Recursive Computation of the MPUM

Jan C. Willems

ESAT-SISTA, K.U. Leuven, B-3001 Leuven, Belgium
`Jan.Willems@esat.kuleuven.be`
`www.esat.kuleuven.be/∼jwillems`

Summary. An algorithm is presented for the computation of the most powerful unfalsified model associated with an observed vector time series in the class of dynamical systems described by linear constant coefficient difference equations. This algorithm computes a module basis of the left kernel the Hankel matrix of the data, and is recursive in the elements of the basis. It is readily combined with subspace identification ideas, in which a state trajectory is computed first, directly from the data, and the parameters of the identified model are derived from the state trajectory.

1 Introduction

It is a true pleasure to contribute an article to this Festschrift in honor of Giorgio Picci on the occasion of his 65-th birthday. In the 35 years since our original acquaintance, I have learned to appreciate Giorgio as a deep thinker and a kind friend. As the topic of this paper, I chose a subject that has dominated Giorgio's research throughout his scientific career: *system identification*.

My paper is purely deterministic in nature, whereas the usual approach to system identification (SYSID) is stochastic. It has always baffled me that so many subjects in systems and control — and in other scientific endeavors as well — immediately pass to a stochastic setting. Motivated by the thought that in the end uncertainty will have to be dealt with, a stochastic framework is adopted *ab initio*, and the question of how the problem would look in a deterministic setting is not even addressed. Moreover, it is considered evident that uncertainty leads to stochasticity. My belief is that from a methodological point of view, it is more reasonable to travel from exact deterministic, to approximate deterministic, to stochastic, and end with approximate stochastic SYSID. See [14] for a more elaborate explanation of my misgivings for using stochastics as basis for SYSID.

This brings up the question what we should mean by 'the' exact deterministic model identified by an observed vector time series. The concept that fits this aim is the *most powerful unfalsified model* (MPUM), the model in the model class that explains the observations, but as little else as possible. The purpose of this article is to develop an algorithm to pass from an observed vector time series to the MPUM in the familiar model class of systems described by linear constant coefficient difference equations.

We start the development with the well-known state construction based on the intersection of the row spaces of a past/future partition of the Hankel matrix of the data. By scrutinizing this algorithm, using the Hankel structure, we deduce that this state construction can be done without the past/future partition, and requires only the computation of the left kernel of the Hankel matrix itself. However, this left kernel is infinite dimensional, but it has the structure of a finitely generated module, and the state construction can be done using a module basis. It therefore suffices to compute a finite number of elements of the left kernel.

This computation can be done recursively, as follows. Truncate the Hankel matrix at consecutive rows, until an element in the left kernel of the Hankel matrix is found. Next, consider the system defined by this element, and construct a direct complement of it. Subsequently, compute the error defined by a kernel representation of this complement, applied to the original time series. This error has lower dimension than the original one. This is recursively repeated until the error is persistently exciting. This leads to an algorithm that computes the MPUM. It readily also gives the state trajectory corresponding to the observed trajectory. The algorithm is therefore very adapted to be used in concordance with subspace identification, in which also a state model of the MPUM is computed, with all the advantages thereof, for example for model reduction.

We present only the ideas underlying this algorithm. Proofs and simulations will appear elsewhere.

A couple of words about the notation used. $\mathbb{N}$ denotes the set of natural numbers, and $\mathbb{R}$ the reals. $\mathbb{R}[\xi]$ denotes, as usual, the ring of polynomials with real coefficients, and $\mathbb{R}(\xi)$ the field of real rational functions, with ξ the indeterminate. Occasionally, we use the notation $\mathbb{R}[\xi]^{\bullet \times \mathtt{w}}$ for polynomial matrices with $\mathtt{w}$ columns but an unspecified (finite) number of rows. The backwards shift is denote by σ, and defined, for $f : \mathbb{N} \to \mathbb{F}$, by

$$\sigma f : \mathbb{N} \to \mathbb{F}, \quad \sigma f(t) := f(t+1).$$

2 Problem statement

The problem discussed in this paper may be compactly formulated as follows.

> *Given an observed vector time series*
>
> $$\tilde{w} = (\tilde{w}(1), \tilde{w}(2), \ldots, \tilde{w}(t), \ldots)$$
>
> *with $\tilde{w}(t) \in \mathbb{R}^{\mathtt{w}}$ and $t \in \mathbb{N}$, find the most powerful unfalsified model associated with $\tilde{w}$ in the model class of dynamical systems consisting of the set of solutions of linear constant coefficient difference equations.*

In section 8 we discuss how the ideas can be adapted to deal with more realistic situations: finite time series, missing data (due to erasures or censoring), multiple time series, approximate modeling, etc. In the present section, the terminology used in the problem statement is explained.

We consider discrete time systems with time set $\mathbb{N}$ and with signal space a finite dimensional real vector space, say $\mathbb{R}^{\mathtt{w}}$. With a dynamical model, we mean a family of trajectories from $\mathbb{N}$ to $\mathbb{R}^{\mathtt{w}}$, a *behavior* $\mathscr{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$. $\mathscr{B}$ is

$$[\![\textit{unfalsified} \text{ by } \tilde{w}]\!] :\Leftrightarrow [\![\tilde{w} \in \mathscr{B}]\!].$$

Let $\mathscr{B}_1, \mathscr{B}_2 \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$. Call

$$[\![\mathscr{B}_1 \textit{ more powerful} \text{ than } \mathscr{B}_2]\!] :\Leftrightarrow [\![\mathscr{B}_1 \subset \mathscr{B}_2]\!].$$

Modeling is prohibition, and the more a model forbids, the better it is. A *model class* is a set of behaviors. The *most powerful unfalsified model* (MPUM) associated with $\tilde{w}$ in a model class is a behavior that is unfalsified by $\tilde{w}$ and more powerful than any other unfalsified model in this model class. In other words, the MPUM explains the data $\tilde{w}$, but as little else as possible. Obviously, for a general model class, the MPUM may not exist. We now describe a model class for which the MPUM does exist.

This model class is a very familiar one. It consists of the behaviors that are the set of solutions of linear constant coefficient difference equations. Explicitly, each behavior $\mathscr{B}$ in this model class is defined by a real polynomial matrix $R \in \mathbb{R}[\xi]^{\bullet \times \mathtt{w}}$ as

$$\mathscr{B} = \{ w : \mathbb{N} \to \mathbb{R}^{\mathtt{w}} \mid R(\sigma) w = 0 \}.$$

Since $\mathscr{B} = \text{kernel}(R(\sigma))$, with $R(\sigma)$ viewed as a map from $(\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ to $(\mathbb{R}^{\text{rowdim}(R)})^{\mathbb{N}}$, we call

$$R(\sigma) w = 0 \qquad (\mathscr{K})$$

a *kernel representation* of the corresponding behavior.

