RECURSIVE COMPUTATION OF THE MOST POWERFUL UNFALSIFIED MODEL

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

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

Ivan.Markovsky@esat.kuleuven.be
Jan.Willems@esat.kuleuven.be
Bart.DeMoor@esat.kuleuven.be
http://www.esat.kuleuven.be/sista

Tel: +32–16–32 17 09, *Fax: +32–16–32 19 70*

Abstract: We consider algorithms for recursive computation of an exact model for a given time series. Existing algorithms that operate recursively in time are reviewed and a new algorithm that operates recursively in the degrees of the system laws is described.

Keywords: Most powerful unfalsified model, recursive identification, partial realization, Berlekamp–Massey algorithm.

1. INTRODUCTION

In this paper we consider exact recursive modeling of time series by a linear time-invariant (LTI) system. The algorithms might operate recursively in

- the number of time series,
- the length of the time series,
- the length of the system laws,

or in a combination of these. The problem can be viewed alternatively as a derivation of computationally efficient algorithms for exact modeling.

Most system identification methods operate on Hankel matrices constructed from the given time series. Under appropriate conditions, see (Willems *et al.*, 2005), the kernel of the Hankel matrix contains all laws of an LTI system that fits the data exactly. Approximate identification methods find "approximate" in a certain specified sense kernel of the Hankel matrix of the data. Therefore an issue for both exact and approximate system identification is how to take into account the Hankel structure in effective computational algorithms.

There is a large body of linear algebra literature on efficient numerical methods for structured matrices that is relevant for the problem at hand. We take, however, a system theoretic approach. We step on the results of (Willems, 1987; Antoulas and Willems, 1993; Kuijper and Willems, 1997; Kuijper, 1997) where the behavioral approach to exact modeling is developed. Our contribution is a new algorithm that works recursively in the degrees of the system laws.

1.1 The most powerful unfalsified model

Denote by $\mathscr{L}^{\mathtt{w}}$ the finite dimensional discrete-time LTI model class with $\mathtt{w}$ external variables. We consider the system trajectories for positive time $1,2,\ldots$, so that a system $\mathscr{B} \in \mathscr{L}^{\mathtt{w}}$ is a linear, shift-invariant, closed (in the topology of pointwise convergence) subspace of the signal space $(\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$. We call a system $\mathscr{B}$ more powerful than a system $\mathscr{B}'$ if $\mathscr{B} \subseteq \mathscr{B}'$. Consider the data set $\mathscr{D} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$. We call $\mathscr{B}$ unfalsified by the data $\mathscr{D}$ if $\mathscr{D} \subseteq \mathscr{B}$. The fact that $\mathscr{B}$ is unfalsified by the data $\mathscr{D}$ means that $\mathscr{D}$ could have been generated by $\mathscr{B}$. The most powerful unfalsified model (MPUM)

$\mathscr{B}_{\mathrm{mpum}}(\mathscr{D})$ of the data $\mathscr{D} \subseteq (\mathbb{R}^{\mathrm{w}})^{\mathbb{N}}$ in $\mathscr{L}^{\mathrm{w}}$ is a model that is:

a. in the model class $\mathscr{L}^{\mathrm{w}}$, *i.e.*, $\mathscr{B} \in \mathscr{L}^{\mathrm{w}}$
b. unfalsified by the data $\mathscr{D}$, *i.e.*, $\mathscr{D} \subseteq \mathscr{B}$, and
c. any other unfalsified system in $\mathscr{L}^{\mathrm{w}}$ is less powerful, *i.e.*, $\mathscr{B}' \in \mathscr{L}^{\mathrm{w}}$ and $\mathscr{D} \subseteq \mathscr{B}'$ imply $\mathscr{B} \subseteq \mathscr{B}'$.

The MPUM $\mathscr{B}_{\mathrm{mpum}}(\mathscr{D})$ exists and is obviously unique for all $\mathscr{D} \subseteq (\mathbb{R}^{\mathrm{w}})^{\mathbb{N}}$, see (Willems, 1986*b*, Theorem 13).

Any finite dimensional LTI system allows a representation as the solution set of a constant coefficient difference equation. Denote by σ the backwards shift operator

$$\sigma w(t) = w(t+1).$$

If $\mathscr{B} \in \mathscr{L}^{\mathrm{w}}$, then there exist an integer g and a polynomial matrix $R \in \mathbb{R}^{g\times \mathrm{w}}[\xi]$, such that

$$\mathscr{B} = \{\, w \mid R(\sigma)w = 0 \,\}. \qquad \text{(Ker)}$$

The representation (Ker) is called a kernel representation of $\mathscr{B}$.

