Data-driven simulation of nonlinear systems via linear time-invariant embedding

Ivan Markovsky

***Abstract*—Nonparameteric representations of linear time-invariant systems that use Hankel matrices constructed from data are the basis for data-driven signal processing and control methods of linear time-invariant systems. This paper extends the approach to data-driven simulation of a class of nonlinear systems. The key step of the generalization is an embedding result that is also of independent interest. The behavior of a polynomial time-invariant system is included into the behavior of a linear time-invariant system. A subclass of polynomial time-invariant systems that leads to a computationally tractable data-driven simulation problem is generalized bilinear. It includes Hammerstein, finite-lag Volterra, and biliner systems. The method proposed is illustrated and compared with the classical model-based method on simulation examples in the errors-invariables setup as well as benchmark real-life data sets.**

Keywords: behavioral approach, system identification, data-driven methods, nonlinear systems.

I. Introduction

Data-driven methods in signal processing and control avoid parametric model identification and subsequent model-based design. In the case of a linear time-invariant system, a nonparameteric representation—the image of a Hankel matrix constructed from the data—replaces the familiar state space and transfer function representations. A key result that gives theoretical justification for the nonparameteric representation based on Hankel matrices is the *fundamental lemma* of [1, Lemma 1]. The fundamental lemma is derived in the behavioral setting and gives sufficient conditions for the image of the Hankel matrix to coincide with the behavior of the data-generating system. Based on the fundamental lemma, data-driven simulation and control methods are proposed in [2], [3], [4], [5], [6], [7], [8]. The fundamental lemma and much of the subsequent results, however, assume that the data generating system is linear time-invariant.

There is significant interest in generalization of the fundamental lemma to nonlinear and time-varying systems. Generalizations for nonlinear systems were proposed in [9], [10], [11]. The classes of nonlinear systems considered are: finite-lag second-order Volterra [9], Hammerstein-Wiener [10], and bilinear [11]. They are special cases of the Nonlinear Auto-Regressive eXogenous (NARX) model class [12]. In this paper, we consider the data-driven simulation problem for polynomial NARX systems, called *polynomial time-invariant* [13]. The problem is equivalent to a structured low-rank matrix completion. Our main result is a generalization of the subspace data-driven simulation method of [2]. The most general subclass of polynomial time-invariant systems, for which the problem is tractable is called *generalized biliner*.

The key result allowing us to generalize the data-driven simulation method of [2] to nonlinear systems is *linear time-invariant embedding*: the behavior of the nonlinear system is included into the behavior of a linear time-invariant system. Linear time-invariant embedding is used for deriving realization and identification methods for Wiener systems [14]. In this paper, we generalize the result of [14] to the polynomial time-invariant model class and use it for data-driven simulation of general-

I. Markovsky is with the Department ELEC, Vrije Universiteit Brussel, 1050 Brussels, Belgium (e-mail: imarkovs@vub.be).

ized biliner systems, that include finite-lag Volterra, Hammerstein, and bilinear systems.

The paper is organized as follows. Section II reviews the data-driven simulation problem and its solution in the case of linear time-invariant systems. Section III introduces the model class of polynomial time-invariant systems. The main result of the paper—the generalization of data-driven simulation method for linear time-invariant systems to polynomial time-invariant systems is presented in Section IV. Simulation results illustrating the resulting method and showing its performance in case of noisy data and benchmark real-life data from the data base for system identification DAISY [17] are presented in Section V.

II. Data-driven simulation of linear time-invariant systems

Section II-A introduces notation and preliminary results for linear time-invariant systems. Section II-B reviews the data-driven simulation problem and the solution method of [2].

A. Notation and preliminaries

The set of q-variate discrete-time signals $w : \mathbb{N} \to \mathbb{R}^q$ is denoted by $(\mathbb{R}^q)^{\mathbb{N}}$. The *cut operator* $w|_L$ restricts w to the interval $[1, L]$, *i.e.*, $w|_L := \big(w(1), \dots, w(L)\big)$. With some abuse of notation, we view the finite L-samples long signal $w \in (\mathbb{R}^q)^L$ also as a qL-dimensional vector $w \in \mathbb{R}^{qL}$.

In the behavoral setting a discrete-time dynamical system $\mathscr{B}$ is defined as a set of trajectories, *i.e.*, $\mathscr{B} \subset (\mathbb{R}^q)^{\mathbb{N}}$. If the system $\mathscr{B}$ is linear, $\mathscr{B}$ is a subspace and if it is time-invariant $\sigma\mathscr{B} = \mathscr{B}$, where $(\sigma w)(t) := w(t+1)$ is the *shift operator*. The class of linear time-invariant systems with q variables is denoted by $\mathscr{L}^q$.

A system is often defined by equations, called *representations* of the system. A subclass of linear time-invariant system, called *finite dimensional*, admits a representation by a difference equation

$$\begin{aligned} \mathscr{B} &= \ker R(\sigma) \\ &:= \{\, w \mid R_0 w + R_1 \sigma w + \cdots + R_\ell \sigma^\ell w = 0 \,\}. \end{aligned} \tag{1}$$

The representation (1) is called a *kernel representation* of the system. A kernel representation is *minimal* if the number of equations, *i.e.*, the number of rows of R is as small as possible over all kernel representations of $\mathscr{B}$. In a minimal kernel representation, the degree of R is also minimal over all kernel representations of $\mathscr{B}$. The minimal degree is invariant of the representation and is called the *lag* $\mathbf{l}(\mathscr{B})$ of the system.

The q variables $w(t) \in \mathbb{R}^q$ of a system $\mathscr{B} \in \mathscr{L}^q$ can be partitioned into inputs $u(t) \in \mathbb{R}^m$ and outputs $y(t) \in \mathbb{R}^p$, *i.e.*, there is a permutation matrix $\Pi \in \mathbb{R}^{q\times q}$, such that $w(t) = \Pi \left[\begin{smallmatrix} u(t) \\ y(t) \end{smallmatrix}\right]$. The partitioning of the variables into inputs and outputs is in general not unique. The number of inputs $\mathbf{m}(\mathscr{B})$, however, is invariant of the choice of the partitioning and is therefore a property of the system $\mathscr{B}$.

An autonomous linear time-invariant system $\mathscr{B} \in \mathscr{L}^q$ is a system without inputs ($\mathbf{m}(\mathscr{B}) = 0$). The order $\mathbf{n}(\mathscr{B})$ of an autonomous linear time-invariant system $\mathscr{B} \in \mathscr{L}^q$ is equal to its dimension, $\dim \mathscr{B}$. More generally, for a system $\mathscr{B} \in \mathscr{L}^q$ with inputs, we have that

$$\dim \mathscr{B}|_L = \mathbf{m}(\mathscr{B})L + \mathbf{n}(\mathscr{B}), \quad \text{for } L \geq \mathbf{l}(\mathscr{B}),$$

i.e., the dimension of the restricted behavior $\mathscr{B}|_L$ is determined by the horizon's length L, the number of inputs $\mathbf{m}(\mathscr{B})$, and the order $\mathbf{n}(\mathscr{B})$ [15].