We denote this model class by $\mathscr{L}^{\mathtt{w}}$. The many ways of arriving at it, and various equivalent representations of its elements are described, for example, in [14, section 3]. Perhaps the simplest, 'representation free', way of characterizing $\mathscr{L}^{\mathtt{w}}$ is as follows. $\mathscr{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ belongs to $\mathscr{L}^{\mathtt{w}}$ if and only if it has the following three properties:

(i) $\mathscr{B}$ is *linear*,
(ii) *shift-invariant* ($\sigma \mathscr{B} \subseteq \mathscr{B}$), and
(iii) *closed* in the topology of point-wise convergence.

Consider $\mathbb{R}[\xi]^{\mathtt{w}}$. Obviously it is a module over $\mathbb{R}[\xi]$. Let $\mathscr{M}^{\mathtt{w}}$ denote the set of $\mathbb{R}[\xi]$-submodules of $\mathbb{R}[\xi]^{\mathtt{w}}$. It is well known that each element of $\mathscr{M}^{\mathtt{w}}$ is finitely generated, meaning that for each $\mathbb{M} \in \mathscr{M}^{\mathtt{w}}$, there exist $g_1, g_2, \ldots, g_{\mathtt{p}}$ such that

$$\mathbb{M} = \{ r \in \mathbb{R}[\xi]^{\mathtt{w}} \mid \exists\, \alpha_1, \alpha_2, \ldots, \alpha_{\mathtt{p}} \in \mathbb{R}[\xi] \text{ such that } r = \alpha_1 g_1 + \alpha_2 g_2 + \cdots + \alpha_{\mathtt{p}} g_{\mathtt{p}} \}.$$

There exists a $1 \leftrightarrow 1$ relation between $\mathscr{L}^{\mathtt{w}}$ and $\mathscr{M}^{\mathtt{w}}$. This may be seen as follows. Call

$$[\![r \in \mathbb{R}[\xi]^{\mathtt{w}} \text{ an } \textit{annihilator} \text{ for } \mathscr{B}]\!] :\Leftrightarrow [\![r^\top(\sigma)\mathscr{B} = 0]\!].$$

Denote the set of annihilators of $\mathscr{B}$ by $\mathscr{B}^\perp$. Clearly $\mathscr{B}^\perp \in \mathscr{M}^{\mathtt{w}}$. This identifies the map $\mathscr{B} \in \mathscr{L}^{\mathtt{w}} \mapsto \mathscr{B}^\perp \in \mathscr{M}^{\mathtt{w}}$. It can be shown that this map is surjective (not totally trivial, but true). When $\mathscr{B}$ has kernel representation ($\mathscr{K}$), then $\mathscr{B}^\perp$ is the $\mathbb{R}[\xi]$-module generated by the transposes of the rows of R. To travel the reverse route and associate an element $\mathscr{B} \in \mathscr{L}^{\mathtt{w}}$ with a module $\mathbb{M} \in \mathscr{M}^{\mathtt{w}}$, take the behavior of the system with kernel representation ($\mathscr{K}$) generated by the polynomial matrix R with as rows the transposes of a set of generators of $\mathbb{M}$.

The module of annihilators is a more appropriate way of thinking about an element $\mathscr{B} \in \mathscr{L}^{\mathtt{w}}$ than the specific, but less intrinsic, difference equation ($\mathscr{K}$) which one happens to have chosen to define $\mathscr{B}$.

The special case of ($\mathscr{K}$) given by the overly familiar

$$P(\sigma)y = Q(\sigma)u, \quad w = (u,y) \qquad \text{(i/o)}$$

with $P \in \mathbb{R}[\xi]^{\mathtt{p}\times\mathtt{p}}, Q \in \mathbb{R}[\xi]^{\mathtt{p}\times\mathtt{m}}, \det(P) \neq 0,$ and with proper transfer function $G = P^{-1}Q \in \mathbb{R}(\xi)^{\mathtt{p}\times\mathtt{m}}$, is called an *input/output* (i/o) system, with $u : \mathbb{N} \to \mathbb{R}^{\mathtt{m}}$ the *input* and $y : \mathbb{N} \to \mathbb{R}^{\mathtt{p}}$ the *output*. The conditions imposed on P, Q ensure that u is free, and that y does not anticipate u, the usual requirements on an input/output system. Clearly (i/o) defines an element of $\mathscr{L}^{\mathtt{m}+\mathtt{p}}$. Conversely, for every $\mathscr{B} \in \mathscr{L}^{\mathtt{w}}$, there exists a system (i/o) with $\mathtt{m} + \mathtt{p} = \mathtt{w}$, that has, up to a mere reordering of the components, behavior $\mathscr{B}$. With this reordering, we mean that there exists a permutation matrix $\Pi \in \mathbb{R}^{\mathtt{w}\times\mathtt{w}}$ (depending on $\mathscr{B}$, of course), such that (i/o) has behavior $\Pi\mathscr{B}$. In the sequel, we often silently assume that the permutation that makes the inputs the leading, and the outputs the trailing components of w has been carried out already.

As in all of system theory, controllability plays an important role also in the theory surrounding the MPUM. We recall the behavioral definition of controllability.

$$\begin{aligned} &[\![\mathscr{B} \in \mathscr{L}^{\mathtt{w}} \text{ is } \textit{controllable}]\!] \\ &\quad :\Leftrightarrow [\![\forall\, w_1, w_2 \in \mathscr{B} \text{ and } t_1 \in \mathbb{N}, \exists\, w \in \mathscr{B} \text{ and } t_2 \in \mathbb{N}, t_2 \geq t_1, \text{ such that} \\ &\qquad w(t) = w_1(t) \text{ for } 1 \leq t \leq t_1, \text{ and } w(t) = w_2(t - t_1 - t_2) \text{ for } t > t_1 + t_2]\!]. \end{aligned}$$

Various characterizations of controllability may be found, for example, in [14, section 5].

It is easy to see that there exists an MPUM in $\mathscr{L}^{\mathtt{w}}$ associated with $\tilde{w}$. Denote it by $\tilde{\mathscr{B}}$. The most convincing proof that this MPUM exists, is by showing what it is:

$$\tilde{\mathscr{B}} = \overline{\text{linear span } (\{\tilde{w}, \sigma\tilde{w}, \ldots, \sigma^t\tilde{w}, \ldots\})},$$

where the right hand side means the closure in the topology of point-wise convergence of the linear span of the elements in the set. Obviously, this linear span is linear, it is shift invariant since it is constructed from $\tilde{w}$ and its shifts, and after taking the closure, it is closed in the topology of point-wise convergence. Consequently it belongs to $\mathscr{L}^{\mathtt{w}}$. It is also unfalsified, since $\tilde{w} \in \tilde{\mathscr{B}}$, and clearly any unfalsified element in $\mathscr{L}^{\mathtt{w}}$ must contain all the $\sigma^t\tilde{w}$'s and hence their linear span, and be closed in the topology of point-wise convergence. This proves that $\tilde{\mathscr{B}}$ is indeed the MPUM in $\mathscr{L}^{\mathtt{w}}$ associated with $\tilde{w}$.