The lag of $\mathscr{B} \in \mathscr{L}^{\mathrm{w}}$, denoted by $\mathbf{l}(\mathscr{B})$, is the smallest degree of R, where $\ker\big(R(\sigma)\big) = \mathscr{B}$. Equivalently $\mathbf{l}(\mathscr{B})$ is the observability index of an input/state/output representation of $\mathscr{B}$.

1.2 Recursive modeling

Let the given data be a finite collection of time series on $\mathbb{N}$

$$\mathscr{D} = \{\, w_{\mathrm{d}}^{(1)}, \ldots, w_{\mathrm{d}}^{(N)} \,\}.$$

The following recursive algorithm computes the MPUM. Assume that $R^{(k)} \in \mathbb{R}^{g\times \mathrm{w}}$ induces via (Ker) the MPUM of the first k time series $\{\, w_{\mathrm{d}}^{(1)}, \ldots, w_{\mathrm{d}}^{(k)} \,\}$. The kth stage in the recursive construction of the MPUM of $\mathscr{D}$ is:

1. compute the error

$$e^{(k+1)} := R^{(k)}(\sigma) w_{\mathrm{d}}^{(k+1)},$$

obtained when the kth model

$$\mathscr{B}^{(k)} := \ker\big(R^{(k)}(\sigma)\big)$$

is applied on the $(k+1)$st time series;
2. let $E^{(k+1)}$ induces $\mathscr{B}_{\mathrm{mpum}}(e^{(k)})$;
3. define $R^{(k+1)} := E^{(k+1)} R^{(k)}$.

Initialized with $R^{(0)} = I$, this algorithm gives at the Nth stage the MPUM of $\mathscr{D}$. Step 1 is a simulation and step 3 is convolution of the coefficients of $e^{(k+1)}$ and $R^{(k)}$. The unspecified step 2 is the crucial one. As discussed next, however, there is an important special case when the error $e^{(k)}$ can be modeled easily.

The recursive in the number of time series algorithm can be used for recursive in time modeling as well. Now we consider a single time series w_{d}. At each stage of the recursion we update the MPUM on the previous stage by taking into account one more sample of the time series. Assuming that the MPUM of $\sigma^T w_{\mathrm{d}}$ is known (*e.g.*, because $\sigma^T w_{\mathrm{d}} = 0$, in which case the corresponding $R^{(T)} = I$), we do the recursion backwards in time. Note that the error

$$e^{(k-1)} := R^{(k)}(\sigma)\sigma^{k-1} w_{\mathrm{d}}$$

on each stage is a pulse and its MPUM is given explicitly by a first order kernel representation.

The lag of the kernel representation (Ker) is the degree of the polynomial matrix R. The rows of R are referred to as laws of the model $\mathscr{B}$. Recursive in the lag computation of the MPUM proceeds sequentially over degree $1, 2, \ldots$ laws exploiting the fact that if $r \in \mathbb{R}^{1\times \mathrm{w}}[\xi]$ is a law of the MPUM (*i.e.*, $r(\sigma)\mathscr{B} = 0$), then $\xi^k r(\xi)$, $k = 1, 2, \ldots$ are also laws of $\mathscr{B}$. In other words the repeated computation of one and the same law (by shifting it) is avoided on the later stages of the recursion. The recursive in lag modeling problem is closely related to the problem of exploiting the kernel structure of a Hankel matrix in numerical linear algebra operations, see (Heinig and Rost, 1984; Heinig and Jankowski, 1992).

2. TIME-RECURSIVE COMPUTATION OF THE MPUM

The question remains what to do for a finite observed time series

$$w_{\mathrm{d}} = \big(w_{\mathrm{d}}(1), \ldots, w_{\mathrm{d}}(T)\big).$$

For a polynomial matrix

$$R = R_0 + R_1\xi + \cdots + R_l\xi^l, \quad \text{with } R_l \neq 0$$

define the reciprocal polynomial matrix

$$\mathrm{Rec}(R) := R_l + R_{l-1}\xi + \cdots + R_0\xi^l.$$

In (Willems, 1997) a recursive procedure for modeling vector time series is put forward. Algorithm 1, given below, is specialized for the scalar case and is equivalent to the Berlekamp–Massey decoding algorithm. It operates on a 2×2 polynomial matrix $R^{(k)}$. On each stage k, the $(1,1)$ entry of the reciprocal of the first row $R^{(k)}_{1,1:2}$ of $R^{(k)}$ gives the MPUM of the time series. Next we show a simulation example.