For a signal $w \in (\mathbb{R}^q)^T$, the sequential application of the shift operator and the cut operator with $L \leq T$ leads to a sequence of L-samples long sub-signals of w, $(\sigma^0 w)|_L, (\sigma^1 w)|_L, \dots, (\sigma^{T-L} w)|_L$. Stacking them next to each other in an matrix leads to the *Hankel matrix* with L block rows

$$\mathscr{H}_L(w) := \begin{bmatrix} w(1) & w(2) & \cdots & w(T-L+1) \\ w(2) & w(3) & \cdots & w(T-L+2) \\ w(3) & w(4) & \cdots & w(T-L+3) \\ \vdots & \vdots & & \vdots \\ w(L) & w(L+1) & \cdots & w(T) \end{bmatrix} \in \mathbb{R}^{qL \times T-L+1}.$$

A generalization of the Hankel matrix for a set of signals $w^1, \dots, w^N$ is the mosaic-Hankel matrix

$$\mathscr{H}_L(w^1, \dots, w^N) := \begin{bmatrix} \mathscr{H}_L(w^1) & \cdots & \mathscr{H}_L(w^N) \end{bmatrix}.$$

The Hankel matrix $\mathscr{H}_{L,j}(w)$ is the submatrix of $\mathscr{H}_L(w)$ consisting of the first j columns.

B. Problem statement and solution method

As shown in [2], initial conditions for a trajectory $w \in \mathscr{B}$ can be specified by a prefix trajectory $w_{\text{ini}} \in (\mathbb{R}^q)^{T_{\text{ini}}}$ of length $T_{\text{ini}} \geq \mathbf{l}(\mathscr{B})$. The condition that w is generated under the initial conditions specified by w_{ini} is that the concatenation $w_{\text{ini}} \wedge w$ of w_{ini} and w is a trajectory of $\mathscr{B}$. In the simulation problem, it is assumed that $\mathscr{B}$ has a given input/output partition. Without loss of generality, throughout the paper, we assume that $w := \left[\begin{smallmatrix} u \\ y \end{smallmatrix}\right]$ is such a partition.

Problem 1 (Data-driven simulation)**.** Given a model class $\mathscr{M}$, data $w_{\text{d}} \in (\mathbb{R}^q)^T$ generated by an unknown system $\mathscr{B} \in \mathscr{M}$, initial conditions $w_{\text{ini}} \in (\mathbb{R}^q)^\ell$, where $\ell \geq \mathbf{l}(\mathscr{B})$, and input $u_{\text{s}} \in (\mathbb{R}^{\mathtt{m}})^L$, find the output $y_{\text{s}} \in (\mathbb{R}^{\mathtt{p}})^L$ of $\mathscr{B}$ to u_{s} under the initial conditions w_{ini}, *i.e.*, $w_{\text{ini}} \wedge \left[\begin{smallmatrix} u_{\text{s}} \\ y_{\text{s}} \end{smallmatrix}\right] \in \mathscr{B}$.

Trajectory-based representation: Consider a linear time-invariant system $\mathscr{B} \in \mathscr{L}^q$. For any $w_{\text{d}} \in \mathscr{B}|_T$ and $L \in [1, T]$, we have that $\mathscr{B}|_L \subseteq \operatorname{image} \mathscr{H}_L(w_{\text{d}})$. If equality holds, *i.e.*,

$$\mathscr{B}|_L = \operatorname{image} \mathscr{H}_L(w_{\text{d}}), \tag{2}$$

we have a data-driven representation of the system. Indeed, (2) characterizes all L-samples long trajectories of the system using directly the given data w_{d} without derivation of a parametric model for the system. However, for (2) to hold true additional conditions have to be satisfied.

Theorem 2 (Corollary 19 [15])**.** *Consider $w_d \in \mathscr{B}|_T$, where $\mathscr{B} \in \mathscr{L}^q$ and $L \geq \mathbf{l}(\mathscr{B})$. Then, (2) holds if and only if*

$$\operatorname{rank} \mathscr{H}_L(w_d) = \mathbf{m}(\mathscr{B})L + \mathbf{n}(\mathscr{B}). \tag{3}$$

The practical significance of the data-driven representation (2) is evident by the following corollary of Theorem 2, which gives a system of linear equations that is equivalent to the constraint $w \in \mathscr{B}|_L$.

Corollary 3. *Let $w_d \in \mathscr{B}|_T$, where $\mathscr{B} \in \mathscr{L}^q$, $L \geq \mathbf{l}(\mathscr{B})$, and (3) holds. Then, for any $w \in \mathscr{B}|_L$ there is a vector $g \in \mathbb{R}^{T-L+1}$, such that*

$$w \in \mathscr{B}|_L \quad \iff \quad \text{there is } g \in \mathbb{R}^{T-L+1}, \text{ such that } w = \mathscr{H}_L(w_d)g. \tag{4}$$

The data-driven simulation method of [2]: For a linear time-invariant data generating system $\mathscr{B}$, *i.e.*, $\mathscr{M} = \mathscr{L}^q$, Problem 1 has a unique solution y_{s}. Using Corollary 3, the solution y_{s} can be given explicitly by a closed form expression in terms of the data $(w_{\text{d}}, w_{\text{ini}}, u_{\text{s}})$. This is result is the data-driven simulation method of [2].

Proposition 4. *Assume that (3) holds. Then, the solution to Problem 1 is given by*

$$y_s = \mathscr{H}_L(\sigma^\ell y_d) \begin{bmatrix} \mathscr{H}_{\ell,j}(w_d) \\ \mathscr{H}_L(\sigma^\ell u_d) \end{bmatrix}^+ \begin{bmatrix} w_{ini} \\ u_s \end{bmatrix}, \tag{5}$$

where

$$j =: T - \ell - L + 1 \tag{6}$$

and A^+ is the pseudo-inverse of A.

Proof. Proposition 4 is equivalent to [2, Proposition 1], however, here we use the necessary and sufficient condition (3) for the data-driven representation (2) instead of the sufficient conditions of [1, Lemma 1], that are used in [2]. □

III. Polynomial time-invariant model class

This section introduces the polynomial time-invariant model class. In its most general form a polynomial time-invariant system is defined as the kernel of a nonlinear operator. In the paper, we consider single-input single-output systems. We show that the constraint on a signal to be a trajectory of a polynomial time-invariant system can be expressed as rank deficiency of a structured matrix constructed from the data.

A. Nonlinear kernel and input/output representations

A discrete-time nonlinear time-invariant system $\mathscr{B}$ is often defined by a difference equation:

$$\mathscr{B} := \{\, w \mid R\big(w, \sigma w, \dots, \sigma^\ell w\big) = 0 \,\}, \tag{7}$$

where $R : \mathbb{R}^{q(\ell+1)} \to \mathbb{R}^g$ and g is the number of equations in the representation. We consider R's defined by a multivariable polynomial. The representation (7) is a generalization of the kernel representation (1) for linear time-invariant system.