($\mathscr{K}$) is unfalsified by $\tilde{w}$ iff $R(\sigma)\tilde{w} = 0$. It follows that among all polynomial matrices $R \in \mathbb{R}[\xi]^{\bullet \times \mathtt{w}}$ such that $R(\sigma)\tilde{w} = 0$, there is one whose behavior is more powerful than any other. And, of course, this MPUM allows an i/o representation. Our aim are algorithms to go from the observed time series $\tilde{w}$ to a representation of its associated MPUM in $\mathscr{L}^{\mathtt{w}}$. The most direct way to go about this is to compute, from $\tilde{w}$, a polynomial matrix R such that ($\mathscr{K}$) is a kernel representation of this MPUM. Equivalently, to compute a set of generators for $\mathscr{B}^{\perp}$. In [11] several such algorithms are described.

The 'consistency' problem consists of finding conditions so that the system that has generated the data is indeed the one that is identified by the system identification algorithm. In our deterministic setting this comes down to checking when the system that has generated $\tilde{w}$ is actually the MPUM in $\mathscr{L}^{\mathtt{w}}$ associated with $\tilde{w}$. Persistency of excitation, but also controllability, are the key conditions leading to consistency. The vector time series $f : \mathbb{N} \to \mathbb{R}^{\mathtt{k}}$ is

$$[\![\textit{persistently exciting}]\!] :\Leftrightarrow [\![\text{the MPUM in } \mathscr{L}^{\mathtt{k}} \text{ associated with } f \text{ equals } \left(\mathbb{R}^{\mathtt{k}}\right)^{\mathbb{N}}]\!].$$

Consider $\mathscr{B} \in \mathscr{L}^{\mathtt{w}}$, with i/o partition $w = (u, y)$. Assume that $\tilde{w} = (\tilde{u}, \tilde{y})$ is partitioned accordingly. Then $\mathscr{B}$ is the MPUM in $\mathscr{L}^{\mathtt{w}}$ associated with $\tilde{w}$ if

1. $\tilde{w} \in \mathscr{B}$,
2. $\tilde{u}$ is persistently exciting,
3. $\mathscr{B}$ is controllable.

In [13] a more general version of this consistency result is proven. Note that the first two conditions are clearly also necessary for consistency.

This result provides additional motivation for making the MPUM the aim of deterministic system identification.

3 Subspace identification

In addition to looking for a kernel representation of this MPUM, we are even more interested in obtaining a state space representation of it. We first explain what we mean by this.

Let $\mathtt{m}, \mathtt{p}, \mathtt{n}$ be nonnegative integers, $A \in \mathbb{R}^{\mathtt{n} \times \mathtt{n}}, B \in \mathbb{R}^{\mathtt{n} \times \mathtt{m}}, C \in \mathbb{R}^{\mathtt{p} \times \mathtt{n}}, D \in \mathbb{R}^{\mathtt{p} \times \mathtt{m}}$, and consider the ubiquitous system

$$\sigma x = Ax + Bu, \; y = Cx + Du, \; w = (u, y). \qquad (\mathscr{S})$$

In this equation, $u : \mathbb{N} \to \mathbb{R}^{\mathtt{m}}$ is the *input*, $y : \mathbb{N} \to \mathbb{R}^{\mathtt{p}}$ the *output*, and $x : \mathbb{N} \to \mathbb{R}^{\mathtt{n}}$ the *state* trajectory. The behavior

$$\{(u, y) : \mathbb{N} \to \mathbb{R}^{\mathtt{m}} \times \mathbb{R}^{\mathtt{p}} \mid \exists x : \mathbb{N} \to \mathbb{R}^{\mathtt{n}} \text{ such that } (\mathscr{S}) \text{ holds}\}$$

is called the *external behavior* of $(\mathscr{S})$. It can be shown that this external behavior belongs to $\mathscr{L}^{\mathtt{m}+\mathtt{p}}$. The (u,y,x)-behavior is obviously an element of $\mathscr{L}^{\mathtt{m}+\mathtt{p}+\mathtt{n}}$. This implies that the (u,y)-behavior, what we call the external behavior, is an element of $\mathscr{L}^{\mathtt{m}+\mathtt{p}}$. This is due to the fact than the projection onto a subset of the components of a linear shift-invariant closed subspace of $(\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ is again linear, shift-invariant, and closed. This result is called the ‘elimination theorem’, and is an important element in the behavioral theory of systems. It implies, for example, that $\mathscr{L}^{\mathtt{w}}$ is closed under addition.

So, the external behavior of $(\mathscr{S})$ belongs of to $\mathscr{L}^{\mathtt{m}+\mathtt{p}}$. Conversely, for every $\mathscr{B} \in \mathscr{L}^{\mathtt{w}}$, there exists a system $(\mathscr{S})$, with $\mathtt{m}+\mathtt{p}=\mathtt{w}$, that has, up to a mere reordering of the components (u,y), external behavior $\mathscr{B}$. With this reordering, we mean that there exists a permutation matrix $\Pi \in \mathbb{R}^{\mathtt{w}\times\mathtt{w}}$ such that $(\mathscr{S})$ has external behavior $\Pi\mathscr{B}$. In the sequel, we again often silently assume that the permutation that makes the inputs the leading, and the outputs the trailing components of w has been carried out already.

$(\mathscr{S})$ is called an *input/state/output* (i/s/o) representation of its external behavior. $(\mathscr{S})$ is called *minimal* if its state has minimal dimension among all i/s/o systems with the same external behavior. It can be shown that minimal is equivalent to state observable, meaning that if (u,y,x') and (u,y,x'') both satisfy $(\mathscr{S})$, then $x'=x''$. In other words, observability means that the state trajectory x can be deduced from the input and output trajectories (u,y) jointly. As is very well-known, observability holds iff the $(\mathtt{np}\times\mathtt{n})$ matrix $\mathrm{col}\left(C,CA,\cdots,CA^{\mathtt{n}-1}\right)$ has rank $\mathtt{n}$. Minimality does not imply controllability. But a minimal i/s/o representation is state controllable iff its external behavior is controllable, in the sense we have defined controllability of behaviors.

As explained in the previous section, we are looking for algorithms that pass from the observed time series $\tilde{w}$ to its MPUM in $\mathscr{L}^{\mathtt{w}}$, for example, by computing a kernel representation $(\mathscr{K})$ of this MPUM. There is, however, another way to go about this, by first computing the state trajectory corresponding to $\tilde{w}$ in the MPUM, and subsequently the system parameters (A,B,C,D) corresponding to an i/s/o representation. Explicitly, assume that we had somehow found the MPUM. We could then compute a minimal i/s/o representation for it, and obtain the (unique) state trajectory $\tilde{x}$ corresponding to $\tilde{w}$. Of course, for every $T \in \mathbb{N}$, there holds (assuming that the reordering of the components discussed before such that $\tilde{w}=(\tilde{u},\tilde{y})$ has been carried out)

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

So, if

$$\begin{bmatrix} \tilde{x}(1) & \tilde{x}(2) & \cdots & \tilde{x}(T) \\ \tilde{u}(1) & \tilde{u}(2) & \cdots & \tilde{u}(T) \end{bmatrix}$$

is of full row rank, (\$), truncated at column T, provides an equation for computing A,B,C,D, and yields an i/s/o representation of the MPUM.