Example 1. (Fibonacci numbers). Consider the sequence

$$w_{\mathrm{d}} = (0, 1, 1, 2, 3, 5, 8, 13)$$

of the first 8 Fibonacci numbers. The MPUM of w_{d} is the first order autonomous system

$$\mathscr{B} = \{\, w \in \mathbb{R}^{\mathbb{N}} \mid w(t+2) = w(t+1) + w(t) \,\}$$

or in a polynomial notation

$$\mathscr{B} = \ker\big(R(\sigma)\big), \quad \text{with} \quad R(\xi) = \xi^2 - \xi - 1.$$

The results per stage from applying Algorithm 1 on w_{d} are given in Table 1. At stage 5 the algorithm computes $\mathscr{B}_{\mathrm{mpum}}(w_{\mathrm{d}})$ and

$$[\mathrm{Rec}(R^{(t)}_{1,1:2})]_{11} = \xi^2 - \xi - 1, \quad \text{for } t = 5, 6, \ldots$$

Algorithm 1 Time-recursive MPUM computation `time_recursive_mpum(`w_d`)`

Input: A scalar time series $w_\mathrm{d} \in \mathbb{R}^T$.

1: Initialization: $R^{(0)} = I$ and $l^{(0)} = 0$.
2: **for** $k = 1, \ldots, T$ **do**
3: Compute
$$e^{(k)} := R_{11}^{(k-1)}(\sigma)\big(w_\mathrm{d}(1), \ldots, w_\mathrm{d}(k)\big).$$
4: **if** $e^{(k)} = 0$ or $l^{(k-1)} > k/2$ **then**
5: Let $E^{(k)}(\xi) := \begin{bmatrix} 1 & -e^{(k)} \\ 0 & \xi \end{bmatrix}$ and $l^{(k)} := l^{(k-1)}$.
6: **else**
7: Let $E^{(k)}(\xi) := \begin{bmatrix} 1 & -e^{(k)} \\ \xi/e^{(k)} & 0 \end{bmatrix}$ and $l^{(k)} := k - l^{(k-1)}$.
8: **end if**
9: Compute $R^{(k)} := E^{(k)} R^{(k-1)}$.
10: **end for**
11: Let R be the element (1,1) of the reciprocal $\mathrm{Rec}(R_{1,1:2}^{(T)})$ of the first row of $R^{(T)}$.

Output: R — induces the MPUM of w_d.

Table 1. Modeling the Fibonacci numbers by Algorithm 1.

k	$\varepsilon^{(k)}$	$E^{(k)}$	$R^{(k)}$
1	0.0	$\begin{bmatrix} 1 & 0 \\ 0 & \xi \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & \xi \end{bmatrix}$
2	1.0	$\begin{bmatrix} 1 & -1 \\ \xi & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & -\xi \\ \xi & 0 \end{bmatrix}$
3	1.0	$\begin{bmatrix} 1 & -1 \\ \xi & 0 \end{bmatrix}$	$\begin{bmatrix} 1-\xi & -\xi \\ \xi & -\xi^2 \end{bmatrix}$
4	1.0	$\begin{bmatrix} 1 & -1 \\ \xi & 0 \end{bmatrix}$	$\begin{bmatrix} 1-2\xi & -\xi+\xi^2 \\ \xi-\xi^2 & -\xi^2 \end{bmatrix}$
5	−1.0	$\begin{bmatrix} 1 & 1 \\ -\xi & 0 \end{bmatrix}$	$\begin{bmatrix} 1-\xi-\xi^2 & -\xi \\ -\xi+2\xi^2 & \xi^2-\xi^3 \end{bmatrix}$
6	0.0	$\begin{bmatrix} 1 & 0 \\ 0 & \xi \end{bmatrix}$	$\begin{bmatrix} 1-\xi-\xi^2 & -\xi \\ -\xi^2+2\xi^3 & \xi^3-\xi^4 \end{bmatrix}$
7	0.0	$\begin{bmatrix} 1 & 0 \\ 0 & \xi \end{bmatrix}$	$\begin{bmatrix} 1-\xi-\xi^2 & -\xi \\ -\xi^3+2\xi^4 & \xi^4-\xi^5 \end{bmatrix}$
8	0.0	$\begin{bmatrix} 1 & 0 \\ 0 & \xi \end{bmatrix}$	$\begin{bmatrix} 1-\xi-\xi^2 & -\xi \\ -\xi^4+2\xi^5 & \xi^5-\xi^6 \end{bmatrix}$

A generalization of Algorithm 1 for the vector case is presented in (Kuijper, 1997). Next we show on a simulation example the application of this algorithm for computing the MPUM of a single input single output LTI system, *i.e.*, of a vector time series with $\mathrm{w} = 2$ components.