In the paper, we consider the special case of (7) when $q=2$ and

$$R(w,\sigma w,\dots,\sigma^\ell w)=f\big(\mathbf{x}(w)\big)-\sigma^\ell y$$

with

$$\mathbf{x}(w):=\operatorname{vec}(w,\sigma w,\dots,\sigma^{\ell-1}w,\sigma^\ell u). \tag{8}$$

The model class considered is defined then by a nonlinear difference equation

$$\sigma^\ell y=f(x)=f(u,y,\sigma u,\sigma y,\dots,\sigma^{\ell-1}u,\sigma^{\ell-1}y,\sigma^\ell u). \tag{9}$$

In (9), the variable u can be chosen freely while the variable y is determined by u and the initial conditions $w_{\text{ini}}:=\big(w(-\ell+1),\dots,w(0)\big)$. Thus, u is an input, y is an output, and (9) is an *input/output representation* of the system. The vector $x(t)\in\mathbb{R}^{n_x}$, $n_x:=2\ell+1$ contains the variables at ℓ past samples and the input at the current moment of time.

Note 5 (Implementation of $\mathbf{x}: w\mapsto x$). Given a signal $w\in(\mathbb{R}^2)^T$ over a finite interval $[1,T]$, the signal $\mathbf{x}(w)$ is defined over the interval $[1,T-\ell]$. The map $\mathbf{x}: w\mapsto x$ is closely related to the Hankel matrix constructor:

$$\begin{bmatrix}x(1) & \cdots & x(T-\ell)\end{bmatrix}=\begin{bmatrix}\mathscr{H}_{\ell,T-\ell}(w)\\ \begin{bmatrix}u(\ell+1) & \cdots & u(T)\end{bmatrix}\end{bmatrix}.$$

In what follows, the function f is an n_x-variate polynomial. It can be parametrized as follows:

$$\begin{aligned}f(x)&=\theta_1\underbrace{x_1^{n_{11}}\cdots x_{n_x}^{n_{1n_x}}}_{\phi_1(x)}+\cdots+\theta_{n_\theta}\underbrace{x_1^{n_{n_\theta 1}}\cdots x_{n_x}^{n_{n_\theta n_x}}}_{\phi_{n_\theta}(x)}\\&=\theta^\top\phi(x).\end{aligned}$$

The vector of monomials ϕ defines the *model structure*. The model structure ϕ is specified by the degrees n_{ij}. This information is collected in an $n_\theta\times n_x$ matrix $\mathtt{N}=\big[n_{ij}\big]$. Once, the model structure is fixed, a particular model is specified by the parameter vector θ

$$\mathscr{B}(\theta):=\big\{\,w\mid\theta^\top\phi\big(\mathbf{x}(w)\big)=0\,\big\}. \tag{10}$$

The model class with structure ϕ defined by (10) is denoted by $\mathscr{P}_\phi$. The set of systems $\mathscr{B}\in\mathscr{P}_\phi$ for all polynomial structures ϕ is called *polynomial time-invariant* and is denoted by $\mathscr{P}$.

The input/output structure of (9) ensures existence and uniqueness of the solution of the simulation problem, *i.e.*, for $\mathscr{B}\in\mathscr{P}$, Problem 1 has a unique solution.

B. Special cases and examples

Separating the elements of ϕ into first order (linear) and other (nonlinear) terms, we have

$$f(x)=:\underbrace{\theta_{\text{lin}}^\top\phi_{\text{lin}}(x)}_{\text{linear part}}+\underbrace{\theta_{\text{nl}}^\top\phi_{\text{nl}}(x)}_{\text{nonlinear part}},$$

where $\theta_{\text{lin}}\in\mathbb{R}^{n_{\text{lin}}}$, and $\theta_{\text{nl}}\in\mathbb{R}^{n_{\text{nl}}}$. If $\phi_{\text{lin}}(x)=x$, we call the linear part *fully parameterized*.

If $\phi_{\text{nl}}=0$, the model $\mathscr{B}_\theta$ is linear time-invariant. Depending on the structure of ϕ_{nl}, we have the following classes of nonlinear systems.

- *Affine system:* $\phi_{\text{nl}}=1$.
- *Hammerstein system:* ϕ_{nl} is of the form
$$\begin{aligned}\phi_{\text{nl}}\big(x(t)\big)=\big[\phi_{\text{h}}\big(u(t)\big)\ \phi_{\text{h}}\big(u(t+1)\big)\ \cdots\\ \phi_{\text{h}}\big(u(t+\ell)\big)\big]^\top,\quad\text{for some }\phi_{\text{h}}:\mathbb{R}\to\mathbb{R}.\end{aligned}$$
- *Finite-lag Volterra systems:* ϕ_{nl} depends on the input u and its shifts only, *i.e.*, there is a function $\phi_{\text{v}}:\mathbb{R}^{\ell+1}\to\mathbb{R}$, such that
$$\phi_{\text{nl}}\big(x(t)\big)=\phi_{\text{v}}\big(x_u(t)\big),$$
where
$$x_u(t):=\operatorname{vec}\big(u(t),u(t+1),\dots,u(t+\ell)\big).$$
- *Bilinear systems:* ϕ_{nl} is linear in x_u as well as in y and its shifts, *i.e.*,
$$\phi_{\text{nl}}\big(x(t)\big)=x_u(t)\otimes x_y(t),$$
where
$$x_y(t):=\operatorname{vec}\big(y(t),y(t+1),\dots,y(t+\ell-1)\big)$$
and $\otimes$ is the Kroneker product.
- *Generalized bilinear systems:* ϕ_{nl} is affine in x_y, *i.e.*, there is $\phi_{\text{b},0}\in\mathbb{R}^{\mathbb{N}}$ and $\phi_{\text{b}}:\mathbb{R}^{\ell+1}\to\mathbb{R}$, such that
$$\phi_{\text{nl}}\big(x(t)\big)=\phi_{\text{b},0}(t)+\phi_{\text{b}}\big(x_u(t)\big)\otimes x_y(t). \tag{11}$$

Lemma 6 (Matrix representation of (11))**.** *For a generalized bilinear system, $\mathscr{B} \in \mathscr{P}_\phi$ and a finite signal $w \in (\mathbb{R}^2)^T$,*

$$\phi_{b,0} = \phi_{nl}\big(\mathbf{x}([\begin{smallmatrix}u\\0\end{smallmatrix}])\big) \in \mathbb{R}^{n_{nl}(T-\ell)}$$

and $\phi_b(x_u) \otimes x_y = \Phi_b y$, where

$$\Phi_b = \big[\phi_{nl}\big(\mathbf{x}([\begin{smallmatrix}u\\\delta\end{smallmatrix}])\big) - \phi_{b,0} \ \cdots \\ \phi_{nl}\big(\mathbf{x}(\big[\begin{smallmatrix}u\\\sigma^{T-1}\delta\end{smallmatrix}\big])\big) - \phi_{b,0}\big] \in \mathbb{R}^{n_{nl}(T-\ell)\times T},$$

and δ is the unit pulse $(1,0,\dots,0) \in \mathbb{R}^T$.