As we explained it, this approach appears to be a vicious circle. For in order to compute $\tilde{x}$, we seem to need the MPUM to start with. But, if we could somehow compute $\tilde{x}$, directly from the data $\tilde{w}$, *without* deriving it from the MPUM, then ($) gives a viable and (see section 8) attractive way to compute an i/s/o representation of the MPUM. In section 8 we shall explain that even when we deduce $\tilde{x}$ from a kernel representation of the MPUM, it is advantageous to return to ($) for the purpose of system identification because of its built-in model reduction.

The SYSID methods that first compute the state trajectory from the data, and then derive the system model from the state trajectory have become known as *subspace identification* algorithms. Before the emergence of these methods, state space representations played a somewhat secondary role in system identification. The purpose of this paper is to take a closer look at (the deterministic version of) these algorithms.

4 State construction by past/future partition

The question is:

How can we compute the state trajectory $\tilde{x}$
directly from $\tilde{w}$, without first computing the MPUM $\tilde{\mathscr{B}}$?

The doubly infinite matrix

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

is called the *Hankel matrix* of the data $\tilde{w}$. It holds the key to the state construction.

The earliest subspace algorithms are based on the intersection of the span of the rows of a past/future partition of this Hankel matrix, and deduce the state trajectory as the common linear combinations of the past and the future. This proceeds as follows. Partition a row truncation of $\mathscr{H}$ as

$$\begin{bmatrix} \mathscr{H}_p \\ \hline \mathscr{H}_f \end{bmatrix} = \left[\begin{array}{ccccc} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \vdots & \vdots & \vdots\vdots\vdots & \vdots & \vdots\vdots\vdots \\ \tilde{w}(T_p) & \tilde{w}(T_p+1) & \cdots & \tilde{w}(T_p+t-1) & \cdots \\ \hline \tilde{w}(T_p+1) & \tilde{w}(T_p+2) & \cdots & \tilde{w}(T_p+t) & \cdots \\ \vdots & \vdots & \vdots\vdots\vdots & \vdots & \vdots\vdots\vdots \\ \tilde{w}(T_p+T_f-1) & \tilde{w}(T_p+T_f) & \cdots & \tilde{w}(T_p+T_f+t-2) & \cdots \\ \tilde{w}(T_p+T_f) & \tilde{w}(T_p+T_f+1) & \cdots & \tilde{w}(T_f+T_p+t-1) & \cdots \end{array}\right] \qquad (\mathscr{H}_p/\mathscr{H}_f)$$

and refer to $\mathscr{H}_p$ as the 'past', and to $\mathscr{H}_f$ as the 'future' of the Hankel matrix. T_p and T_f are sufficiently large nonnegative integers. Actually, it is possible to proceed after truncating these Hankel matrices also column-wise at a sufficiently large column T. We do not enter into details about what 'sufficiently large' exactly means in these statements — those issues are glossed over here.

Consider the intersection of the linear space spanned by the rows of $\mathscr{H}_p$ and the linear space spanned by the rows of $\mathscr{H}_f$. Let $\mathtt{n}$ be the dimension of this intersection. This means that there are $\mathtt{n}$ linearly independent linear combinations of the rows of $\mathscr{H}_p$ that are equal to linear combinations of the rows of $\mathscr{H}_f$. These linear combinations can be stacked into a matrix with $\mathtt{n}$ rows,

$$\tilde{X} = \left[\ \tilde{x}(T_p+1)\ \tilde{x}(T_p+2)\ \cdots\ \tilde{x}(T_p+t)\ \cdots\ \right]$$

It turns out that, under suitable conditions which are spelled out in [11], the dimension of this intersection equals the dimension of the state space of a minimal i/s/o representation of the MPUM in $\mathscr{L}^{\mathtt{w}}$ associated with $\tilde{w}$. Moreover, as the notation suggests, the columns of $\tilde{X}$ form the state trajectory corresponding to $\tilde{w}$ in the MPUM $\tilde{\mathscr{B}}$ (more precisely, corresponding to a minimal i/s/o representation of $\tilde{\mathscr{B}}$). This then leads, by equation ($), to an algorithm to compute the matrices A, B, C, D and to an identification procedure for the MPUM.

In [11] this intersection algorithm is applied to a variety of situations, including classical realization theory. These algorithms have been given good numerical linear algebra based implementations in [8].

In the *purely deterministic* case the state trajectory can be obtained, as we have just seen, as the *intersection* of the linear span of the rows of the past with the linear span of the rows of the future of the Hankel matrix of the data. This is, in a sense, analogous to the fact that in the *purely stochastic* case the state trajectory can be obtained as the *orthogonal projection* of the linear span of the rows of the past onto the linear span of the rows of the future of the Hankel matrix of the data, as noted in [1]. This idea was used in [6] for the purposes of stochastic SYSID. The resulting subspace methods in the context of the purely stochastic case have been followed up by many authors, see, in particular, [5] and [4]. The combined deterministic/stochastic case is a significant generalization of the purely deterministic case and the purely stochastic case individually. It has been studied in [7, 8]. Similar algorithms have been developed in [10]. In the mean time, many articles dealing with subspace algorithms for the combined case have appeared, for instance [2, 3].

5 The Hankel structure and the past/future partition

Let us now take a closer look at the intersection of the spaces spanned by the rows of $\mathscr{H}_p$ and by the rows of $\mathscr{H}_f$. *How can we obtain this intersection?* Consider this question first for a general partitioned matrix

$$M = \begin{bmatrix} M' \\ \hline M'' \end{bmatrix}.$$

The common linear combinations of the row span of M' and the row span of M'' can be computed from the left kernel of M. Indeed,

$$[\![kM = 0 \leftrightarrow \left[k' \,\middle|\, k''\right] \left[\begin{array}{c} M' \\ \hline M'' \end{array}\right] = 0]\!] \Leftrightarrow [\![k'M' = -k''M'']\!],$$

and hence the common linear combinations of the span of the rows of M' and M'' follow immediately from a set of vectors that span the left kernel of M, by truncating these vectors conformably with the partition of the matrix M, to k', and multiplying by M'. This can be applied to the partitioned Hankel matrix $(\mathscr{H}_p/\mathscr{H}_f)$, and we observe that the state construction amounts to computing the left kernel of the partitioned matrix $(\mathscr{H}_p/\mathscr{H}_f)$.