Example 2. (SISO system). Consider the second order SISO system

$$\mathscr{B} = \{ w \in (\mathbb{R}^2)^{\mathbb{N}} \mid w_1(t) + 0.85 w_1(t+1) + w_1(t+2) = w_2(t+1) + 0.5 w_2(t+1) + w_2(t+2) \},$$

or in a polynomial notation

$$\mathscr{B} = \ker\big(R(\sigma)\big), \quad \text{with}$$
$$R(\xi) = \begin{bmatrix} 1 & 0 \end{bmatrix} + \begin{bmatrix} 0.85 & -1 \end{bmatrix} \xi + \begin{bmatrix} 1 & -1 \end{bmatrix} \xi^2.$$

A random trajectory $w_\mathrm{d} \in (\mathbb{R}^2)^{10}$ of $\mathscr{B}$ is selected and used for modeling. Table 2 shows $R_{21}^{(k)}$ and $R_{22}^{(k)}$ per stage for the algorithm of (Kuijper, 1997), applied on w_d. For $k \geq 4$, $\begin{bmatrix} R_{21}^{(k)} & R_{22}^{(k)} \end{bmatrix}$ induces $\mathscr{B}_\mathrm{mpum}(w_\mathrm{d})$.

Table 2. Modeling a SISO system by the algorithm of (Kuijper, 1997).

k	$R_{21}^{(k)}$	$R_{22}^{(k)}$
1	1	-0.7093
2	$1 - 0.0788\xi$	-0.7093
3	$1 + 0.4796\xi$	-1.4344
4	$1 + 0.3259\xi$	$-1.4206 + 0.1014\xi$
5	$1 + 0.2414\xi + 0.8116\xi^2$	$-1.4206 + 0.1460\xi - 0.5625\xi^2$
6	$1 + 0.2332\xi + 0.8087\xi^2$	$-1.4129 + 0.1616\xi - 0.5724\xi^2$
7	$1 + 0.8500\xi + 1.0000\xi^2$	$-1.0000 - 0.5000\xi - 1.0000\xi^2$
8	$1 + 0.8500\xi + 1.0000\xi^2$	$-1.0000 - 0.5000\xi - 1.0000\xi^2$
9	$1 + 0.8500\xi + 1.0000\xi^2$	$-1.0000 - 0.5000\xi - 1.0000\xi^2$

3. LAG-RECURSIVE COMPUTATION OF THE MPUM

We return to infinite time series and consider recursive in the degrees of the laws (*i.e.*, in the lags) computation of the MPUM. The left kernel of the Hankel matrix

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

contains laws of $\mathscr{B}_\mathrm{mpum}(w_\mathrm{d})$. Under the assumption that $\mathscr{B}_\mathrm{mpum}(w_\mathrm{d})$ is controllable, all laws of $\mathscr{B}_\mathrm{mpum}(w_\mathrm{d})$ are contained in a basis of the subspace

$$\mathscr{N}_{L+1}(w_\mathrm{d}) := \mathrm{left\,ker}\big(\mathscr{H}_{L+1}(w_\mathrm{d})\big),$$

for $L \geq \mathbf{l}\big(\mathscr{B}_\mathrm{mpum}(w_\mathrm{d})\big)$.

Let the rows of R form a basis for $\mathscr{N}_{L+1}(w_\mathrm{d})$, *i.e.*,

$$R\mathscr{H}_{L+1}(w_\mathrm{d}) = 0, \quad \mathrm{rank}(R) = \dim\big(\mathscr{N}_{L+1}(w_\mathrm{d})\big).$$

Define a polynomial matrix $R(\xi)$ constructed from R,

$$R(\xi) = \sum_{i=0}^{L} R_i \xi^i \in \mathbb{R}^{g \times \mathrm{w}}[\xi],$$

where $R =: \begin{bmatrix} R_0 & R_1 & \cdots & R_l \end{bmatrix}$, $R_i \in \mathbb{R}^{g \times \mathrm{w}}$. Then

$$\ker\big(R(\sigma)\big) = \mathscr{B}_\mathrm{mpum}(w_\mathrm{d}).$$

The lag $\mathbf{l}\big(\mathscr{B}_\mathrm{mpum}(w_\mathrm{d})\big)$, however, is typically unknown in practice, so L is selected "sufficiently large" (*i.e.*, an upper bound for $\mathbf{l}\big(\mathscr{B}_\mathrm{mpum}(w_\mathrm{d})\big)$ is assumed known).

The computation of a basis R for $\mathscr{N}_{L+1}(w_{\mathrm{d}})$ is inefficient for large L. Suppose that for $l < L$, $\mathscr{H}_{l+1}(w_{\mathrm{d}})$ is rank deficient and let

$$\underbrace{\begin{bmatrix} r_0 & r_1 & \cdots & r_l \end{bmatrix}}_{r} \mathscr{H}_{l+1}(w_{\mathrm{d}}) = 0.$$

Then due to the Hankel structure

$$\begin{bmatrix} r_0 & r_1 & \cdots & r_l & & 0 \\ & r_0 & r_1 & \cdots & r_l & \\ & & \cdots & \cdots & & \cdots \\ 0 & & & r_0 & r_1 & \cdots & r_l \end{bmatrix} \mathscr{H}_{L+1}(w_{\mathrm{d}}) = 0.$$

Moreover, extra work has to be done in order to "extract" from the unstructured kernel R a short law r. In polynomial language $R(\xi)$ is not row reduced and an extra work is needed for its row reduction.

