Data-driven simulation of generalized bilinear 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 simulation and control. This paper extends the approach to data-driven simulation of a class of nonlinear systems, called generalized bilinear. The generalized bilinear class includes Hammerstein, finite-lag Volterra, and bilinear systems. The key step of the generalization is an embedding result that is of independent interest. The behavior of a nonlinear system is included into the behavior of a linear time-invariant system. The method proposed is illustrated and compared with a model-based method on simulation examples and real-life data.**

***Index Terms*—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 (LTI) 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], [9]. The fundamental lemma and much of the subsequent results, however, assume LTI dynamics.

Generalizations of the fundamental lemma to nonlinear systems were proposed in [10], [11], [12]. The classes of nonlinear systems considered are: finite-lag second-order Volterra [10], Hammerstein-Wiener [11], and bilinear [12]. They are special cases of the Nonlinear Auto-Regressive eXogenous (NARX) model class [13]. In this paper, we consider the data-driven simulation problem for NARX systems and show that it is equivalent to structured low-rank matrix completion. Our main result is a generalization of the method of [2] to *generalized bilinear* systems, which include the finite-lag Volterra, Hammerstein, and bilinear systems as special cases.

The key result for generalizing 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 an LTI system. LTI embedding is used for deriving realization and identification methods for Wiener systems [14]. We generalize the result of [14] and use it for data-driven simulation of generalized bilinear systems.

The paper is organized as follows. Section II reviews the data-driven simulation problem and its solution in the case of LTI systems. Section III introduces the nonlinear model class. The main result of the paper—the generalization of data-driven simulation method for LTI systems to nonlinear 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 [15] are presented in Section V.

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

II. Data-driven simulation of LTI systems

Section II-A introduces notation and preliminary results for LTI 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 $w \in \mathbb{R}^{qL}$.

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 LTI 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 $\mathscr{L}^q$, called *finite dimensional*, admits a *kernel representation*

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

(1) is *minimal* if $\operatorname{row\,dim} R$ 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 $\mathscr{B}$ [16, Proposition X.5]. The minimal degree is invariant of the representation and is called the *lag* $\boldsymbol{\ell}(\mathscr{B})$ 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 $w(t) = \Pi \left[\begin{smallmatrix} u(t) \\ y(t) \end{smallmatrix}\right]$. Although the partitioning of the variables into inputs and outputs is not unique, the number of inputs $\boldsymbol{m}(\mathscr{B})$ is invariant of the choice of the partitioning.

An autonomous LTI system $\mathscr{B} \in \mathscr{L}^q$ is a system without inputs ($\boldsymbol{m}(\mathscr{B}) = 0$). The order $\boldsymbol{n}(\mathscr{B})$ of an autonomous LTI 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 $\dim \mathscr{B}|_L = \boldsymbol{m}(\mathscr{B})L + \boldsymbol{n}(\mathscr{B})$, for $L \geq \boldsymbol{\ell}(\mathscr{B})$, *i.e.*, the dimension of the restricted behavior $\mathscr{B}|_L$ is determined by the horizon's length L, number of inputs $\boldsymbol{m}(\mathscr{B})$, and order $\boldsymbol{n}(\mathscr{B})$ [17].