Proof. Follows directly from (11), *i.e.*, the fact that ϕ_{nl} is affine in x_y. □

Example 7 (Hammerstein system). As a simple example with lag $\ell = 1$, consider the Hammerstein system with quadratic input nonlinearity:

$$y(t+1) = 0.95y(t) + u(t+1) + 2u^2(t+1). \quad (12)$$

We have

$$x(t) = \begin{bmatrix} u(t) \\ y(t) \\ u(t+1) \end{bmatrix} \quad \text{and} \quad \phi\big(x(t)\big) = \begin{bmatrix} y(t) \\ u(t+1) \\ u^2(t+1) \end{bmatrix}.$$

The degrees matrix specifying the structure ϕ is

$$\mathtt{N} = \begin{array}{c} \begin{array}{ccc} x_1 & x_2 & x_3 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 2 \end{bmatrix} \end{array} \begin{array}{c} \\ \phi_1 \\ \phi_2 \\ \phi_3 \end{array}.$$

The parameter vector θ identifying the system $\mathscr{B}(\theta) \in \mathscr{P}_\phi$ defined by (12) is $\theta = \begin{bmatrix}0.95 & 1 & 2\end{bmatrix}^\top$.

Example 8 (Finite-lag Volterra system). Another example with lag $\ell = 2$ is the Volterra system involving a non-static nonlinearity of the input

$$y(t+2) = -p_0 y(t) - p_1 y(t+1) + q_0 u(t) + \\ q_1 u(t+1) + q_2 u(t+2) + 0.1u^2(t+2) + \\ 0.1u(t+1)u(t+2) + 0.1u(t)u(t+1)u(t+2). \quad (13)$$

In this case,

$$x(t) = \operatorname{col}\big(u(t), y(t), u(t+1), y(t+1), u(t+2)\big)$$

and

$$\phi\big(x(t)\big) = \begin{bmatrix} x(t) \\ \phi_{\mathrm{nl}}\big(x(t)\big) \end{bmatrix},$$

i.e., the "linear part" of (13) is fully parametrized and the nonlinear part

$$\phi_{\mathrm{nl}}\big(x(t)\big) = \begin{bmatrix} \big(u^2(t+2) \\ u(t+1)u(t+2) \\ u(t)u(t+1)u(t+2) \end{bmatrix}$$

involves three terms—two of degree 2 and one of degree 3. This structure is specified by the degrees matrix

$$\mathtt{N} = \begin{bmatrix} \mathtt{N}_{\mathrm{l}} \\ \mathtt{N}_{\mathrm{n}} \end{bmatrix}, \quad \text{where } \mathtt{N}_{\mathrm{l}} = I_5$$

and

$$\mathtt{N}_{\mathrm{n}} = \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ \phi_1 \\ \phi_2 \\ \phi_3 \end{array}.$$

The parameter vector identifying the system defined by (13) is

$$\theta = \begin{bmatrix} \theta_{\mathrm{l}} \\ \theta_{\mathrm{n}} \end{bmatrix}, \quad \text{where } \theta_{\mathrm{l}} = \begin{bmatrix}-p_0 & -p_1 & q_0 & q_1 & q_2\end{bmatrix}^\top$$

and

$$\theta_{\mathrm{n}} = \begin{bmatrix}0.1 & 0.1 & 0.1\end{bmatrix}^\top.$$

Example 9 (Generalized bilinear system). As a generalization of the finite-lag Volterra system in Example 8, we consider the system defined by the difference equation

$$y(t+2) = -p_0 y(t) - p_1 y(t+1) + q_0 u(t) + \\ q_1 u(t+1) + q_2 u(t+2) + 0.1y(t)u^2(t+2) + \\ 0.1u(t+1)y(t+1)u(t+2) + \\ 0.1u(t)u(t+1)u(t+2). \quad (14)$$

The $x(t)$ vector is the same as in Example 8. The nonlinear part of (14) consists of three terms of degree 3

$$\phi_{\mathrm{nl}}\big(x(t)\big) = \begin{bmatrix} u^2(t+2) \\ u(t+1)y(t+1)u(t+2) \\ u(t)u(t+1)u(t+2) \end{bmatrix}.$$

Note that two of them involve both input and output variables, so that the system is not of the finite-lag Volterra type. The structure is specified by the degrees matrix

$$\mathtt{N} = \begin{bmatrix} \mathtt{N}_{\mathrm{l}} \\ \mathtt{N}_{\mathrm{n}} \end{bmatrix}, \quad \text{where } \mathtt{N}_{\mathrm{l}} = I_5$$

and

$$\mathtt{N}_\mathrm{n} = \begin{array}{c} \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ \phi_1 \\ \phi_2 \\ \phi_3 \end{array}.$$

The parameter vector identifying the system defined by (14) is the same as in Example 8.

C. Rank condition for $w \in \mathscr{B}(\theta) \in \mathscr{P}_\phi$

From (7) and (9), we have

$$R(w, \sigma w, \dots, \sigma^\ell w) = \begin{bmatrix} \theta^\top & -1 \end{bmatrix} \begin{bmatrix} \phi(x) \\ \sigma^\ell y \end{bmatrix}.$$

Therefore, $w \in (\mathbb{R}^2)^T$ is a trajectory of a system $\mathscr{B}(\theta) \in \mathscr{P}_\phi$ if and only if

$$\begin{bmatrix} \theta^\top & -1 \end{bmatrix} \mathscr{S}(w) = 0, \tag{15}$$

where

$$\mathscr{S}(w) := \begin{bmatrix} \phi\big(x(1)\big) & \cdots & \phi\big(x(T-\ell)\big) \\ y(\ell+1) & \cdots & y(T) \end{bmatrix}. \tag{16}$$

In general, the matrix $\mathscr{S}(w)$ is nonlinearly structured. In the linear time-invariant case, with fully parameterized model, *i.e.*, $\phi(x) = x$, the structure is Hankel $\mathscr{S} = \mathscr{H}_{\ell+1}$.

Checking if a signal $w \in (\mathbb{R}^2)^T$ is a trajectory of a given system $\mathscr{B}(\theta) \in \mathscr{P}_\phi$ can be done using (15):

$$w \in \mathscr{B}(\theta)|_T \quad \iff \quad \begin{bmatrix} \theta^\top & -1 \end{bmatrix} \mathscr{S}(w) = 0.$$

More generally, checking if a signal $w \in (\mathbb{R}^2)^T$ is a trajectory of an *unknown* polynomial time-invariant system in a *given model class* $\mathscr{P}_\phi$ can be done by a rank test: $\operatorname{rank} \mathscr{S}(w) \leq n_\theta + 1$.

