0 & 0 \end{bmatrix}$$

Compute $\mathscr{X} = N_{\text{ext}} \; \mathscr{H}(\tilde{w}_\infty)_{[1:\mathtt{d}-1]}$.

4: Row reduce $\mathscr{X} \rightsquigarrow \mathscr{X}'$ by deleting the (approximately) dependent rows. Columnwise:

$$\mathscr{X}' = \begin{bmatrix} \tilde{x}(1) & \tilde{x}(2) & \cdots & \tilde{x}(\mathtt{t}) & \cdots \end{bmatrix}.$$

Solve (by LS) the following equations for $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$:

$$\begin{bmatrix} \tilde{x}(2) & \tilde{x}(3) & \cdots & \tilde{x}(\mathtt{t}+1) & \cdots \\ \tilde{y}(1) & \tilde{y}(2) & \cdots & \tilde{y}(\mathtt{t}) & \cdots \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \tilde{x}(1) & \tilde{x}(2) & \cdots & \tilde{x}(\mathtt{t}) & \cdots \\ \tilde{u}(1) & \tilde{u}(2) & \cdots & \tilde{u}(\mathtt{t}) & \cdots \end{bmatrix}.$$

Output: (A,B,C,D), (an estimate of) $\mathscr{B}^\star(\tilde{w}_\infty)$ in i/s/o form.

The advantage of this algorithm compared to those based on past/future partition is that it requires computing the left kernel of a matrix of half the size. The use of a set of module generators further reduces the computational burden.

The past/future partition idea is used in the N4SID algorithm (Van Overschee and De Moor, 1994; Van Overschee and De Moor, 1996) based on oblique projection, and in various other algorithms for (approximate) deterministic SYSID. It is a question whether the above ideas (which exploit the Hankel structure) render these algorithms obsolete. However, since these algorithms also apply to the stochastic case, further research is required to warrant this conclusion.

5. THE MPUM FOR FINITE TIME-SERIES

In this section, we extend the notion of an MPUM to the case that we observe a finite time-series

$$\tilde{w}_{\mathtt{T}} = (\tilde{w}(1), \tilde{w}(2), \ldots, \tilde{w}(\mathtt{T})) \qquad \tilde{w}(\mathtt{t}) \in \mathbb{R}^{\mathtt{w}}.$$

The behavior $\mathscr{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ is *unfalsified* by $\tilde{w}_{\mathtt{T}}$ if $\tilde{w}_{\mathtt{T}} \in \mathscr{B}|_{[1,\mathtt{T}]}$. $\mathscr{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ is *more powerful* than $\mathscr{B}' \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ if $\mathscr{B} \subset \mathscr{B}'$. $\mathscr{B} \in \mathscr{L}^{\mathtt{w}}$ is the *MPUM in* $\mathscr{L}^{\mathtt{w}}$ *for* $\tilde{w}_{\mathtt{T}}$ if it is unfalsified by $\tilde{w}_{\mathtt{T}}$ and more powerful than any other behavior in $\mathscr{L}^{\mathtt{w}}$ that is also unfalsified by $\tilde{w}_{\mathtt{T}}$. For the finite time-series case, the MPUM in $\mathscr{L}^{\mathtt{w}}$ may not exist, but it is not particularly useful anyway: when it exists, its behavior will be finite dimensional (corresponding to an autonomous behavior - a behavior with only outputs). However, it is desirable to have an MPUM that can also recover a behavior with some of the variables inputs. This can be accomplished by looking for the MPUM in the class of systems with a restricted lag, which can be viewed as limiting the complexity of the model.