The *Hankel matrix* of $w_\mathrm{d} \in (\mathbb{R}^q)^T$ with L block rows is

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

A generalization of the Hankel matrix for a set of signals $w_\mathrm{d}^1, \dots, w_\mathrm{d}^N$ is the mosaic-Hankel matrix $\mathscr{H}_L(w_\mathrm{d}^1, \dots, w_\mathrm{d}^N) := \begin{bmatrix} \mathscr{H}_L(w_\mathrm{d}^1) & \cdots & \mathscr{H}_L(w_\mathrm{d}^N) \end{bmatrix}$. The Hankel matrix $\mathscr{H}_{L,j}(w_\mathrm{d})$ is the submatrix of $\mathscr{H}_L(w_\mathrm{d})$ 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 $w_\mathrm{ini} \in (\mathbb{R}^q)^{T_\mathrm{ini}}$ of length $T_\mathrm{ini} \geq \boldsymbol{\ell}(\mathscr{B})$. The condition that w is generated under the initial conditions specified by w_ini is that the concatenation $w_\mathrm{ini} \wedge w := \big(w_\mathrm{ini}(1), \dots, w_\mathrm{ini}(T_\mathrm{ini}), w(1), \dots, w(T)\big)$ 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. 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 $w_\mathrm{d} \in (\mathbb{R}^q)^T$ generated by an unknown system $\mathscr{B} \in \mathscr{M}$, initial conditions $w_\mathrm{ini} \in (\mathbb{R}^q)^\ell$, where $\ell \geq \boldsymbol{\ell}(\mathscr{B})$, and input $u_\mathrm{s} \in (\mathbb{R}^\mathtt{m})^{T_\mathrm{s}}$, find the output $y_\mathrm{s} \in (\mathbb{R}^\mathtt{p})^{T_\mathrm{s}}$ of $\mathscr{B}$ to u_s under the initial conditions w_ini, *i.e.*, $w_\mathrm{ini} \wedge \left[\begin{smallmatrix} u_\mathrm{s} \\ y_\mathrm{s} \end{smallmatrix}\right] \in \mathscr{B}$.

Trajectory-based representation: Consider an LTI system $\mathscr{B} \in \mathscr{L}^q$. For any trajectory $w_\mathrm{d} \in \mathscr{B}|_T$ and $L \in [1, T]$, due to linearity and time-invariance image $\mathscr{H}_L(w_\mathrm{d}) \subseteq \mathscr{B}|_L$. When equality holds, image $\mathscr{H}_L(w_\mathrm{d}) = \mathscr{B}|_L$ is a *data-driven representation* of $\mathscr{B}|_L$. Indeed, the image of the Hankel matrix 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, additional conditions have to be satisfied for the validity of the data-driven representation.

Theorem 2 (Corollary 19 [17])**.** *Consider $w_d \in \mathscr{B}|_T$, where $\mathscr{B} \in \mathscr{L}^q$ and $L \geq \boldsymbol{\ell}(\mathscr{B})$. Then, $\mathscr{B}|_L = \text{image}\, \mathscr{H}_L(w_d)$ iff*

$$\text{rank}\, \mathscr{H}_L(w_d) = \boldsymbol{m}(\mathscr{B})L + \boldsymbol{n}(\mathscr{B}). \tag{2}$$

The significance of the data-driven representation 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 ([18])**.** *Let $w_d \in \mathscr{B}|_T$, where $\mathscr{B} \in \mathscr{L}^q$, $L \geq \boldsymbol{\ell}(\mathscr{B})$, and (2) holds. Then, $w \in \mathscr{B}|_L$ if and only if there is $g \in \mathbb{R}^{T-L+1}$, such that $w = \mathscr{H}_L(w_d)g$.*

The data-driven simulation method of [2]: For an LTI 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 in terms of the data $(w_\mathrm{d}, w_\mathrm{ini}, u_\mathrm{s})$. This result is the data-driven simulation method of [2].

Proposition 4. *Assume that (2) holds. Then, the solution to Problem 1 is given by*

$$y_s = \mathscr{H}_{T_s}(\sigma^\ell y_d) \begin{bmatrix} \mathscr{H}_{\ell,j}(w_d) \\ \mathscr{H}_{T_s}(\sigma^\ell u_d) \end{bmatrix}^+ \begin{bmatrix} w_{ini} \\ u_s \end{bmatrix}, \tag{3}$$

where $j =: T - L + 1$ 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 (2) for the data-driven representation instead of the sufficient conditions of [1, Lemma 1], that are used in [2]. ■

III. The nonlinear model class

This section introduces the nonlinear model class. In its most general form it 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 valid trajectory 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:

$$\mathscr{B} := \{\, w \mid R\big(w, \sigma w, \dots, \sigma^\ell w\big) = 0 \,\}, \tag{4}$$

where $R : \mathbb{R}^{q(\ell+1)} \to \mathbb{R}^g$ and g is the number of equations in the representation. The representation (4) is a generalization of the kernel representation (1) for LTI system.