The generalization of (15) to multiple trajectories $w^1, \dots, w^N$ is straightforward:

$$\begin{bmatrix} \theta^\top & -1 \end{bmatrix} \underbrace{\begin{bmatrix} \mathscr{S}(w^1) & \cdots & \mathscr{S}(w^N) \end{bmatrix}}_{\mathscr{S}(w^1,\dots,w^N)} = 0. \tag{17}$$

In the case of linear time-invariant system, with fully parameterized model, $\mathscr{S}(w^1, \dots, w^N)$ is the mosaic-Hankel matrix $\mathscr{H}_{\ell+1}(w^1, \dots, w^N)$, see [15].

IV. Data-driven simulation of polynomial time-invariant systems

Since both w_d and w_s are trajectories of $\mathscr{B}(\theta)$, by (17) we have that

$$\operatorname{rank} \begin{bmatrix} \mathscr{S}(w_\mathrm{d}) & \mathscr{S}(w_\mathrm{s}) \end{bmatrix} \leq n_\theta. \tag{18}$$

The data-driven simulation problem for polynomial time-invariant systems can be restated then as a matrix completion problem:

find y_s, so that (18) holds.

The problem can be solved by the following divide-and-conquer strategy:

1) *system identification:* compute a basis vector $\widehat{\theta}$ for the left kernel of $\mathscr{S}(w_\mathrm{d})$, and
2) *model-based simulation:* using $\widehat{\theta}$, compute y_s from (9).

For well-posedness of the identification problem on step 1 the left kernel of $\mathscr{S}(w_\mathrm{d})$ has to be one-dimensional. This is an identifiability condition that is also a sufficient condition for well-posedness of the data-driven simulation problem.

Theorem 10 (Identifiability condition)**.** *Consider a polynomial time-invariant system $\mathscr{B}(\theta) \in \mathscr{P}_\phi$ and a trajectory $w_d \in \mathscr{B}(\theta)|_T$. Assume that $\operatorname{rank} \mathscr{S}(w_d) = n_\theta$ and let $\widehat{\theta}$ be a basis vector for the left kernel of $\mathscr{S}(w_d)$. Then, $\mathscr{B}(\widehat{\theta}) = \mathscr{B}(\theta)$,* i.e.*, $\widehat{\theta}$ identifies the data generating system $\mathscr{B}(\theta)$.*

Proof. By the assumption that $\operatorname{rank} \mathscr{S}(w_\mathrm{d}) = n_\theta$, the left kernel of $\mathscr{S}(w_\mathrm{d})$ has dimension equal to one. Then, there is $\alpha \neq 0$, such that $\widehat{\theta} = \alpha\theta$, and therefore $\mathscr{B}(\widehat{\theta}) = \mathscr{B}(\theta)$. □

The rank condition, $\operatorname{rank} \mathscr{S}(w_\mathrm{d}) = n_\theta$, is a *persistency of excitation* type condition. In the case of a linear time-invariant system, Theorem 10 is a special case of the identifiability result of [15].

As an alternative to the *model-based simulation approach* outlined above, next, we propose a data-driven approach that is a generalization of the method of [2].

A. Linear time-invariant embedding

A key idea exploited next is to embed the nonlinear polynomial time-invariant system $\mathscr{B} = \mathscr{B}(\theta) \in$

$\mathscr{P}_\phi$ into a linear time-invariant system $\mathscr{B}_{\text{ext}}$ by replacing the vector $\phi_{\text{nl}}\big(\mathbf{x}(w)\big)$ with an additional input variable $u_{\text{nl}} \in (\mathbb{R}^{n_{\text{nl}}})^{\mathbb{N}}$, where $n_{\text{nl}} := \dim \phi_{\text{nl}}(x)$. As a result the embedding system

$$\mathscr{B}_{\text{ext}} := \big\{\, w_{\text{ext}} = \left[\begin{smallmatrix} w \\ u_{\text{nl}} \end{smallmatrix}\right] \;\big|\; \sigma^\ell y = \theta_{\text{lin}}^\top \phi_{\text{lin}}\big(\mathbf{x}(w)\big) + \theta_{\text{nl}}^\top u_{\text{nl}} \,\big\} \quad (19)$$

has $1+n_{\text{nl}}$ inputs. Let Π_w be the projection of the extended trajectory w_{ext} onto the w variables and let $\Pi_{u_{\text{nl}}}$ be the projection of w_{ext} onto the u_{nl} variable.

Lemma 11 (Linear time-invariant embedding of a polynomial time-invariant system)**.** *The polynomial time-invariant system $\mathscr{B} \in \mathscr{P}_\theta$ is embedded in the linear time-invariant system $\mathscr{B}_{ext}$ defined in (19),* i.e.*, $\mathscr{B} \subseteq \Pi_w \mathscr{B}_{ext}$. Moreover, the embedding is exact by imposing the constraint $u_{nl} = \phi_{nl}\big(\mathbf{x}(w)\big)$:*

$$\mathscr{B} = \big\{\, \Pi_w w_{ext} \mid w_{ext} \in \mathscr{B}_{ext} \text{ and } \Pi_{u_{nl}} w_{ext} = \phi_{nl}\big(\mathbf{x}(\Pi_w w_{ext})\big) \,\big\}.$$

Proof. Let $w \in \mathscr{B}$. Then, $\sigma^\ell y = \theta_{\text{lin}}^\top \phi_{\text{lin}}\big(\mathbf{x}(w)\big) + \theta_{\text{nl}}^\top u_{\text{nl}}$, with $u_{\text{nl}} = \phi_{\text{nl}}\big(\mathbf{x}(w)\big)$. It follows by the definition of (19) that $w \in \Pi_w \mathscr{B}_{\text{ext}}$. Therefore, $\mathscr{B} \subseteq \Pi_w \mathscr{B}_{\text{ext}}$. For w, such that $w \in \Pi_w \mathscr{B}_{\text{ext}}$, in general, it is not true that $w \in \mathscr{B}$. Indeed, the constraint

$$u_{\text{nl}} := \Pi_{u_{\text{nl}}} w_{\text{ext}} = \phi_{\text{nl}}\big(\mathbf{x}(\left[\begin{smallmatrix} u_{\text{ini}} \\ u_{\text{s}} \end{smallmatrix}\right])\big) \quad (20)$$

may not be satisfied. Imposing this extra constraint (in addition to $w \in \Pi_w \mathscr{B}_{\text{ext}}$) however ensures that $w \in \mathscr{B}$. □

Example 12. The linear time-invariant relaxation $\mathscr{B}_{\text{ext}}$ of the Hammerstein system $\mathscr{B}$ in example 7 is defined by

$$y(t+1) = 0.95y(t) + u_1(t+1) + 2u_2(t).$$

$\mathscr{B}_{\text{ext}}|_T$ is a subspace of $\mathbb{R}^{3T}$ with dimension $\dim \mathscr{B}_{\text{ext}}|_T = 2T+2$, while $\mathscr{B}$ is a manifold of $\mathbb{R}^{2T}$ with dimension $\dim \mathscr{B}|_T = T+2$.