To this end, define $\mathscr{L}^{\mathtt{w}}_{\mathtt{L}} := \{\mathscr{B} \in \mathscr{L}^{\mathtt{w}} \mid \mathsf{L}(\mathscr{B}) \leq \mathtt{L}\}$. $\mathscr{B} \in \mathscr{L}^{\mathtt{w}}$ is the *MPUM in* $\mathscr{L}^{w}_{L}$ *for* $\tilde{w}_{\mathtt{T}}$ if it is unfalsified by $\tilde{w}_{\mathtt{T}}$ and more powerful than any other behavior in $\mathscr{L}^{\mathtt{w}}_{\mathtt{L}}$ that is also unfalsified by $\tilde{w}_{\mathtt{T}}$. Denote the MPUM in $\mathscr{L}^{\mathtt{w}}_{\mathtt{L}}$ for $\tilde{w}_{\mathtt{T}}$ by $\mathscr{B}^{\star}_{\mathtt{L}}(\tilde{w}_{\mathtt{T}})$). The following questions arise: *When does* $\mathscr{B}^{\star}_{L}(\tilde{w}_{\mathtt{T}})$ *exist? How can it be computed from* $\tilde{w}_{\mathtt{T}}$*? If* $\tilde{w}_{\mathtt{T}}$ *is generated by* $\mathscr{B} \in \mathscr{L}^{\mathtt{w}}$*, when will* $\mathscr{B}^{\star}_{L}(\tilde{w}_{\mathtt{T}}) = \mathscr{B}$*? How does subspace ID (meaning first constructing the state trajectory and then the MPUM) generalize to this case?*

Conditions for existence of this MPUM are readily deduced from the Hankel matrix formed by the data. We first introduce our notation for finite Hankel matrices. Let $f : [1, \mathtt{t}] \to \mathbb{R}^{\mathtt{w}}$. Define the *Hankel matrix formed by* f with *depth* Δ_1 and *width* Δ_2, $\Delta_1 + \Delta_2 - 1 \leq \mathtt{t}$, by

$$\mathscr{H}_{(\Delta_1,\Delta_2)}(f) := \begin{bmatrix} f(1) & f(2) & \cdots & f(\Delta_2) \\ f(2) & f(3) & \cdots & f(\Delta_2+1) \\ \vdots & \vdots & \ddots & \vdots \\ f(\Delta_1) & f(\Delta_1+1) & \cdots & f(\Delta_1+\Delta_2-1) \end{bmatrix}.$$

View the elements in the left kernel again as vector polynomials. Assume that $n = \begin{bmatrix} n_0 & n_1 & \cdots & n_{\Delta_1-1} \end{bmatrix} \cong n_0 + n_1\xi + \cdots + n_{\Delta_1-1}\xi^{\Delta_1-1} \in \mathsf{leftker}\big(\mathscr{H}_{(\Delta_1,\Delta_2)}(f)\big)$. It may *appear* as if $\mathrm{degree}(n) = \Delta_1 - 1$, but the actual degree may be lower. Call $\mathsf{leftker}\big(\mathscr{H}_{(\Delta_1,\Delta_2)}(f)\big)$ *module-like* if

$$\begin{bmatrix} n_0 & n_1 & \cdots & n_{\Delta_1-2} & 0 \end{bmatrix} \in \mathsf{leftker}\big(\mathscr{H}_{(\Delta_1,\Delta_2)}(f)\big)$$
$$\Rightarrow \begin{bmatrix} 0 & n_0 & n_1 & \cdots & n_{\Delta_1-2} \end{bmatrix} \in \mathsf{leftker}\big(\mathscr{H}_{(\Delta_1,\Delta_2)}(f)\big).$$

Equivalently, if

$$\big[n(\xi) \in \mathsf{leftker}\big(\mathscr{H}_{(\Delta_1,\Delta_2)}(f)\big) \wedge \mathrm{degree}(n) < \Delta_1 - 1\big]$$
$$\Rightarrow \big[\xi n(\xi) \in \mathsf{leftker}\big(\mathscr{H}_{(\Delta_1,\Delta_2)}(f)\big)\big].$$

This shows where 'module-like' refers to.

These notions are very useful for computing $\mathscr{B}^{\star}_{\mathtt{L}}(\tilde{w}_{\mathtt{T}})$ (which may not exist – take $\tilde{w}_{\mathtt{T}} = (1,1,2,2,2)$ and $\mathtt{L} = 2$). If $\mathscr{B} \in \mathscr{L}^{\mathtt{w}}$ is unfalsified by $\tilde{w}_{\mathtt{T}}$, then

$$\big[n \in \mathscr{N}_{\mathscr{B}} \text{ and } \mathrm{degree}(n) \leq \mathtt{L}\big]$$
$$\Rightarrow \big[n \in \mathsf{leftker}\big(\mathscr{H}_{(\mathtt{L}+1,\mathtt{T}-\mathtt{L})}(\tilde{w}_{\mathtt{T}})\big)\big].$$