Next, we consider a special case of (4) when $q = 2$ and

$$R(w, \sigma w, \dots, \sigma^\ell w) = f\big(\mathbf{x}(w)\big) - \sigma^\ell y, \\ \text{with } \mathbf{x}(w) := \mathrm{vec}(w, \sigma w, \dots, \sigma^{\ell-1} w, \sigma^\ell u). \tag{5}$$

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

$$\sigma^\ell y = f(x) = f(u, y, \sigma u, \sigma y, \dots, \sigma^{\ell-1} u, \sigma^{\ell-1} y, \sigma^\ell u). \tag{6}$$

In (6), the variable u can be chosen freely while the variable y is determined by u and the initial conditions $w_\mathrm{ini} := \big(w(-\ell+1), \dots, w(0)\big)$. Thus, u is an input, y is an output, and (6) is an *input/output representation* of the system.

Note 5. 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:

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

For example, f can be an n_x-variate polynomial, *i.e.*,

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

The vector of monomials ϕ defines the *model structure*. In case of a polynomial R, the model structure ϕ is specified by the degrees n_{ij}. More generally, ϕ is a vector of basis functions. Once ϕ is fixed[1], a particular model is specified by the parameter vector θ

$$\mathscr{B}(\theta) := \left\{ w = \left[\begin{smallmatrix} u \\ y \end{smallmatrix}\right] \mid \sigma^{\ell} y = \theta^{\top} \phi\big(\mathbf{x}(w)\big) \right\}. \tag{7}$$

The model class with structure ϕ defined by (7) is denoted $\mathscr{P}_{\phi}$.

The input/output structure of (6) ensures existence and uniqueness of the solution of the simulation problem, *i.e.*, 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 $f(x) =: \theta_{\text{lin}}^{\top} \phi_{\text{lin}}(x) + \theta_{\text{nl}}^{\top} \phi_{\text{nl}}(x)$, 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 $\theta_{\text{lin}}^{\top} \phi_{\text{lin}}(x)$ *fully parameterized*. If $\phi_{\text{nl}} = 0$, the model $\mathscr{B}_{\theta}$ is LTI. Depending on the structure of ϕ_{nl}, we have the following classes of nonlinear systems.

- *Hammerstein system:* ϕ_{nl} is of the form
$$\phi_{\text{nl}}\big(x(t)\big) = \big[\phi_{\text{h}}\big(u(t)\big) \;\; \phi_{\text{h}}\big(u(t+1)\big) \;\; \cdots \;\; \phi_{\text{h}}\big(u(t+\ell)\big)\big]^{\top}, \quad \text{for some } \phi_{\text{h}} : \mathbb{R} \to \mathbb{R}.$$
- *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
$$x_u(t) := \operatorname{vec}\big(u(t), u(t+1), \ldots, u(t+\ell)\big).$$
- *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
$$x_y(t) := \operatorname{vec}\big(y(t), y(t+1), \ldots, y(t+\ell-1)\big)$$
and $\otimes$ is the Kroneker product.
- *Generalized bilinear systems:* ϕ_{nl} is affine in x_y, *i.e.*, there is $\phi_{\text{b},0} \in (\mathbb{R}^{n_{\text{nl}}})^{\mathbb{N}}$ and $\phi_{\text{b}} : \mathbb{R}^{\ell+1} \to \mathbb{R}^{n_{\text{b}}}$, such that
$$\phi_{\text{nl}}\big(x(t)\big) = \phi_{\text{b},0}(t) + \phi_{\text{b}}\big(x_u(t)\big) \otimes x_y(t). \tag{8}$$

Lemma 6 (Matrix representation of (8))**.** *For a generalized bilinear system, $\mathscr{B} \in \mathscr{P}_{\phi}$ and a finite signal $w = \left[\begin{smallmatrix} u \\ y \end{smallmatrix}\right] \in (\mathbb{R}^2)^T$ with $T > \ell$, it holds that $\phi_{b,0} = \phi_{nl}\big(\mathbf{x}(\left[\begin{smallmatrix} u \\ 0 \end{smallmatrix}\right])\big) \in \mathbb{R}^{n_{nl}(T-\ell)}$ and*

$$\phi_b(x_u) \otimes x_y = \Phi_b y, \quad \text{where } \Phi_b \in \mathbb{R}^{n_{nl}(T-\ell) \times T} \text{ is given by}$$
$$\Phi_b = \big[\phi_{nl}\big(\mathbf{x}(\left[\begin{smallmatrix} u \\ \delta \end{smallmatrix}\right])\big) - \phi_{b,0} \;\cdots\; \phi_{nl}\big(\mathbf{x}(\left[\begin{smallmatrix} u \\ \sigma^{T-1}\delta \end{smallmatrix}\right])\big) - \phi_{b,0}\big] \tag{9}$$

with δ being the unit pulse $(1, 0, \ldots, 0) \in \mathbb{R}^T$.