B. Generalization of (5)

Lemma 11 allows us to apply methods developed for linear time-invariant systems to polynomial time-invariant systems. In this section, we consider the data-driven simulation problem and generalize the method of [2] to polynomial time-invariant systems, *i.e.*, in Problem 1, $\mathscr{M} = \mathscr{P}_\phi$ and the structure specification ϕ is given.

Let $\mathscr{B}_{\text{ext}}$ be the linear time-invariant embedding of $\mathscr{B} \in \mathscr{P}_\phi$ and define the "extended" signals

$$w_{\text{d,ext}}(t) := \begin{bmatrix} w_{\text{d}}(t) \\ \phi_{\text{nl}}\big(\mathbf{x}\big(w_{\text{d}}(t)\big)\big) \end{bmatrix}$$

and

$$w_{\text{ext}}(t) := \begin{bmatrix} w(t) \\ \phi_{\text{nl}}\big(\mathbf{x}\big(w(t)\big)\big) \end{bmatrix}.$$

As in Section II-B, the signal w_{ext} has length $\ell+L$ and is a concatenation of initial conditions $w_{\text{ini,ext}}$ of length ℓ and a to-be-simulated signal $w_{\text{s,ext}}$ of length L. The data-driven simulation method (5) applied to $\mathscr{B}_{\text{ext}}$ gives us

$$y_{\text{s,ext}} = \mathscr{H}_L(\sigma^\ell y_{\text{d}}) \mathscr{A}^+(w_{\text{d}}) \begin{bmatrix} w_{\text{ini,ext}} \\ u_{\text{s,ext}} \end{bmatrix}, \quad (21)$$

where

$$\mathscr{A}(w_{\text{d}}) := \begin{bmatrix} \mathscr{H}_{\ell,j}(w_{\text{d,ext}}) \\ \mathscr{H}_L(\sigma^\ell u_{\text{d,ext}}) \end{bmatrix}$$

and j is defined in (6). If (20) holds true, we have that by Lemma 11 $\mathscr{B} = \mathscr{B}_{\text{ext}}$ and therefore $y_{\text{s,ext}}$ defined in (21) is the response of $\mathscr{B}$ to input u_{s} and initial conditions w_{ini}. Then, (21) gives us the solution to the data-driven simulation problem.

The extended signal $w_{\text{d,ext}}$ is constructed from the given data w_{d} and the structure specification ϕ. In general, however, u_{ext} can not be constructed from the given initial condition w_{ini}, simulation input u_{s}, and ϕ. Indeed, the unknown y_{s} may be needed for the construction of u_{ext}. This problem does not appear in the special case of a finite-lag Volterra model structure ϕ, where u_{nl} depends on u_{s} only.

The following theorem shows that (21) can be used also for data-driven simulation of bilinear time-invariant systems, where u_{nl} depends on both u_{s} and y_{s}, however, the dependence on y_{s} is affine.

Theorem 13 (Data-driven simulation of generalized bilinear time-invariant systems)**.** *Consider a generalized bilinear system $\mathscr{B} \in \mathscr{P}_\phi$ and data $w_d \in \mathscr{B}|_T$. Let $\mathscr{B}_{ext}$ be the linear time-invariant embedding (19) of $\mathscr{B}$. Assume that*

$$\operatorname{rank} \mathscr{H}_L(w_{d,ext}) = (1+n_{nl})L + \ell \quad (22)$$

holds. Then, the solution to Problem 1 is given by

$$y_s := \mathscr{H}_L(\sigma^\ell y_d)\mathscr{A}^+(w_d)\begin{bmatrix} w_{ini} \\ u_s \\ \Phi_{b,p}y_{ini} + \phi_{b,0} \end{bmatrix}, \quad (23)$$

where

$$\mathscr{A}(w_d) := \begin{bmatrix} \mathscr{H}_{\ell,j}(w_d) \\ \mathscr{H}_L(\sigma^\ell u_d) \\ \mathscr{H}_L\big(\phi_{nl}\big(\mathbf{x}(w_d)\big)\big) - \Phi_{b,f}\mathscr{H}_L(\sigma^\ell y_d) \end{bmatrix},$$

with $\begin{bmatrix}\Phi_{b,p} & \Phi_{b,f}\end{bmatrix} := \Phi_b$, *where* $\Phi_{b,p} \in \mathbb{R}^{n_{nl}T\times\ell}$ *and* $\Phi_{b,f} \in \mathbb{R}^{n_{nl}T\times L}$.

Proof. By (22), using Corollary 3, we have that there is g such that

$$\mathscr{H}_{\ell+L}(w_{\text{d,ext}})g = w_{\text{ext}}.$$

Partitioning the equations according to the partitioning $\text{col}(u,y,u_{\text{nl}})$ of w_{ext}, we have

$$\begin{bmatrix} \mathscr{H}_{\ell+L}(u_\text{d}) \\ \mathscr{H}_{\ell+L}(y_\text{d}) \\ \mathscr{H}_{\ell+L}(u_{\text{d,nl}}) \end{bmatrix} g = \begin{bmatrix} \left[\begin{smallmatrix} u_\text{ini} \\ u_\text{s}\end{smallmatrix}\right] \\ \left[\begin{smallmatrix} y_\text{ini} \\ y_\text{s}\end{smallmatrix}\right] \\ u_\text{nl} \end{bmatrix}. \quad (24)$$

The equation

$$y_\text{s} = \mathscr{H}_L(\sigma^\ell y_\text{d})g \quad (25)$$

is a definition of y_s and imposes no constraints on g.

In order for the embedding $\mathscr{B}_\text{ext}$ to satisfy $\Pi_w\mathscr{B}_\text{ext} = \mathscr{B}$, the constraint (20) has to be added to (24). Since $\mathscr{B}$ is a generalized bilinear system, by Lemma 6, (20) can be written as

$$u_\text{nl} = \phi_{\text{b},0} + \Phi_\text{b}\left[\begin{smallmatrix} y_\text{ini} \\ y_\text{s}\end{smallmatrix}\right] = \phi_{\text{b},0} + \Phi_{\text{b,p}}y_\text{ini} + \Phi_{\text{b,f}}y_\text{s}.$$

Substituting back in (24) and using (25), we obtain

$$\mathscr{A}(w_\text{d})g = \begin{bmatrix} w_\text{ini} \\ u_\text{s} \\ \phi_{\text{b},0} + \Phi_{\text{b,p}}y_\text{ini} \end{bmatrix}.$$

□

Note 14 (Persistency of excitation assumption). The key assumption of Theorem 13 is the persistency of excitation assumption (22). The matrix $\mathscr{H}_L(w_{\text{d,ext}})$ is of dimension $(2+n_\text{nl})L \times (T-\ell-L+1)$, where n_nl is the dimension of ϕ_nl, *i.e.*, the number of nonlinear terms in the representation (9) of the system. A necessary condition for (22) to hold true is that

$$T \geq (3+n_\text{nl})L + \ell - 1.$$

Note 15 (Comparison with model-based simulation). In comparison with the identifiability condition $\text{rank}\,\mathscr{S}(w_\text{d}) = n_\theta$ of Theorem 10, the condition (22) for data-driven simulation is stronger. This is true also in the case of linear time-invariant systems. As shown in [16, Lemma 3], however the condition for data-driven simulation can be relaxed by "weaving together segments of the desired response". Such a relaxation can be applied also in the nonlinear case.