Assume that the k shortest laws $r^{(1)}, \ldots, r^{(k)} \in \mathbb{R}^{1\times \mathrm{w}}[\xi]$ have been identified, *i.e.*,

$$r^{(i)}(\sigma) w_{\mathrm{d}} = 0, \quad \text{for } i = 1, \ldots, k.$$

The question occurs of how to use this information in order to compute the remaining laws $r^{(k+1)}, \ldots, r^{(\mathrm{p})}$ of $\mathscr{B}_{\mathrm{mpum}}(w_{\mathrm{d}})$ efficiently. Assuming that the system induced by the matrix

$$R' := \begin{bmatrix} r^{(1)} \\ \vdots \\ r^{(k)} \end{bmatrix}$$

is controllable, this can be done as follows.

1. Compute D, such that $\left[\begin{smallmatrix} R' \\ D \end{smallmatrix}\right]$ is unimodular.
2. Define $e := D(\sigma) w_{\mathrm{d}}$.
3. Compute E, such that $\mathscr{B}_{\mathrm{mpum}}(e) = \ker\big(E(\sigma)\big)$.
4. Define $R := \left[\begin{smallmatrix} R' \\ ED \end{smallmatrix}\right]$.

We will show that $\mathscr{B}_{\mathrm{mpum}}(w_{\mathrm{d}}) = \ker\big(R(\sigma)\big)$.

Lemma 1. Assume that $R(\sigma)w = 0$ defines a controllable system $\mathscr{B}$. Let R' define a set of annihilators for $\mathscr{B}$, such that $R'(\sigma)w = 0$ also defines a controllable system $\mathscr{B}'$. Let D be such that $\left[\begin{smallmatrix} R' \\ D \end{smallmatrix}\right]$ is unimodular. Then there exists a unimodular matrix U and a matrix F, such that $UR = \left[\begin{smallmatrix} R' \\ FD \end{smallmatrix}\right]$.

PROOF. Denote the behavior of $D(\sigma)w = 0$ by $\mathscr{B}''$. Then

$$\mathscr{B}' \oplus \mathscr{B}'' = (\mathbb{R}^{\mathrm{w}})^{\mathbb{N}}.$$

Therefore

$$\mathscr{B} = (\mathscr{B}' \oplus \mathscr{B}'') \cap \mathscr{B} = \mathscr{B}' \oplus (\mathscr{B}'' \cap \mathscr{B})$$

is a decomposition of $\mathscr{B}$, leading to a kernel representation of the form $\left[\begin{smallmatrix} R' \\ FD \end{smallmatrix}\right](\sigma)w = 0$. □

Proposition 1. Let $w_{\mathrm{d}} \in (\mathbb{R}^{\mathrm{w}})^{\mathbb{N}}$ and assume that its MPUM $\mathscr{B}_{\mathrm{mpum}}(w_{\mathrm{d}})$ is controllable. Let R' be a set of annihilators of w_{d} such that $R'(\sigma)w = 0$ defines a controllable system. Let $D \in \mathbb{R}^{(\mathrm{w}-\mathrm{row\,dim}(R'))\times \mathrm{w}}[\xi]$ be such that $\left[\begin{smallmatrix} R' \\ D \end{smallmatrix}\right]$ is unimodular. Consider $e := D(\sigma)w_{\mathrm{d}}$ and assume that $E(\sigma)e = 0$ defines $\mathscr{B}_{\mathrm{mpum}}(e)$. Then

$$\begin{bmatrix} R' \\ ED \end{bmatrix}(\sigma)w = 0 \tag{1}$$

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

PROOF. Clearly (1) defines a system that is unfalsified by w_{d}. Let $R(\sigma)w = 0$ define $\mathscr{B}_{\mathrm{mpum}}(w_{\mathrm{d}})$. Then by Lemma 1, there exists a unimodular matrix U and a matrix F, such that

$$UR = \begin{bmatrix} R' \\ FD \end{bmatrix}.$$

Since FD annihilates w_{d}, F annihilates e. But since E is the MPUM for e, there exists H, such that

$$F = HE.$$

Therefore

$$\begin{bmatrix} R' \\ FD \end{bmatrix} = \begin{bmatrix} R' \\ HED \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & H \end{bmatrix} \begin{bmatrix} R' \\ ED \end{bmatrix}.$$