Proof: Follows from the fact that ϕ_{nl} is affine in x_y. ■

Example 7 (Hammerstein system). A simple example with lag $\ell = 1$ is the Hammerstein system with quadratic nonlinearity:

$$y(t+1) = 0.95 y(t) + u(t+1) + 2u^2(t+1). \tag{10}$$

In this case, $x(t) = \operatorname{col}\big(u(t), y(t), u(t+1)\big)$ and $\phi\big(x(t)\big) = \operatorname{col}\big(y(t), u(t+1), u^2(t+1)\big)$, with parameter vector identifying the system $\mathscr{B}(\theta) \in \mathscr{P}_{\phi}$ defined by (10), $\theta = \operatorname{col}(0.95, 1, 2)$.

[1]Choosing ϕ is called *structure selection*. We do not address the structure selection problem in this paper and consider ϕ as a user specified hyper-parameter. Methods for structure selection based on sparsity promoting regularization are described in [19], [20], [21].

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

$$\begin{aligned} y(t+2) = {} & -p_0 y(t) - p_1 y(t+1) + q_0 u(t) + \\ & q_1 u(t+1) + q_2 u(t+2) + 0.1u^2(t+2) + \\ & 0.1u(t+1)u(t+2) + 0.1u(t)u(t+1)u(t+2). \end{aligned} \tag{11}$$

In this case, $x(t) = \operatorname{col}\big(u(t), y(t), u(t+1), y(t+1), u(t+2)\big)$ and $\phi\big(x(t)\big) = \left[\begin{smallmatrix} x(t) \\ \phi_{\text{nl}}(x(t)) \end{smallmatrix}\right]$, *i.e.*, the "linear part" of (11) is fully parametrized and the nonlinear part has three terms—two of degree 2 and one of degree 3. $\theta = \operatorname{col}(\theta_{\text{l}}, \theta_{\text{n}})$, where $\theta_{\text{l}} = \operatorname{col}(-p_0, -p_1, q_0, q_1, q_2)$ and $\theta_{\text{n}} = \operatorname{col}(0.1, 0.1, 0.1)$.

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

$$\begin{aligned} y(t+2) = {} & -p_0 y(t) - p_1 y(t+1) + q_0 u(t) + \\ & q_1 u(t+1) + q_2 u(t+2) + 0.1y(t)u^2(t+2) + \\ & 0.1u(t+1)y(t+1)u(t+2) + 0.1u(t)u(t+1)u(t+2). \end{aligned} \tag{12}$$

The $x(t)$ vector is the same as in Example 8. The nonlinear part of (12) now consists of three terms of degree 3, with two involving both inputs and outputs, so that the system is not of the finite-lag Volterra type. The parameter vector identifying the system defined by (12) is the same as in Example 8.

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

From (4) and (6), we have $w \in \mathscr{B}(\theta) \in \mathscr{P}_{\phi}$ if and only if

$$\begin{bmatrix} \theta^{\top} & -1 \end{bmatrix} \underbrace{\begin{bmatrix} \phi\big(x(1)\big) & \cdots & \phi\big(x(T-\ell)\big) \\ y(\ell+1) & \cdots & y(T) \end{bmatrix}}_{\mathscr{S}(w)} = 0. \tag{13}$$

In general, the matrix $\mathscr{S}(w)$ is nonlinearly structured, *i.e.*, the map $w \mapsto \mathscr{S}(w)$ is nonlinear. In the LTI case, with fully parameterized model, *i.e.*, $\phi(x) = x$, the structure is Hankel.