It is logical to aim at the module generated by $\mathsf{leftker}\big(\mathscr{H}_{(\mathtt{L}+1,\mathtt{T}-\mathtt{L})}(\tilde{w}_{\mathtt{T}})\big)$ as the annihilators of the MPUM $\mathscr{B}^{\star}_{\mathtt{L}}(\tilde{w}_{\mathtt{T}})$. However, for arbitrary $\tilde{w}_{\mathtt{T}}$, this module may contain falsified elements (consider, e.g. $\tilde{w}_{\mathtt{T}} = (0,0,\ldots,0,0,1)$). The module-like property ensures that all elements in the module generated by the$\mathsf{leftker}\big(\mathscr{H}_{(\mathtt{L}+1,\mathtt{T}-\mathtt{L})}(\tilde{w}_{\mathtt{T}})\big)$ are unfalsified. This leads to the following results.

(i) A sufficient condition for $\mathscr{B}^{\star}_{\mathtt{L}}(\tilde{w}_{\mathtt{T}})$ to exist is that $\mathsf{leftker}\big(\mathscr{H}_{(\mathtt{L}+1,\mathtt{T}-\mathtt{L})}(\tilde{w}_{\mathtt{T}})\big)$ is module-like.

(ii) $\mathsf{leftker}\big(\mathscr{H}_{(\mathtt{L}+1,\mathtt{T}-\mathtt{L})}(\tilde{w}_{\mathtt{T}})\big)$ is module-like iff

$$\mathrm{rank}\big(\mathscr{H}_{(\mathtt{L},\mathtt{T}-\mathtt{L})}(\tilde{w}_{\mathtt{T}})\big) = \mathrm{rank}\big(\mathscr{H}_{(\mathtt{L},\mathtt{T}-\mathtt{L}+1)}(\tilde{w}_{\mathtt{T}})\big).$$

(iii) If $\mathsf{leftker}\big(\mathscr{H}_{(\mathtt{L}+1,\mathtt{T}-\mathtt{L})}(\tilde{w}_{\mathtt{T}})\big)$ is module-like then $\mathscr{B}^{\star}_{\mathtt{L}}(\tilde{w}_{\mathtt{T}}) = \ker(N(\sigma))$, where $N \in \mathbb{R}^{\bullet\times\mathtt{w}}[\xi]$ is any polynomial matrix such that the $\mathbb{R}[\xi]$-module generated by its rows equals the $\mathbb{R}[\xi]$-module generated by $\mathsf{leftker}\big(\mathscr{H}_{(\mathtt{L}+1,\mathtt{T}-\mathtt{L})}(\tilde{w}_{\mathtt{T}})\big)$.

Under mild conditions (Willems *et al.*, 2005), $\mathscr{B}^{\star}_{\mathtt{L}}(\tilde{w}_{\mathtt{T}})$ recovers the system that generated the data: the procedure is consistent. The crucial condition here is that the input component in the observations must be persistently exciting.

6. SUBSPACE ID FOR FINITE-TIME SERIES

We now apply these ideas to obtain an algorithm to compute $\mathscr{B}^{\star}_{\mathtt{L}}(\tilde{w}_{\mathtt{T}})$ by first constructing the state, and subsequently an i/s/o model for $\mathscr{B}^{\star}_{\mathtt{L}}(\tilde{w}_{\mathtt{T}})$. We state the result in the from of a procedure, similar to the one obtained for the infinite time series in section 4.2.

Data: $\tilde{w}_{\mathtt{T}} : [1, T] \to \mathbb{R}^{\mathtt{w}}$ and the lag $\mathtt{L} \in \mathbb{Z}_+$. We are looking for the MPUM in $\mathscr{L}^{\mathtt{w}}_{\mathtt{L}}$ for $\tilde{w}_{\mathtt{T}}$. Assume that $\mathsf{leftker}\big(\mathscr{H}_{(\mathtt{L}+1,\mathtt{T}-\mathtt{L})}(\tilde{w}_{\mathtt{T}})\big)$ is module-like, and that the i/o partitioning $w = \left[\begin{smallmatrix} u \\ y \end{smallmatrix}\right]$ of $\mathscr{B}^{\star}_{\mathtt{L}}(\tilde{w}_{\mathtt{T}})$ is known.