We shall now argue that, *because of the Hankel structure*, the left kernel of $(\mathscr{H}_p/\mathscr{H}_f)$ can be deduced from the left kernel of $\mathscr{H}_p$ all by itself, and so, there is no need to use the past/future partitioning in order to construct the left kernel and the state. To see this, assume that

$$\left[\, k_1 \; k_2 \; \cdots \; k_{T_p} \,\right]$$

is in the left kernel of $\mathscr{H}_p$, i.e. $k\mathscr{H}_p = 0$. Notice that, because of the Hankel structure of $\mathscr{H}$, the vectors

$$\left[\, 0 \cdots 0 \; k_1 \; k_2 \; \cdots \; k_{T_p} \; 0 \cdots 0 \,\right],$$

obtained by putting in total T_f zeros in front and in back of $\left[\, k_1 \; k_2 \; \cdots \; k_{T_p} \,\right]$, are all contained in the left kernel of $(\mathscr{H}_p/\mathscr{H}_f)$. It can be shown that, provided T_p is sufficiently large (but it need not be larger that what was required to validate the intersection argument of the row spans of $\mathscr{H}_p$ and $\mathscr{H}_f$ of the previous section), we obtain this way, from a set of vectors that span the left kernel of $\mathscr{H}_p$, a set of vectors that span the whole left kernel of $(\mathscr{H}_p/\mathscr{H}_f)$. After truncation to its first T_{p} elements,

$$\left[\, 0 \cdots 0 \; k_1 \; k_2 \; \cdots \; k_L \,\right],$$

this leads to a set of vectors that, when multiplied from the right with $\mathscr{H}_p$, span the intersection of the spaces spanned by the rows of $\mathscr{H}_p$ and the rows of $\mathscr{H}_f$. Note that this truncation results from applying repeatedly the shift-and-cut operator to the row vector $\left[\, k_1 \; k_2 \; \cdots \; k_{T_p} \,\right]$, i.e. putting a zero in the first element and deleting the last element of this row vector, so as to obtain a vector of length T_p. In other words, from the vector

$$\left[\, k_1 \; k_2 \; \cdots \; k_{T_p} \,\right]$$

in the left kernel of $\mathscr{H}_p$, we obtain the vectors

$$\begin{array}{c}
[\; 0 \; k_1 \; k_2 \; \cdots \; k_{T_{p-2}} \; k_{T_{p-1}} \;] \\
[\; 0 \; 0 \; k_1 \; \cdots \; k_{T_{p-3}} \; k_{T_{p-2}} \;] \\
\vdots \\
[\; 0 \; 0 \; 0 \; \cdots \; k_1 \; k_2 \;] \\
[\; 0 \; 0 \; 0 \; \cdots \; 0 \; k_1 \;]
\end{array}$$

that are truncations of elements from the left kernel of $\left[\begin{array}{c} \mathscr{H}_f \\ \hline \mathscr{H}_p \end{array}\right]$

Using the ideas explained in the previous section, this leads to the construction of the state trajectory associated with $\tilde{w}$ in the MPUM, by computing a basis of the left kernel of $\mathscr{H}_p$, stacking these vectors as the rows of the matrix

$$\left[\, K_1 \; K_2 \; \cdots \; K_{T_{p-1}} \; K_{T_p} \,\right],$$

and repeatedly applying the shift-and-cut operator to obtain the state trajectory

$$\left[\, \tilde{x}(T_p+1) \; \tilde{x}(T_p+2) \; \cdots \; \tilde{x}(T_p+t) \; \cdots \,\right] =$$
$$\begin{bmatrix} 0 & K_1 & K_2 & \cdots & K_{T_p-2} & K_{T_p-1} \\ 0 & 0 & K_1 & \cdots & K_{T_p-3} & K_{T_p-2} \\ \vdots & \vdots & \vdots & \vdots\vdots\vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & K_1 & K_2 \\ 0 & 0 & 0 & \cdots & 0 & K_1 \end{bmatrix} \begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t+2) & \cdots \\ \vdots & \vdots & \vdots\vdots\vdots & \vdots & \vdots\vdots\vdots \\ \tilde{w}(T_p-1) & \tilde{w}(T_p) & \cdots & \tilde{w}(T_p+t-2) & \cdots \\ \tilde{w}(T_p) & \tilde{w}(T_p+1) & \cdots & \tilde{w}(T_p+t-3) & \cdots \end{bmatrix}.$$

Actually, it turns out that we can also apply the shift-and-cut backwards, leading to

$$\left[\, \tilde{x}(1) \;\; \tilde{x}(2) \;\; \cdots \;\; \tilde{x}(t) \;\; \cdots \,\right] =$$
$$\begin{bmatrix} K_2 & K_3 & \cdots & K_{T_p-1} & K_{T_p} \\ K_3 & K_4 & \cdots & K_{T_p} & 0 \\ \vdots & \vdots & \vdots\vdots\vdots & \vdots & \vdots \\ K_{T_p-1} & K_{T_p} & \cdots & 0 & 0 \\ K_{T_p} & 0 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \vdots & \vdots & \vdots\vdots\vdots & \vdots & \vdots\vdots\vdots \\ \tilde{w}(T_p-2) & \tilde{w}(T_p-1) & \cdots & \tilde{w}(T_p+t-3) & \cdots \\ \tilde{w}(T_p-1) & \tilde{w}(T_p) & \cdots & \tilde{w}(T_p+t-2) & \cdots \end{bmatrix}.$$

This then yields the desired state trajectory to which the subspace algorithm ($) can be applied in order to obtain an i/s/o representation of the MPUM.

6 The left kernel of a Hankel matrix

In the previous section, we have seen the relevance to the problem at hand of computing the left kernel of the doubly infinite Hankel matrix $\mathscr{H}$. We are interested in characterizing the infinite vectors in its left kernel that have 'compact support', i.e. the infinite vectors of the form

$$k = \left[\, k_1 \; k_2 \; \cdots \; k_t \; \cdots \; 0 \; \cdots \; 0 \; \cdots \,\right], \quad k_t \in \mathbb{R}^{1\times \mathtt{w}}, t \in \mathbb{N},$$

with $k\mathscr{H} = 0$. Denote the set of compact support elements in the left kernel by $\mathscr{N}$. For simplicity, we call $\mathscr{N}$ the left kernel of $\mathscr{H}$.

In general, $\mathscr{N}$ is infinite dimensional. In fact, $\mathscr{N}$ equals $\{0\}$, or it is infinite dimensional. However, we shall now argue that by considering the left kernel of $\mathscr{H}$ as a module, $\mathscr{N}$ is effectively finite dimensional, of dimension $\leq \mathtt{w}$. Observe that $\mathscr{N}$ is closed under addition (obvious), scalar multiplication (obvious), and under the right shift (also obvious, using the Hankel structure):

$$[\![\begin{bmatrix} k_1 \; k_2 \; \cdots \; k_t \; 0 \; 0 \cdots \; 0 \cdots \end{bmatrix} \in \mathscr{N}]\!] \Rightarrow [\![\begin{bmatrix} 0 \; k_1 \; \cdots \; k_{t-1} \; k_t \; 0 \cdots \; 0 \cdots \end{bmatrix} \in \mathscr{N}]\!].$$

This implies (identify elements $k \in \mathscr{N}$ with polynomial vectors $k_1 + k_2\xi + \cdots + K_t\xi^{t-1} + \cdots \in \mathbb{R}[\xi]^{1\times \mathtt{w}}$, and the right shift with multiplication by ξ) that $\mathscr{N}$ has the structure of a module (a submodule of $\mathbb{R}[\xi]^{1\times \mathtt{w}}$, viewed as an $\mathbb{R}[\xi]$-module). This submodule is finitely generated (all $\mathbb{R}[\xi]$-submodules of $\mathbb{R}[\xi]^{1\times \mathtt{w}}$ are finitely generated, with at most $\mathtt{w}$ generators). This means that there exist elements $n_1, n_2, \ldots, n_{\mathtt{p}} \in \mathscr{N}$, with $\mathtt{p} \leq \mathtt{w}$, such that all other elements of $\mathscr{N}$ can be obtained as linear combinations of these elements and their repeated right shifts.