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 (13):

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

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

The generalization of (13) to multiple trajectories $w^1, \ldots, w^N$ is straightforward:

$$\begin{bmatrix} \theta^{\top} & -1 \end{bmatrix} \underbrace{\begin{bmatrix} \mathscr{S}(w^1) & \cdots & \mathscr{S}(w^N) \end{bmatrix}}_{\mathscr{S}(w^1, \ldots, w^N)} = 0. \tag{14}$$

In case of a fully parameterized LTI system, the data matrix $\mathscr{S}(w^1, \ldots, w^N)$ is mosaic-Hankel $\mathscr{H}_{\ell+1}(w^1, \ldots, w^N)$, see [17].

IV. Data-driven simulation of nonlinear systems

Since both w_{d} and w_{s} are trajectories of $\mathscr{B}(\theta)$, by (14)

$$\operatorname{rank} \begin{bmatrix} \mathscr{S}(w_{\text{d}}) & \mathscr{S}(w_{\text{s}}) \end{bmatrix} \leq n_{\theta}. \tag{15}$$

The nonlinear data-driven simulation problem can be restated then as a matrix completion problem: find y_{s}, so that (15) holds. The problem can be solved by the following divide-and-conquer model-based strategy:

1) *system identification:* compute an estimate $\widehat{\theta}$ for the parameter vector θ, and
2) *model-based simulation:* using $\widehat{\theta}$, compute y_s from (6).

For well-posedness of the identification problem on step 1 the left kernel of $\mathscr{S}(w_d)$ has to be one-dimensional.

Theorem 10 (Identifiability condition)**.** *Consider a 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 $\widehat{\theta}_{ext}$ be a basis vector for the left kernel of $\mathscr{S}(w_d)$. Then, $\widehat{\theta} := \frac{-1}{\widehat{\theta}_{ext,n_\theta+1}} \begin{bmatrix} \widehat{\theta}_{ext,1} & \cdots & \widehat{\theta}_{ext,n_\theta} \end{bmatrix}^\top = \theta$,* i.e., *$\widehat{\theta}$ identifies the data-generating system $\mathscr{B}(\theta)$.*

Proof: By the assumption that $\operatorname{rank} \mathscr{S}(w_d) = n_\theta$, the left kernel of $\mathscr{S}(w_d)$ has dimension equal to one. Then, there is $\alpha \neq 0$, such that $\widehat{\theta}_{ext} = \alpha \begin{bmatrix} \theta^\top & -1 \end{bmatrix}$, and therefore $\widehat{\theta} = \theta$. ■

The rank condition, $\operatorname{rank} \mathscr{S}(w_d) = n_\theta$, is a *persistency of excitation* type condition. In the case of an LTI system, Theorem 10 is a special case of the identifiability result of [17].

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. LTI embedding

A key idea exploited next is to embed the nonlinear system $\mathscr{B} = \mathscr{B}(\theta) \in \mathscr{P}_\phi$ into an LTI system $\mathscr{B}_{ext}$ by replacing the vector $\phi_{nl}\big(\mathbf{x}(w)\big)$ with an additional input variable $u_{nl} \in (\mathbb{R}^{n_{nl}})^\mathbb{N}$, where $n_{nl} := \dim \phi_{nl}(x)$. As a result the embedding

$$\mathscr{B}_{ext} := \left\{ w_{ext} = \left[\begin{smallmatrix} w \\ u_{nl} \end{smallmatrix}\right] \;\middle|\; \sigma^\ell y = \theta_{lin}^\top \phi_{lin}\big(\mathbf{x}(w)\big) + \theta_{nl}^\top u_{nl} \right\} \tag{16}$$

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

Lemma 11 (LTI embedding)**.** *The system $\mathscr{B} \in \mathscr{P}_\phi$ is embedded in the LTI system $\mathscr{B}_{ext}$ defined in (16),* i.e., *$\mathscr{B} \subseteq \Pi_w \mathscr{B}_{ext}$. Moreover, the embedding is exact,* i.e., *$\mathscr{B} = \Pi_w \mathscr{B}_{ext}$, when the extra constraint $u_{nl} = \phi_{nl}\big(\mathbf{x}(w)\big)$ is imposed,* i.e.,