Note 16 (Explicit characterization of the behavior of a generalized bilinear system). Theorem 13 shows that the restricted behavior $\mathscr{B}|_L$ of a generalized bilinear system $\mathscr{B}$ is the image of the map $M : \mathbb{R}^{2\ell+L} \mapsto \mathbb{R}^L$ defined by

$$M(w_\text{ini}, u_\text{s}) := \mathscr{H}_L(\sigma^\ell y_\text{d})\mathscr{A}^+(w_\text{d})\begin{bmatrix} w_\text{ini} \\ u_\text{s} \\ \Phi_{\text{b,p}}y_\text{ini} + \phi_{\text{b},0} \end{bmatrix}.$$

Corollary 17 (Data-driven simulation of finite-lag Volterra systems)**.** *Consider a lag-ℓ Volterra system $\mathscr{B} \in \mathscr{P}_\phi$ and data $w_d \in \mathscr{B}|_T$. Let $\mathscr{B}_{ext}$ be the linear time-invariant embedding (19) of $\mathscr{B}$. Assume that (22) holds. Then, the solution to Problem 1 is*

$$y_s = \mathscr{H}_L(\sigma^\ell y_d)\begin{bmatrix} \mathscr{H}_{\ell,j}(w_d) \\ \mathscr{H}_L(\sigma^\ell u_d) \\ \mathscr{H}_L\big(\phi_v\big(\mathbf{x}(u_d)\big)\big) \end{bmatrix}^+ \begin{bmatrix} w_{ini} \\ u_s \\ \phi_v\big(\mathbf{x}(\left[\begin{smallmatrix} u_{ini} \\ u_s\end{smallmatrix}\right])\big) \end{bmatrix}.$$

Theorem 13 gives a constructive method for data-driven simulation of generalized bilinear systems, in particular, affine, Hammerstein, finite-lag Volterra, and bilinear systems. In the following section, we show simulation results illustrating the method.

V. Simulation results

In this section, we compare empirically the data-driven simulation method with the model-based and show the performance of the methods in case of simulated noisy data as well as real-life data from the data base for system identification DAISY [17]. The simulation results are made reproducible in the sense of [18] by providing the implementation of the method and the data generating scripts. The computational environment used is MATLAB. The files reproducing the simulation results are available from http://homepages.vub.ac.be/~imarkovs/software/ddsim.tar. The code is presented in a literate programming style [19], [20] http://homepages.vub.ac.be/~imarkovs/software/ddsim.pdf.

A. Illustrative examples

In order to test the method in as diverse situations as possible, we consider the three nonlinear systems from Section III-B—Hammerstein, finite-lag Volterra, and generalized biliner—and do data-driven simulation of

- S1: response to random initial conditions and random input,
- S2: free response—nonzero initial conditions and zero input, and
- S3: step response—zero initial conditions and unit step input.

The data w_{d} is a $T = 100$ samples long trajectory obtained from the system with random initial conditions and random input. The simulation horizon is $L = 10$ samples.

The result $\widehat{y}_{\mathrm{s}}$ obtained by the data-driven simulation method (23) is evaluated by the relative percentage error

$$e = 100\frac{\|y_{\mathrm{s}} - \widehat{y}_{\mathrm{s}}\|}{\|y_{\mathrm{s}}\|}, \tag{26}$$

where y_{s} is the response obtained by model-based simulation. In all experiments the relative percentage error is of the order of the machine precision ($e < 10^{-13}$). This is an empirical confirmation of the correctness of the method and its implementation.

B. Performance in case of noisy data w_d

In this section, the data w_{d} is generated in the *errors-in-variables* setup [21]:

$$w_{\mathrm{d}} = \overline{w}_{\mathrm{d}} + \widetilde{w}_{\mathrm{d}}, \quad \text{where } \overline{w}_{\mathrm{d}} \in \mathscr{B}|_T \quad \text{and } \widetilde{w}_{\mathrm{d}} \sim \mathrm{N}(0, s^2 I). \tag{27}$$

The *true value* $\overline{w}_{\mathrm{d}}$ is a trajectory of the to-be-simulated system $\mathscr{B} \in \mathscr{P}_\phi$ and the *measurement noise* $\widetilde{w}_{\mathrm{d}}$ is zero mean Gaussian with covariance matrix $s^2 I$. In the simulation example, the noise standard deviation s is selected so that the noise to signal ratio is varied in the interval of $[0, 0.05]$, *i.e.*, starting from exact data and ranging up to 5% noise. The to-be-simulated system $\mathscr{B} \in \mathscr{P}_\phi$ is the finite-lag Volterra model of example (13). The trajectory w_{d} has $T = 1000$ samples and w_{s} has $L = 10$ samples. The simulated response $\widehat{y}_{\mathrm{s}}$, computed by the data-driven simulation algorithm, is evaluated by the relative percentage error (26), where y_{s} is computed by model-based simulation using the true model $\mathscr{B}$.

Figure 1 shows the relative percentage error as a function of the noise to signal ratio. The model-based method is uniformly better than the data-driven method. We attribute this to the fact that the model-based method uses more effectively the data by shaping it in the matrix $\mathscr{S}(w_{\mathrm{d}})$ rather than the matrix in (24). This is related also to the weaker persistency of exaction for identification compared to the one for data-driven simulation, see Note 14.

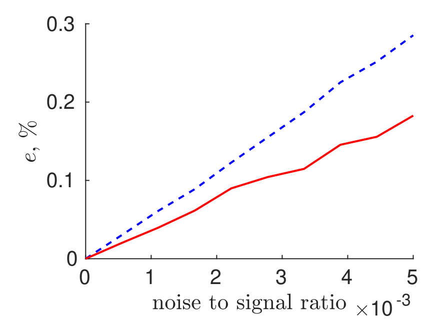

Fig. 1. Relative percentage error e as a function of the noise-to-signal ratio: solid line—model-based method, dashed line—data-driven method.

C. Performance on data from the DAISY dataset

In this section we apply the data-driven simulation method on data from the data-base for system identification DAISY [17]. From the 24 dataset in DAISY we choose the 6 single-input single-output ones (see Table I). The given data (a single input/output experiment) is split into two parts: the first 80% is used as the "data trajectory" w_{d}, the remaining 20% as the "to-be-simulated trajectory" $w_{\text{ini}} \wedge w_{\text{s}}$. In all experiments, the model class $\mathscr{M}$ is the set of affine models with lag ℓ, where the value of ℓ is taken from [22].