It turns out that *for the construction of the state trajectory, we need only these generators*. In other words, rather that compute the whole left kernel of $\mathscr{H}_p$, it suffices to obtain a set of generators of the left kernel of $\mathscr{H}$ in the left kernel of $\mathscr{H}_p$. We assume that L_p is sufficiently large, so that the left kernel of $\mathscr{H}_p$ contains a set of generators of the left kernel of $\mathscr{H}$.

This leads to the following state construction algorithm. Let $n_1, n_2, \cdots, n_{\mathtt{p}}$ of $\mathscr{N}$ be a set of generators of the left kernel of $\mathscr{H}$. Truncate these vectors at their last non-zero element:

$$n_i \cong \begin{bmatrix} n_{i,1} \; n_{i,2} \; \cdots \; n_{i,L_i} \end{bmatrix}, \quad n_{i,t} \in \mathbb{R}^{1\times \mathtt{w}}.$$

Now apply the shift-and-cut to the i-th generator. This leads to the 'partial' state trajectory

$$\begin{bmatrix} \tilde{x}_i(1) & \tilde{x}_i(2) & \cdots & \tilde{x}_i(t) & \cdots \end{bmatrix} =$$
$$\begin{bmatrix} n_{i,2} & n_{i,3} & \cdots & n_{i,L_i-1} & n_{i,L_i} \\ n_{i,3} & n_{i,4} & \cdots & n_{i,L_i} & 0 \\ \vdots & \vdots & \vdots\vdots\vdots & \vdots & \vdots \\ n_{i,L_i-1} & n_{i,L_i} & \cdots & 0 & 0 \\ n_{i,L_i} & 0 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \vdots & \vdots & \vdots\vdots\vdots & \vdots & \vdots\vdots\vdots \\ \tilde{w}(L_i-2) & \tilde{w}(L_i-1) & \cdots & \tilde{w}(L_i+t-3) & \cdots \\ \tilde{w}(L_i-1) & \tilde{w}(L_i) & \cdots & \tilde{w}(L_i+t-2) & \cdots \end{bmatrix}.$$

Now, stack the $\tilde{x}_i$'s. This leads to the state trajectory

$$\begin{bmatrix} \tilde{x}_1(1) \\ \tilde{x}_2(1) \\ \vdots \\ \tilde{x}_{\mathtt{p}}(1) \end{bmatrix}, \begin{bmatrix} \tilde{x}_1(2) \\ \tilde{x}_2(2) \\ \vdots \\ \tilde{x}_{\mathtt{p}}(2) \end{bmatrix}, \cdots, \begin{bmatrix} \tilde{x}_1(t) \\ \tilde{x}_2(t) \\ \vdots \\ \tilde{x}_{\mathtt{p}}(t) \end{bmatrix}, \cdots$$

to which the subspace algorithm ($) can be applied.

In conclusion, from a set of generators of $\mathscr{N}$ viewed as a module, we obtain, by repeatedly using the shift-and-cut, and matrix which multiplied by the Hankel matrix of the data, yields the state trajectory $\tilde{x}$. In the next section, we provide a recursive way of computing a set of generators.

7 Recursive computation of a module basis

Consider the left kernel $\mathscr{N}$ of $\mathscr{H}$. Call a minimal set of elements $n_1, n_2, \ldots, n_{\mathtt{p}} \in \mathscr{N}$ that generate all of $\mathscr{N}$, through linear combinations of these elements and their repeated right shifts, a *module basis* of $\mathscr{N}$. In the present section, we set up a recursive algorithm to compute a module basis of $\mathscr{N}$ from $\tilde{w}$.

Assume henceforth that the MPUM in $\mathscr{L}^{\mathtt{w}}$ associated with $\tilde{w}$ is controllable. This assumption is made for reasons of exposition. The algorithm can be generalized without this assumption, but it becomes considerably more involved to explain.

We start with a brief digression about controllability. The following basic property characterizes controllable behaviors in $\mathscr{L}^{\mathtt{w}}$:

$$[\![\mathscr{B} \in \mathscr{L}^{\mathtt{w}} \text{ is controllable}]\!] \Leftrightarrow [\![\exists \mathscr{B}' \in \mathscr{L}^{\mathtt{w}} \text{ such that } \mathscr{B} \oplus \mathscr{B}' = (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}]\!].$$

Evidently, $\mathscr{B}'$ is also controllable. In other words, a behavior has a direct complement iff it is controllable. This property of controllable behaviors can be translated in terms of a kernel representation $(\mathscr{K})$ of $\mathscr{B}$, with R of full row rank (every $\mathscr{B} \in \mathscr{L}^{\mathtt{w}}$ allows such a full row rank representation). It states that

$$[\![\text{kernel}\,(R\,(\sigma)) \in \mathscr{L}^{\mathtt{w}} \text{ is controllable}]\!]$$
$$\Leftrightarrow [\![\exists R' \in \mathbb{R}\,[\xi]^{\bullet\times\mathtt{w}} \text{ such that } \begin{bmatrix} R \\ R' \end{bmatrix} \text{ is unimodular}]\!].$$

In fact, $R'(\sigma)w = 0$ is a kernel representation of the direct complement $\mathscr{B}'$.

We now return to the construction of the MPUM. Start with the data $\tilde{w}$. Consider the associated Hankel matrix $\mathscr{H}$, and its consecutive truncations

$$\begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \end{bmatrix},$$

$$\begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \end{bmatrix},$$

$$\vdots$$

$$\begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \vdots & \vdots & \vdots\vdots\vdots & \vdots & \vdots\vdots\vdots \\ \tilde{w}(L) & \tilde{w}(L+1) & \cdots & \tilde{w}(t+L-1) & \cdots \end{bmatrix},$$

until a vector in the left kernel is obtained. Denote this element by

$$\begin{bmatrix} k_1 \; k_2 \; \cdots \; k_L \end{bmatrix} \in \mathbb{R}^{1\times\mathtt{w}L}.$$

Now consider the corresponding vector polynomial

$$n(\xi) = k_1 + k_2\xi + \cdots + k_L\xi^{L-1} \in \mathbb{R}\,[\xi]^{1\times\mathtt{w}}.$$

It can be shown that, because of the assumed controllability of the MPUM, the system with kernel representation $n(\sigma)w = 0$ is also controllable. Consequently, there exists $N \in \mathbb{R}[\xi]^{(\mathtt{w}-1)\times\mathtt{w}}$ such that the polynomial matrix

$$\begin{bmatrix} n \\ \hline N \end{bmatrix}$$

is unimodular. The system described by $n(\sigma)w = 0$ is unfalsified by $\tilde{w}$, but, of course, $N(\sigma)w = 0$ need not be. Compute the 'error' vector time series

$$\tilde{e} = N(\sigma)\tilde{w} = (\tilde{e}(1), \tilde{e}(2), \dots, \tilde{e}(t), \dots),$$

Now apply the above algorithm again, with $\tilde{w}$ replaced by $\tilde{e}$, and proceed recursively. Note that $\tilde{e}(t) \in \mathbb{R}^{\mathtt{w}-1}$: the dimension of the time series that needs to be examined goes down by one at each step.