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

Proof: Let $w = \left[\begin{smallmatrix} u \\ y \end{smallmatrix}\right] \in \mathscr{B}$ and $\mathscr{B} = \mathscr{B}(\theta)$. Then,

$$\sigma^\ell y = \theta_{lin}^\top \phi_{lin}\big(\mathbf{x}(w)\big) + \theta_{nl}^\top u_{nl}, \quad \text{with } u_{nl} = \phi_{nl}\big(\mathbf{x}(w)\big).$$

It follows by the definition of (16) that $w \in \Pi_w \mathscr{B}_{ext}$. Therefore, $\mathscr{B} \subseteq \Pi_w \mathscr{B}_{ext}$. For w, such that $w \in \Pi_w \mathscr{B}_{ext}$, in general, it is not true that $w \in \mathscr{B}$. Indeed,

$$u_{nl} := \Pi_{u_{nl}} w_{ext} = \phi_{nl}\big(\mathbf{x}(\left[\begin{smallmatrix} u_{ini} \\ u_s \end{smallmatrix}\right])\big) \tag{17}$$

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

Example 12. The LTI relaxation $\mathscr{B}_{ext}$ of the Hammerstein system in example 7 is defined by $y(t+1) = 0.95y(t) + u_1(t+1) + 2u_2(t)$. $\mathscr{B}_{ext}|_T$ is a subspace of $\mathbb{R}^{3T}$ with $\dim \mathscr{B}_{ext}|_T = 2T + 2$, while $\mathscr{B}$ is a manifold of $\mathbb{R}^{2T}$ with $\dim \mathscr{B}|_T = T + 2$.

B. Generalization of (3)

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

Let $\mathscr{B}_{ext}$ be the LTI embedding of $\mathscr{B} \in \mathscr{P}_\phi$ and define

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

As in Section II-B, the signal w_{ext} has length L and is a concatenation of initial conditions $w_{ini,ext}$ of length ℓ and a to-be-simulated signal $w_{s,ext}$ of length T_s. The data-driven simulation method (3) applied to $\mathscr{B}_{ext}$ gives us

$$y_{s,ext} = \mathscr{H}_{T_s}(\sigma^\ell y_d) \underbrace{\begin{bmatrix} \mathscr{H}_{\ell,j}(w_{d,ext}) \\ \mathscr{H}_{T_s}(\sigma^\ell u_{d,ext}) \end{bmatrix}^+}_{\mathscr{A}^+(w_d)} \begin{bmatrix} w_{ini,ext} \\ u_{s,ext} \end{bmatrix}, \tag{18}$$

where $j =: T - L + 1$. If (17) holds true, we have that by Lemma 11 $\mathscr{B} = \mathscr{B}_{ext}$ and therefore $y_{s,ext}$ defined in (18) is the response of $\mathscr{B}$ to input u_s and initial conditions w_{ini}. Then, (18) gives us the solution to the data-driven simulation problem.

The extended signal $w_{d,ext}$ 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 (18) can be used for data-driven simulation of generalized bilinear systems, where u_{nl} depends on both u_s and y_s, however, u_{nl} is affine in y_s.

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 LTI embedding (16) of $\mathscr{B}$. Assume that*

$$\operatorname{rank} \mathscr{H}_L(w_{d,ext}) = (1 + n_{nl})L + \ell \tag{19}$$

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

$$y_s := \mathscr{H}_{T_s}(\sigma^\ell y_d)\mathscr{A}^+(w_d) \left[\begin{smallmatrix} w_{ini} \\ u_s \\ \Phi_{b,p} y_{ini} + \phi_{b,0} \end{smallmatrix}\right], \tag{20}$$

where

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

and $\begin{bmatrix} \Phi_{b,p} & \Phi_{b,f} \end{bmatrix} := \Phi_b$, with Φ_b defined in (9).