1: Compute the left kernel N of $\mathscr{H}_{(\mathtt{L}+1,\mathtt{T}-\mathtt{L})}(\tilde{w}_{\mathtt{T}})$ and partition it conformably as

$$N = \begin{bmatrix} N_0 & N_1 & N_2 & \cdots & N_{\mathtt{L}-2} & N_{\mathtt{L}-1} & N_{\mathtt{L}} \end{bmatrix}.$$

2: Construct the follwoing extension of N,

$$N_{\mathrm{ext}} = \begin{bmatrix} N_1 & N_2 & N_3 & \cdots & N_{\mathtt{L}-1} & N_{\mathtt{L}} \\ N_2 & N_3 & N_4 & \cdots & N_{\mathtt{L}} & 0 \\ N_3 & N_4 & N_5 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots\vdots & \vdots & \vdots \\ N_{\mathtt{L}-2} & N_{\mathtt{L}-1} & N_{\mathtt{L}} & \cdots & 0 & 0 \\ N_{\mathtt{L}-1} & N_{\mathtt{L}} & 0 & \cdots & 0 & 0 \\ N_{\mathtt{L}} & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

and compute $\mathscr{X} = N_{\mathrm{ext}}\,\mathscr{H}_{(\mathtt{L},\mathtt{T}-\mathtt{L}+1)}(\tilde{w}_{\mathtt{T}})$.

3: Row reduce $\mathscr{X} \rightsquigarrow \mathscr{X}'$ by deleting the (approximately) dependent rows. Columnwise:

$$\mathscr{X}' = \begin{bmatrix} \tilde{x}(1) & \tilde{x}(2) & \cdots & \tilde{x}(\mathtt{T}-\mathtt{L}+1) \end{bmatrix}.$$

Solve (by LS) the following equations for $\left[\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right]$:

$$\begin{bmatrix} \tilde{x}(2) & \tilde{x}(3) & \cdots & \tilde{x}(\mathtt{T}-\mathtt{L}+1) \\ \tilde{y}(1) & \tilde{y}(2) & \cdots & \tilde{y}(\mathtt{T}-\mathtt{L}) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \tilde{x}(1) & \tilde{x}(2) & \cdots & \tilde{x}(\mathtt{T}-\mathtt{L}) \\ \tilde{u}(1) & \tilde{u}(2) & \cdots & \tilde{u}(\mathtt{T}-\mathtt{L}) \end{bmatrix}.$$

Output: (A,B,C,D), (an estimate of) $\mathscr{B}^{\star}_{\mathtt{L}}(\tilde{w}_{\mathtt{T}})$ in i/s/o form.

7. CONCLUSIONS

In this paper, we discussed exact SYSID. The central notion is that of the MPUM, the model in the model class that explains the data and as little else as possible. For linear time-invariant complete systems and an infinite time-series as data, the MPUM always exists. In the case of a finite time series, it exists, under suitable assumptions, in the model class of linear time-invariant complete systems with a restricted lag.

In subspace ID, an input/state/output representation of the MPUM is obtained by first deducing, directly from the observed time-series, the corresponding state trajectory, and subsequently solving for the system parameter matrices (A,B,C,D). This state construction is usually obtained from a past/future partitioning of the Hankel matrix of the data. In the present paper, we have shown that, by exploiting the Hankel structure, there is no need to make this partitioning and that the state trajectory may be obtained from the past (or the future). It is interest to examine the implications of these algorithms in the combined deterministic/stochastic case.

ACKNOWLEDGEMENTS

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 AMBioRICS, and by several grants the Flemish Fund for Scientific Research: ICCoS, ANMMM, MLDM.

REFERENCES

Markovsky, I., J. C. Willems and B. De Moor (2005). Recursive computation of the most powerful unfalsified model. Technical Report 05–169. Dept. EE, K.U.Leuven.

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

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 (1996). *Subspace Identification for Linear Systems: Theory, Implementation, Applications*. Kluwer.

Willems, J. C. (1986, 1987). From time series to linear system—Part I. Finite dimensional linear time invariant systems, Part II. Exact modelling, Part III. Approximate modelling. *Automatica* **22, 23**, 561–580, 675–694, 87–115.

Willems, J. C., P. Rapisarda, I. Markovsky and B. De Moor (2005). A note on persistency of excitation. *Systems & Control Letters* **54**(4), 325–329.