Recursively this leads to the algorithm

$$\tilde{w} \mapsto n_1 \mapsto N_1 \mapsto \tilde{e}_1 \mapsto n_2 \mapsto N_2 \mapsto \tilde{e}_2 \mapsto \cdots \mapsto \tilde{e}_{\mathtt{p}-2} \mapsto n_{\mathtt{p}-1} \mapsto N_{\mathtt{p}-1} \mapsto \tilde{e}_{\mathtt{p}-1} \mapsto n_{\mathtt{p}}.$$

This algorithm terminates when there are no more vectors in the left kernel of the associated Hankel matrix, i.e. when the error $\tilde{e}$ is persistently exciting. If we assume that the MPUM has $\mathtt{m}$ input and $\mathtt{p}$ output components, then $\tilde{e}_{\mathtt{p}}$ will be the first persistently exciting error time series obtained.

Now consider the polynomial vectors

$$r_1 = n_1, r_2 = n_2N_1, r_3 = n_3N_2N_1, \cdots, r_{\mathtt{p}} = n_{\mathtt{p}}N_{\mathtt{p}-1}N_{\mathtt{p}-2}\cdots N_2N_1.$$

Define

$$\tilde{R} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_{\mathtt{p}} \end{bmatrix}.$$

It can be shown that

$$\tilde{R}(\sigma)w = 0$$

is a kernel representation of $\tilde{\mathscr{B}}$, the MPUM in $\mathscr{L}^{\mathtt{w}}$ associated with $\tilde{w}$. The intermediate calculations of $r_1, r_2, \dots, r_{\mathtt{p}}$ lead to the state trajectory in a similar way as explained in section 6. Let

$$r_i(\xi) = r_{i,1} + r_{i,2}\xi + \cdots + r_{i,L_i}\xi^{L_1-1} \ \in \mathbb{R}[\xi]^{1\times\mathtt{w}}.$$

Form the vector

$$r_i \cong \left[\, r_{i,1} \; r_{i,2} \; \cdots \; r_{i,L_i} \,\right], \quad r_{i,t} \in \mathbb{R}^{1\times\mathtt{w}}.$$

Now apply the shift-and-cut. This leads to the 'partial' state trajectory

$$\begin{bmatrix} \tilde{x}_i(1) & \tilde{x}_i(2) & \cdots & \tilde{x}_i(t) & \cdots \end{bmatrix} =$$
$$\begin{bmatrix} r_{i,2} & r_{i,3} & \cdots & r_{i,L_i-1} & r_{i,L_i} \\ r_{i,3} & r_{i,4} & \cdots & r_{i,L_i} & 0 \\ \vdots & \vdots & \vdots\vdots\vdots & \vdots & \vdots \\ r_{i,L_i-1} & r_{i,L_i} & \cdots & 0 & 0 \\ r_{i,L_i} & 0 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \vdots & \vdots & \vdots\vdots\vdots & \vdots & \vdots\vdots\vdots \\ \tilde{w}(L_i-2) & \tilde{w}(L_i-1) & \cdots & \tilde{w}(L_i+t-3) & \cdots \\ \tilde{w}(L_i-1) & \tilde{w}(L_i) & \cdots & \tilde{w}(L_i+t-2) & \cdots \end{bmatrix}.$$

Now, stack the $\tilde{x}_i$'s, and obtain the state trajectory

$$\begin{bmatrix} \tilde{x}_1(1) \\ \tilde{x}_2(1) \\ \vdots \\ \tilde{x}_{\mathrm{p}}(1) \end{bmatrix}, \begin{bmatrix} \tilde{x}_1(2) \\ \tilde{x}_2(2) \\ \vdots \\ \tilde{x}_{\mathrm{p}}(2) \end{bmatrix}, \cdots, \begin{bmatrix} \tilde{x}_1(t) \\ \tilde{x}_2(t) \\ \vdots \\ \tilde{x}_{\mathrm{p}}(t) \end{bmatrix}, \cdots$$

to which the subspace algorithm ($) can be applied.

8 Concluding remarks

8.1 Subspace ID

Solving equation ($) as the basis for system identification has many appealing features. We can begin by reducing the number of rows of

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

by numerically approximating this matrix by one with fewer rows. This leads to a reduction of the state dimension and hence of the model complexity. Of course, ($) will then no longer be exactly solvable, even when the observed data is noise free, but since this equation is linear in the unknown matrices A, B, C, D, it is very amenable to a least-squares (LS) solution. Introducing the state in a sense linearizes the SYSID problem. Solving equation ($) using (LS) also accommodates for noisy data and for numerical errors in the intermediate calculations.

Missing data can be dealt with by deleting columns in the equation ($). If multiple time series are observed (this is the case in classical realization theory), then equation ($) can readily be extended with the vectors $\tilde{u}, \tilde{y}, \tilde{x}$ replaced by matrices.

8.2 State construction by shift-and-cut

The state construction that permeates sections 5, 6, and 7 is actually well-known, and our presentation of it via the past/future partition and the left kernel of the Hankel matrix $\mathscr{H}$ to some extent hides the simplicity and generality of the ideas behind it.

Indeed, in [9], a state construction algorithm based on the shift-and-cut operator has been presented as a very direct and general method for constructing state representations, starting from a variety of other system representations. Let $p \in \mathbb{R}[\xi]$ and define the *shift-and-cut* operator $\sigma^* : \mathbb{R}[\xi] \to \mathbb{R}[\xi]$ by

$$\sigma^* : \ p_0 + p_1\xi + \cdots + p_\mathrm{L}\xi^\mathrm{L} \mapsto p_1 + p_2\xi + \cdots + p_\mathrm{L}\xi^{\mathrm{L}-1}.$$

By applying the operator σ^* element-wise, it is readily extended to polynomial vectors and matrices.