Proof: By (19), using Corollary 3, we have that there is g such that $\mathscr{H}_{\ell+L}(w_{d,ext})g = w_{ext}$. Partitioning the equations according to the partitioning $\operatorname{col}(u, y, u_{nl})$ of w_{ext}, we have

$$\begin{bmatrix} \mathscr{H}_L(u_d) \\ \mathscr{H}_L(y_d) \\ \mathscr{H}_{T_s}(u_{d,nl}) \end{bmatrix} g = \begin{bmatrix} \left[\begin{smallmatrix} u_{ini} \\ u_s \end{smallmatrix}\right] \\ \left[\begin{smallmatrix} y_{ini} \\ y_s \end{smallmatrix}\right] \\ u_{nl} \end{bmatrix}. \tag{21}$$

The equation $y_s = \mathscr{H}_L(\sigma^\ell y_d)g$ 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 (17) has to be added to (21). Since $\mathscr{B}$ is a generalized bilinear system, by Lemma 6, (17) becomes

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

Substituting back in (21) and using $y_{\text{s}} = \mathscr{H}_{T_{\text{s}}}(\sigma^{\ell} y_{\text{d}})g$, we obtain

$$\mathscr{A}(w_{\text{d}})g = \left[\begin{smallmatrix} w_{\text{ini}} \\ u_{\text{s}} \\ \phi_{\text{b},0} + \Phi_{\text{b,p}} y_{\text{ini}} \end{smallmatrix}\right]. \qquad \blacksquare$$

Note 14 (Persistency of excitation assumption). The key assumption of Theorem 13 is the persistency of excitation assumption (19). The matrix $\mathscr{H}_L(w_{\text{d,ext}})$ is of dimension $(2+n_{\text{nl}})L \times (T-L+1)$, where $n_{\text{nl}} := \dim \phi_{\text{nl}}$ is the number of nonlinear terms in the representation (6) of the system. A necessary condition for (19) then is $T \geq (3+n_{\text{nl}})L - 1$.

Note 15 (Comparison with model-based simulation). In comparison with the identifiability condition $\operatorname{rank} \mathscr{S}(w_{\text{d}}) = n_{\theta}$ of Theorem 10, the condition (19) for data-driven simulation is stronger. This is true also in the case of LTI systems. As shown in [22, 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 $M : \mathbb{R}^{2\ell+T_{\text{s}}} \mapsto \mathbb{R}^{T_{\text{s}}}$ defined by

$$M(w_{\text{ini}}, u_{\text{s}}) := \mathscr{H}_{T_{\text{s}}}(\sigma^{\ell} y_{\text{d}}) \mathscr{A}^{+}(w_{\text{d}}) \left[\begin{smallmatrix} w_{\text{ini}} \\ u_{\text{s}} \\ \Phi_{\text{b,p}} y_{\text{ini}} + \phi_{\text{b},0} \end{smallmatrix}\right].$$

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 LTI embedding (16) of $\mathscr{B}$. Assume that (19) holds. Then, the solution to Problem 1 is*

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

Theorem 13 gives a constructive method for data-driven simulation of generalized bilinear systems, in particular, affine, Hammerstein, finite-lag Volterra, and bilinear systems. The next section shows 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 [15]. The simulation results are made reproducible in the sense of [23] by providing the implementation of the method and the data generating scripts [24].

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 bilinear—and do data-driven simulation of

S1: response to random initial conditions and random input,
S2: free response—nonzero initial conditions and zero input,
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 $T_{\text{s}} = 10$ samples.

The result $\widehat{y}_{\text{s}}$ obtained by the data-driven simulation method (20) is evaluated by the relative error $e = 100\% \, \|y_{\text{s}} - \widehat{y}_{\text{s}}\| / \|y_{\text{s}}\|$, where y_{s} is the response obtained by model-based simulation. In all experiments the relative error is of the order of the machine precision ($e < 10^{-13}$). This is an empirical confirmation 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 [25]: $w_{\text{d}} = \overline{w}_{\text{d}} + \widetilde{w}_{\text{d}}$, where $\overline{w}_{\text{d}} \in \mathscr{B}|_T$ and $\widetilde{w}_{\text{d}} \sim \mathrm{N}(0, s^2 I)$. The *true value* $\overline{w}_{\text{d}}$ is a trajectory of the to-be-simulated system $\mathscr{B} \in \mathscr{P}_{\phi}$ and the *measurement noise* $\widetilde{w}_{\text{d}}$ 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 (11). The trajectory w_{d} has $T = 1000$ samples and w_{s} has $T_{\text{s}} = 10$ samples. The simulated response $\widehat{y}_{\text{s}}$, computed by the data-driven simulation algorithm, is evaluated by the relative error e, where y_{s} is computed by model-based simulation using the true model $\mathscr{B}$.