The computed responses $\widehat{y}_{\text{s}}$ by the data-driven simulation method and the model-based method are compared with the measured response y_{s} in terms of the relative percentage estimation error (26). The results shown in Table I indicate that the data-driven method is uniformly better than the model-based method. We attribute this to the fact that in this case the model class is not correct, which leads to a potentially larger bias error in the parameter estimation step of the model-based method. The nonparameteric data-driven representation and the resulting lack of parameter estimation step in the data-driven method is an advantage in this case.

TABLE I
RESULTS WITH DATA FROM THE DAISY DATA BASE: e_{MB}—RELATIVE PERCENTAGE ESTIMATION ERROR FOR THE MODEL-BASED METHOD, e_{DD}—RELATIVE PERCENTAGE ESTIMATION ERROR FOR THE DATA-DRIVEN METHOD. ("FAIL" INDICATES RELATIVE ERROR ABOVE 100%)

	data set name	T_{d}	T_{s}	ℓ	e_{mb}	e_{dd}
1	Hair dryer	800	200	5	2.7	2.3
2	Ball and beam	800	200	2	fail	44.0
3	Wing flutter	800	200	5	fail	fail
4	Robot arm	800	200	4	23.0	3.6
5	Heating system	600	200	2	9.9	5.6
6	Steam exchanger	3200	800	2	0.7	0.5

VI. CONCLUSIONS

We considered data-driven simulation of polynomial time-invariant systems (10) and showed that the problem is equivalent to matrix completion. The classical solution method computes first the parameter vector θ. Then, using θ, the classical method computes the desired response from the difference equation (9). In contrast, the data-driven solution proposed in the paper computes directly the desired response without obtaining the model parameter θ.

The method proposed is a generalization of the method of [2]. The key technical result for the generalization is linear time-invariant embedding: the behavior of the polynomial time-invariant system is included into to behavior of a linear time-invariant system. Under an addition nonlinear constraint the behavior of the embedding system coincides with the behavior of original nonlinear system. The embedding result reveals that the generalization of the linear time-invariant data-driven simulation method to finite-lag Volterra systems is trivial however it suggests also a novel extension to a class of systems called generalized bilinear. The resulting method requires only a solution of a system of linear equations (a pseudo-inverse computation).

Simulation results show the performance of the method in case of simulated noisy data as well as real-life benchmark problems. The empirical study suggests that in case of simulated noisy data from a true polynomial time-invariant system, using the correct model structure model-based simulation gives better results than data-driven simulation. In case of true data where the correct model structure does not exist or is unknown however the data-driven simulation method has an advantage.

Statistical analysis, improvements of the method in the case of noisy data, and methods for model structure detection are topics for future research. Another interesting direction of future research is using the embedding result for finding data-driven solutions to other data-driven problems, such as interpolation, smoothing, and control.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from: ERC Grant 258581, FWO projects G028015N and G090117N; and FNRS–FWO EOS Project 30468160.

REFERENCES

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

[2] I. Markovsky and P. Rapisarda, "Data-driven simulation and control," *Int. J. Control*, vol. 81, no. 12, pp. 1946–1959, 2008.

[3] T. Maupong and P. Rapisarda, "Data-driven control: A behavioral approach," *Syst. Control Lett.*, vol. 101, pp. 37–43, 2017.

[4] J. Coulson, J. Lygeros, and F. Dörfler, "Regularized and distributionally robust data-enabled predictive control," in *Proc. of the IEEE Conf. on Decision and Control*, Nice, France, December 2019, pp. 7165–7170.

[5] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," in *18th European Control Conference*, 2019, pp. 307–312.

[6] J. Coulson, J. Lygeros, and F. Dörfler, "Distributionally robust chance constrained data-enabled predictive control," https://arxiv.org/abs/2006.01702, May 2020.

[7] H. van Waarde, J. Eising, H. Trentelman, and K. Camlibel, "Data informativity: A new perspective on data-driven analysis and control," *IEEE Trans. Automat. Contr.*, 2019.

[8] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Trans. Automat. Contr.*, vol. 65, pp. 909–924, 2020.

[9] J. Rueda-Escobedo and J. Schiffer, "Data-driven internal model control of second-order discrete volterra systems," arXiv:2003.14158, 2020.

[10] J. Berberich and F. Allgöwer, "A trajectory-based framework for data-driven system analysis and control," in *European Control Conference*, 2020, pp. 1365–1370.

[11] A. Bisoffi, C. De Persis, and P. Tesi, "Data-based stabilization of unknown bilinear systems with guaranteed basin of attraction," arXiv preprint arXiv:2004.11630, 2020.

[12] S. Billings, *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons, 2013.

[13] I. Markovsky and K. Usevich, "Nonlinearly structured low-rank approximation," in *Low-Rank and Sparse Modeling for Visual Analysis*, Y. R. Fu, Ed. Springer, 2014, pp. 1–22.

[14] I. Markovsky, "On the behavior of autonomous Wiener systems," *Automatica*, vol. 110, p. 108601, 2019.

[15] I. Markovsky and F. Dörfler, "Identifiability in the behavioral setting," Vrije Universiteit Brussel, Tech. Rep., 2020.

[16] I. Markovsky, J. C. Willems, P. Rapisarda, and B. De Moor, "Algorithms for deterministic balanced subspace identification," *Automatica*, vol. 41, no. 5, pp. 755–766, 2005.

[17] B. De Moor, P. De Gersem, B. De Schutter, and W. Favoreel, "DAISY: A database for identification of systems," *Journal A*, vol. 38, no. 3, pp. 4–5, 1997, available from http://homes.esat.kuleuven.be/~smc/daisy/.

[18] J. Buckheit and D. Donoho, *Wavelets and statistics*. Springer-Verlag, 1995, ch. Wavelab and reproducible research.

[19] D. Knuth, "Literate programming," *Comput. J.*, vol. 27, no. 2, pp. 97–111, 1984.

[20] N. Ramsey, "Literate programming simplified," *IEEE Software*, vol. 11, pp. 97–105, 1994.

[21] T. Söderström, "Errors-in-variables methods in system identification," *Automatica*, vol. 43, pp. 939–958, 2007.

[22] I. Markovsky and S. Van Huffel, "High-performance numerical algorithms and software for structured total least squares," *J. Comput. Appl. Math.*, vol. 180, no. 2, pp. 311–331, 2005.

Ivan Markovsky is an Associate Professor at the department ELEC of the Vrije Universiteit Brussel. He received his Ph.D. degree in Electrical Engineering from the Katholieke Universiteit Leuven in February 2005. From 2006 to 2012 he was an Assistant Professor at the School of Electronics and Computer Science of the University of Southampton. He is a recipient of an ERC starting grant "Structured low-rank approximation: Theory, algorithms, and applications" 2010–2015, Householder Prize honorable mention 2008, and research mandate by the Vrije Universiteit Brussel research council 2012–2022. His main research interests are computational methods for system theory, identification, and data-driven control in the behavioral setting.