We now explain how this state construction works, starting with a kernel representation ($\mathscr{K}$). Associate with

$$R(\xi) = R_0 + R_1\xi + \cdots + R_L\xi^L$$

the stacked polynomial matrix

$$X(\xi) = \begin{bmatrix} \sigma^* R \\ \sigma^{*2} R \\ \vdots \end{bmatrix}(\xi) = \begin{bmatrix} R_1 + R_2\xi + R_3\xi^3 + \cdots + R_{L-1}\xi^{L-2} + R_L\xi^{L-1} \\ R_2 + R_3\xi + \cdots + R_{L-1}\xi^{L-3} + R_L\xi^{L-2} \\ \vdots \\ R_{L-1} + R_L\xi \\ R_L \end{bmatrix}(\xi),$$

obtained by repeatedly applying σ^* to R until we get the zero matrix. It can be shown that $X(\sigma)$ is a *state map*: it associates to $w \in \operatorname{kernel}(R(\sigma))$ the corresponding state trajectory $x = X(\sigma)w$ of a (in general non-minimal) state representation of ($\mathscr{K}$). In other words, as soon as we have a kernel representation of the MPUM, the shift-and-cut operator gives us the underlying state.

This shift-and-cut state construction is precisely what is done in sections 5, 6, and 7. Starting from the MPUM as

$$\tilde{\mathscr{B}} = \overline{\text{linear span } (\{\tilde{w}, \sigma\tilde{w}, \ldots, \sigma^t\tilde{w}, \ldots\})},$$

we construct annihilators for $\tilde{\mathscr{B}}$. These annihilators are, of course, exactly the elements of the left kernel of the Hankel matrix $\mathscr{H}$. By subsequently applying the shift-and-cut operator, we obtain the state trajectory $\tilde{x}$ corresponding to $\tilde{w}$.

8.3 Return to the data

One of the attractive features of subspace methods, is that after construction of $\tilde{x}$, the model is obtained using equation ($) which involves $\tilde{w}$ and $\tilde{x}$. In other words, it allows to return to the original observed time series $\tilde{w}$ in order to fit the final parameter estimates of the identified system to the data.

The shift-and-cut state construction shows how to obtain a state trajectory from any kernel representation of the MPUM. We originally posed the question of how to construct the state trajectory by avoiding the intermediate computation of a kernel representation of the MPUM. But, we have come full circle on this. We demonstrated that the state construction based on the intersection of the row spans of $\mathscr{H}_p$ and $\mathscr{H}_f$ actually amounts to finding a module basis of a kernel representation of the MPUM. It is an interesting matter to investigate to what extent these insights can also be used in the purely stochastic or in the mixed deterministic/stochastic case.

8.4 Approximation and balanced reduction

The algorithm proposed in section 7 lends itself very well for approximate implementation. Checking whether the consecutive truncations of the Hankel matrix have, up to reasonable level of approximation, a non-trivial element in the left kernel, and finding the optimal element in the left kernel, are typical decisions that can be made using SVD based numerical linear algebra computations.

It is of interest to combine our recursive algorithms with model reduction. In particular, it ought to be possible to replace the state construction based on the shift-and-cut operator applied to an annihilator by an alternative set of low order polynomial vectors that lead to a balanced state model. Some ideas in this direction have been given in [12].

8.5 The complementary system

The most original feature of this article is the recursive computation of a basis of the module of annihilators of a given behavior in section 7. In the controllable case, this may be done by complementing a kernel representation to a unimodular polynomial matrix. It is of interest to explore if the recursive computation of the module basis explained in section 7, combined with the shift-and-cut map, can also be used in the general constructions of a state map, starting from a kernel, an image, or a latent variable representation of a behavior.

The finer features and numerical aspects of this recursive computation and extension of a polynomial matrix to a unimodular one is a matter of future research. In particular, one is led to wonder if the $\ell_2(\mathbb{N},\mathbb{R}^{\mathtt{w}})$-orthogonal complement of $\mathscr{B}\cap\ell_2(\mathbb{N},\mathbb{R}^{\mathtt{w}})$ can play a role in obtaining a complement of a $\mathscr{B}$. Or if the singular value decomposition of the truncated Hankel matrix

$$\begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \vdots & \vdots & \vdots\vdots\vdots & \vdots & \vdots\vdots\vdots \\ \tilde{w}(L) & \tilde{w}(L+1) & \cdots & \tilde{w}(t+L-1) & \cdots \end{bmatrix}$$

can be used for complementing n with N to obtain a unimodular polynomial matrix. The left singular vector corresponding to the smallest singular value should serve to identify the element n in the left kernel, and the others somehow to find the complement N.

Acknowledgments

This research is supported by the Research Council KUL project CoE EF/05/006 (OPTEC), Optimization in Engineering, and by the Belgian Federal Science Policy Office: IUAP P6/04 (Dynamical systems, Control and Optimization, 2007-2011).

References

1. H. Akaike, Markovian representation of stochastic processes by canonical variables, *SIAM Journal on Control*, volume 13, pages 162–173, 1975.
2. A. Chiuso and G. Picci, Asymptotic variance of subspace estimates, *Journal of Econometrics*, volume 118, pages 257–291, 2004.
3. A. Chiuso and G. Picci, Consistency analysis of certain closed-loop subspace identification methods, *Automatica*, volume 41, pages 377–391, 2005.
4. D. Bauer, Comparing the CCA subspace method to pseudo maximum likelihood methods in the case of no exogenous inputs, *Journal of Time Series Analysis*, volume 26, pages 631–668, 2005.
5. M. Deistler, K. Peternell, and W. Scherrer, Consistency and relative efficiency of subspace methods, *Automatica*, volume 31, pages 185–1875, 1995.
6. W.E. Larimore, System identification, reduced order filters and modeling via canonical variate analysis, *Proceedings of the American Control Conference*, pages 445-451, 1983.
7. P. Van Overschee and B. L. M. De Moor, N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems, *Automatica*, volume 30, pages 75-93, 1994.
8. P. Van Overschee and B. L. M. De Moor, *Subspace Identification for Linear Systems: Theory, Implementation, Applications*, Kluwer Academic Press, 1996.
9. P. Rapisarda and J.C. Willems, State maps for linear systems, *SIAM Journal on Control and Optimization*, volume 35, pages 1053-1091, 1997.
10. M. Verhaegen, Identification of the deterministic part of MIMO state space models given in innovations form from input-output data, *Automatica*, volume 30, pages 61–74, 1994.
11. J. C. Willems, From time series to linear system, Part I. Finite dimensional linear time invariant systems, Part II. Exact modelling, Part III. Approximate modelling, *Automatica*, volume 22, pages 561-580 and 675-694, 1986, volume 23, pages 87-115, 1987.
12. J.C. Willems and P. Rapisarda, Balanced state representations with polynomial algebra, in *Directions in Mathematical Systems Theory and Optimization*, (edited by A. Rantzer and C.I. Byrnes), Springer Lecture Notes in Control and Information Sciences, volume 286, pages 345-357, 2002.
13. J.C. Willems, P. Rapisarda, I. Markovsky, and B. De Moor, A note on persistency of excitation, *Systems & Control Letters*, volume 54, pages 325-329, 2005.
14. J.C. Willems, Thoughts on system identification, in *Control of Uncertain Systems: Modelling, Approximation and Design* (edited by B.A. Francis, M.C. Smith, and J.C. Willems), Springer Verlag Lecture Notes on Control and Information Systems, volume 329, pages 389–416, 2006.