Figure 1 shows the relative 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 using more effectively the data by shaping it in the matrix $\mathscr{S}(w_{\text{d}})$ rather than (21). This is related to the weaker persistency of exaction for identification compared to the one for data-driven simulation, see Note 14.

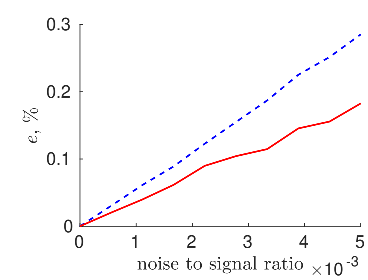

Fig. 1. Relative error e as a function of the noise-to-signal ratio: solid line—model-based method, dashed line—data-driven method.

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. 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" $w_{\text{d}} \in \mathbb{R}^T$ and the remaining 20% as the "to-be-simulated trajectory" $w_{\text{ini}} \wedge w_{\text{s}} \in \mathbb{R}^L$. 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 [26].

The computed responses $\widehat{y}_s$ by the data-driven simulation method and the model-based method are compared with the measured response y_s in terms of the relative error e. 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 ERROR FOR THE MODEL-BASED METHOD, e_{DD}—RELATIVE ERROR FOR THE DATA-DRIVEN METHOD. ("FAIL" MEANS $e > 100\%$)

	data set name	T	L	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 nonlinear systems (7) and showed that the problem is equivalent to matrix completion. The solution method proposed in the paper is a generalization of the method of [2].The key technical result for the generalization is LTI embedding: the behavior of the nonlinear system is included in the behavior of an LTI system. Under an additional 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 LTI 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. Empirical results show the performance of the method in case of simulated noisy data and real-life benchmark problems. In case of noisy data simulated, 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 gives better results. Theoretical justification for this performance is a topic of future work. Other important directions for future work include generalization to multivariable systems, statistical analysis, methods for model structure detection, and using the embedding result for other data-driven problems, such as interpolation, smoothing, and control.

Acknowledgements

Seven anonymous referees contributed to this paper by many detailed and constructive comments. 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 IEEE Conf. on Decision and Control*, 2019, pp. 7165–7170.

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

[6] J. Coulson, J. Lygeros, and F. Dörfler, "Distributionally robust chance constrained data-enabled predictive control," *IEEE Transactions on Automatic Control*, 2020, available at https://arxiv.org/abs/2006.01702.

[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] I. Markovsky, "Recent progress on variable projection methods for structured low-rank approximation," *Signal Processing*, vol. 96PB, pp. 406–419, 2014.

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

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

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

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

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

[15] 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/.

[16] J. C. Willems, "Paradigms and puzzles in the theory of dynamical systems," *IEEE Trans. Automat. Contr.*, vol. 36, no. 3, pp. 259–294, 1991.

[17] I. Markovsky and F. Dörfler, "Identifiability in the behavioral setting," 2020, available at http://homepages.vub.ac.be/~imarkovs/publications/identifiability.pdf.

[18] ——, "Data-driven dynamic interpolation and approximation," Vrije Universiteit Brussel, Tech. Rep., 2021, available at http://homepages.vub.ac.be/~imarkovs/publications/ddint.pdf.

[19] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.

[20] S. Brunton, J. Proctor, and N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proc. Nat. Academy of Sci.*, vol. 113, pp. 3932–3937, 2016.

[21] I. Markovsky, "Application of low-rank approximation for nonlinear system identification," in *25th IEEE Mediterranean Conference on Control and Automation*, Valletta, Malta, July 2017, pp. 12–16.

[22] 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.

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

[24] I. Markovsky, "Data-driven simulation of nonlinear systems via linear time-invariant embedding," Vrije Universiteit Brussel, Tech. Rep., 2021, available at http://homepages.vub.ac.be/~imarkovs/publications/ddsim-narx.pdf.

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

[26] 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.