The behavior of $\left[\begin{smallmatrix} R' \\ FD \end{smallmatrix}\right](\sigma)w = 0$ is unfalsified by w_{d} and is included in the behavior of

$$\begin{bmatrix} R' \\ ED \end{bmatrix}(\sigma)w = 0.$$

Therefore $\left[\begin{smallmatrix} R' \\ ED \end{smallmatrix}\right](\sigma)w = 0$ defines $\mathscr{B}_{\mathrm{mpum}}(w_{\mathrm{d}})$. Note that H is a unimodular matrix. □

By "processing" the data w_{d} with D, we filter out the currently computed laws $r^{(1)}, \ldots, r^{(k)}$, and leave the remaining ones, which still have to be found out. Thus in computing $\mathscr{B}_{\mathrm{mpum}}(e)$ we do not find back $r^{(1)}, \ldots, r^{(k)}$ and their shifts. Applying recursively this idea leads to Algorithm 2.

If $\mathscr{H}_{l+1}(w_{\mathrm{d}})$ is full row rank for $l = 0, 1, 2, \ldots$, then w_{d} is an input, *i.e.*, $\mathscr{B}_{\mathrm{mpum}}(w_{\mathrm{d}}) = (\mathbb{R}^{\mathrm{w}})^{\mathbb{N}}$. In this case we define R to be the empty matrix [].

Computing a matrix D, such that $\left[\begin{smallmatrix} r \\ D \end{smallmatrix}\right]$ is unimodular is a Bézout-type problem. Assuming that the first two entries r_1 and r_2 of r are coprime, we have to solve the Bézout equation

$$r_1 b - r_2 a = 1,$$

for the polynomials a and b. Then we can choose

$$D = \begin{bmatrix} a & b & 0 \\ 0 & 0 & I_{\mathrm{w}-2} \end{bmatrix},$$

where $I_{\mathrm{w}-2}$ is the $(\mathrm{w}-2)\times(\mathrm{w}-2)$ identity matrix.

The matrix D exists if and only if r has no common factors in all elements. Equivalently, D exists if and only if the system $\ker\big(r(\sigma)\big)$ is controllable, see (Willems, 1997, Theorem 5). It can be shown that if $\mathscr{B}_{\mathrm{mpum}}(w_{\mathrm{d}})$ is controllable, then at each stage the model $\ker\big(r(\sigma)\big)$ is controllable. Therefore under the controllability assumption, the matrix D exists on each stage of the recursion. Applying Proposition 1 inductively we have the following theorem.

Algorithm 2 Lag-recursive MPUM computation `lag_recursive_mpum(`w_d`)`

Input: A time series $w_\mathrm{d} \in (\mathbb{R}^\mathtt{w})^\mathbb{N}$.

if $w_\mathrm{d} = 0$ **then**
  Let $R :=$ [] (*i.e.*, an empty matrix).
else
  Find the smallest $l \in \{0,1,2,\dots\}$, for which $\mathscr{H}_{l+1}(w_\mathrm{d})$ is rank deficient, or define $l := -1$ if there is no such l.
  if $l = -1$ **then**
    Let $R :=$ [] (*i.e.*, an empty matrix).
  else
    Compute a vector $r \in \mathbb{R}^{1\times(l+1)\mathtt{w}}$ in the left null space of $\mathscr{H}_{l+1}(w_\mathrm{d})$ and let $r(\xi) := \sum_{i=0}^{l} r_i \xi^i$, where $r =: \begin{bmatrix} r_0 & r_1 & \cdots & r_l \end{bmatrix}$ and $r_i \in \mathbb{R}^{1\times\mathtt{w}}$.
    Find $D \in \mathbb{R}^{(\mathtt{w}-1)\times\mathtt{w}}[\xi]$, such that $\left[\begin{smallmatrix} r \\ D \end{smallmatrix}\right]$ is unimodular.
    Compute $e := D(\sigma)w_\mathrm{d}$.
    $r' :=$ `lag_recursive_mpum(`e`)`.
    if $r' =$ [] **then**
      Let $R := r$.
    else
      Let $R := \begin{bmatrix} r \\ r'D \end{bmatrix}$.
    end if
  end if
end if

Output: R — induces the MPUM of w_d.

Theorem 2. Assume that $\mathscr{B}_\mathrm{mpum}(w_\mathrm{d})$ is controllable and let $R \in \mathbb{R}^{g\times\mathtt{w}}[\xi]$ be the result of applying Algorithm 2 on $w_\mathrm{d} \in (\mathbb{R}^\mathtt{w})^\mathbb{N}$. Then $\ker\big(R(\sigma)\big) = \mathscr{B}_\mathrm{mpum}(w_\mathrm{d})$.

Algorithm 2 applies a sequence of unimodular transformations on the time series w_d that makes its MPUM obvious. To show this, we index by superscript in brackets the stage of the recursion. Define $e^{(0)} := w_\mathrm{d}$. At stage $k = 0$, the algorithm computes an annihilator $r^{(0)}$ and a matrix $D^{(0)}$, such that $\left[\begin{smallmatrix} r^{(0)} \\ D^{(0)} \end{smallmatrix}\right]$ is unimodular and

$$\begin{bmatrix} r^{(0)} \\ D^{(0)} \end{bmatrix}(\sigma)e^{(0)} = \begin{bmatrix} 0 \\ e^{(1)} \end{bmatrix}. \tag{2}$$

At the kth stage of the recursion, the algorithm computes an annihilator $r^{(k)}$ and a matrix $D^{(k)}$, such that $\left[\begin{smallmatrix} r^{(k)} \\ D^{(k)} \end{smallmatrix}\right]$ is unimodular and

$$\begin{bmatrix} r^{(k)} \\ D^{(k)} \end{bmatrix}(\sigma)e^{(k)} = \begin{bmatrix} 0 \\ e^{(k+1)} \end{bmatrix},$$

or equivalently

$$\begin{bmatrix} I_k & 0 \\ 0 & r^{(k)} \\ 0 & D^{(k)} \end{bmatrix}(\sigma)\begin{bmatrix} 0_{k\times 1} \\ e^{(k)} \end{bmatrix} = \begin{bmatrix} 0_{(k+1)\times 1} \\ e^{(k+1)} \end{bmatrix}. \tag{3}$$

On stage k, w_d is transformed by pre multiplication with a unimodular matrix, so that the resulting signal $\left[\begin{smallmatrix} 0_{(k+1)\times 1} \\ e^{(k+1)} \end{smallmatrix}\right]$ has zeros in its first $k+1$ elements. The recursion stops at stage $k = \mathtt{p}$, when $r^{(\mathtt{p})} =$ [], *i.e.*, when $e^{(\mathtt{p})}$ is an input. (Under the assumption that $\mathscr{B}_\mathrm{mpum}(w_\mathrm{d})$ is controllable $\mathtt{p} < \mathtt{w}$.)

At stage $k = \mathtt{p}-1$,

$$\begin{bmatrix} I_{\mathtt{p}-1} & 0 \\ 0 & r^{(\mathtt{p}-1)} \end{bmatrix}(\sigma)\begin{bmatrix} 0_{(\mathtt{p}-1)\times 1} \\ e^{(\mathtt{p}-1)} \end{bmatrix} = 0_{\mathtt{p}\times 1}. \tag{4}$$

Substituting back (3) for $k = \mathtt{p}-2, \mathtt{p}-3, \dots, 1$ and (2) in (4), we have

$$\begin{bmatrix} I_{\mathtt{p}-1} & 0 \\ 0 & r^{(\mathtt{p}-1)} \end{bmatrix}\begin{bmatrix} I_{\mathtt{p}-2} & 0 \\ 0 & r^{(\mathtt{p}-2)} \\ 0 & D^{(\mathtt{p}-2)} \end{bmatrix}\cdots \begin{bmatrix} I_2 & 0 \\ 0 & r^{(2)} \\ 0 & D^{(2)} \end{bmatrix}\begin{bmatrix} I_1 & 0 \\ 0 & r^{(1)} \\ 0 & D^{(1)} \end{bmatrix}\begin{bmatrix} r^{(0)} \\ D^{(0)} \end{bmatrix}(\sigma)w_\mathrm{d} = 0,$$

which gives

$$\underbrace{\begin{bmatrix} r^{(0)} \\ r^{(1)}D^{(0)} \\ r^{(2)}D^{(1)}D^{(0)} \\ \vdots \\ r^{(\mathtt{p}-1)}D^{(\mathtt{p}-2)}\cdots D^{(0)} \end{bmatrix}}_{R}(\sigma)w_\mathrm{d} = 0.$$

Next we show a numerical example.

Example 3. Consider the 2 output 1 input system $\mathscr{B}$ induced by the polynomial matrix $R = \left[\begin{smallmatrix} r_1 \\ r_2 \end{smallmatrix}\right]$, given in (5). This system has two laws: one of degree 2 (the first row of R) and one of degree 1 (the second row of R).

Let w_d be a random trajectory of $\mathscr{B}$. For $l = 1$, the Hankel matrix $\mathscr{H}_{l+1}(w_\mathrm{d})$ has a 1-dimensional left kernel $\mathscr{N}_2(w_\mathrm{d})$, which is given by $r_1(\xi)$, see (6). $\dim\big(\mathscr{N}_3(w_\mathrm{d})\big) = 3$, and $\mathscr{N}_3(w_\mathrm{d})$ is given by $r_1(\xi)$, $\xi r_1(\xi)$, and r_2, see (7). In general,

$$\dim\big(\mathscr{N}_{L+1}(w_\mathrm{d})\big) = 2L-1, \quad \text{for } L \geq 1.$$

Note that if the lag $\mathbf{l}(\mathscr{B}) = 2$ is unknown and an upper bound, say, $L = 10$ is chosen, then a 19th dimensional subspace has to be computed, while the system has only two short laws.

Applied on w_d, Algorithm 2 returns the matrix $\hat{R}$, see (8), which is up to a pre-multiplication by a unimodular matrix equivalent to R, *i.e.*, $\mathscr{B} = \ker\big(\hat{R}(\sigma)\big)$.

4. CONCLUSIONS

We considered of exact recursive modeling of time series by LTI systems. The first problem reviewed in the paper—recursive in time modeling—is well covered in the system identification literature. We outlined an algorithms for scalar time series that is equivalent to the celebrated Berlekamp–Massey algorithm. The generalization of this algorithm for vector time series, presented in (Kuijper, 1997), was applied on a simulation example for modeling of a SISO system.

$$R(\xi) = \begin{bmatrix} -1.7379 & 1.3738 & -2.0313 \\ 0.3580 & -0.5071 & -0.2409 \end{bmatrix} \xi^0 + \begin{bmatrix} 5.9728 & -8.3627 & 12.2535 \\ 0.1680 & 0.6571 & 0.1987 \end{bmatrix} \xi^1 + \begin{bmatrix} 2.0008 & 7.8263 & -11.3336 \\ 0 & 0 & 0 \end{bmatrix} \xi^2. \tag{5}$$

$$\mathscr{N}_2(w_{\mathrm{d}}) = \operatorname{col\,span}\left(\begin{bmatrix} 0.3580 & -0.5071 & -0.2409 & 0.1680 & 0.6571 & 0.1987 \end{bmatrix}^\top \right). \tag{6}$$

$$\mathscr{N}_3(w_{\mathrm{d}}) = \operatorname{col\,span}\left(\begin{bmatrix} 0.3580 & -0.5071 & -0.2409 & 0.1680 & 0.6571 & 0.1987 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.3580 & -0.5071 & -0.2409 & 0.1680 & 0.6571 & 0.1987 \\ -1.7379 & 1.3738 & -2.0313 & 5.9728 & -8.3627 & 12.2535 & 2.0008 & 7.8263 & -11.3336 \end{bmatrix}^\top \right). \tag{7}$$

$$\hat{R}(\xi) = \begin{bmatrix} 0.3687 & -0.5222 & -0.2481 \\ 0.0089 & 0.0350 & 0.1342 \end{bmatrix} \xi^0 + \begin{bmatrix} 0.1730 & 0.6767 & 0.2046 \\ -0.0323 & -0.1264 & -0.7496 \end{bmatrix} \xi^1 + \begin{bmatrix} 0 & 0 & 0 \\ 0.0347 & 0.1359 & 0.6412 \end{bmatrix} \xi^2. \tag{8}$$

The second problem—recursive in lag modeling—is believed to be less familiar in the system identification community, although it is extensively treated by the numerical linear algebra community in the context of structured matrix computations. We presented a new algorithm that avoids the calculation of one and the same law of the system repeatedly. The main idea of the algorithm is the projection of the time series on the orthogonal complement of the currently found law. This is achieved by solving a Bézout equation.

ACKNOWLEDGEMENTS

Our research is supported by:

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

REFERENCES

Antoulas, A. and J. C. Willems (1993). A behavioral approach to linear exact modeling. *IEEE Trans. on Aut. Control* **38**(12), 1776–1802.

Heinig, G. and K. Rost (1984). *Algebraic methods for Toeplitz-like matricies and operators*. Birkhauser.

Heinig, G. and P. Jankowski (1992). Kernel structure of block Hankel and Toeplitz matricies and partial realization. *Lin. Alg. and Its Appl.* **175**, 1–30.

Kuijper, M. (1997). An algorithm for constructing a minimal partial realization in the multivariable case. *Systems & Control Letters* **31**(4), 225–233.

Kuijper, M. and J. C. Willems (1997). On constructing a shortest linear recurrence relation. *IEEE Trans. on Aut. Control* **42**(11), 1554–1558.

Willems, J. C. (1986*a*). From time series to linear system—Part I. Finite dimensional linear time invariant systems. *Automatica* **22**(5), 561–580.

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

Willems, J. C. (1987). From time series to linear system—Part III. Approximate modelling. *Automatica* **23**(1), 87–115.

Willems, J. C. (1997). On interconnections, control, and feedback. *IEEE Trans. on Aut. Control* **42**, 326–339.

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.