Data-Driven Output Matching of Output-Generalized Bilinear and Linear Parameter-Varying systems

Leander Hemelhof[1], Ivan Markovsky[2] and Panagiotis Patrinos[3]

***Abstract*— There is a growing interest in data-driven control of nonlinear systems over the last years. In contrast to related works, this paper takes a step back and aims to solve the output matching problem, a problem closely related to the reference tracking control problem, for a broader class of nonlinear systems called output-generalized bilinear (OGB), thereby offering a new direction to explore for data-driven control of nonlinear systems. It is shown that discrete-time linear parameter-varying systems are included in this model class, with affine systems easily shown to also be included. The proposed model class and method are illustrated using the simulation of a real-life system.**

***Index Terms*— Nonlinear system theory, Linear parameter-varying systems, Behavioral approach, Data-driven methods.**

I. Introduction

Data-driven simulation and control of nonlinear systems has been a popular topic over the last years. Some results have already been obtained for specific classes of nonlinear systems. A few examples are simulation of Wiener and Hammerstein systems [1], control of linear parameter-varying systems with affine dependence on the parameters [2], [3], simulation of generalized bilinear systems [4], control of second-order discrete Volterra systems [5], control of discrete-time *Multiple-Input Multiple-Output* (MIMO) feedback linearizable systems [6], and control of flat nonlinear systems [7]. More general control results also exist, for example making use of local linear approximations [8], kernels [9], [10] or the Koopman operator [11], [12].

When dealing with problems that are too difficult to solve exactly, like optimal direct data-driven control of arbitrary nonlinear systems, there are two options to continue. The first option is to solve the difficult problem approximately, aiming to get as close as possible to the true optimal solution. The second option is to solve a simplified version of the problem exactly, adding more complexity step by step to get as close to the original problem as possible.

In this work we will choose the second option, solving the output matching problem in the deterministic setting for a broad class of nonlinear systems by making use of a user-specified set of basis functions. The output matching problem was originally discussed in [13] while building up to reference tracking control. Since the output matching problem is closely related to the reference tracking control problem, this work offers an additional direction to explore for data-driven control of nonlinear systems. The main contributions of this paper are the following:

1) We define a novel model class that encompasses multiple other model classes discussed in the literature, among which are linear parameter-varying systems with known parameter values, and affine systems.
2) We provide a method to solve the output matching problem for this model class in the deterministic case.

Section II introduces the notation and definitions necessary in the following sections. Section III introduces the output-generalized bilinear model class and gives some examples. Section IV gives the main theorem of this work, offering a method to solve the output matching problem for our model class. Section V gives a simulation example, illustrating that the proposed method works. Section VI concludes this work and offers some final remarks regarding future work.

II. Preliminaries

In the behavioral framework [14] we define a discrete-time dynamical system $\mathscr{B}$ as a set of trajectories. We write for example $\mathscr{B} \subset (\mathbb{R}^q)^{\mathbb{N}}$, where $(\mathbb{R}^q)^{\mathbb{N}}$ is the set of q-variate signals. $\mathscr{B}|_L$ is the subset containing all possible trajectories of length L of the system. $\mathscr{L}^q$ is the set of *linear time-invariant* (LTI) models with q variables. The complexity of an LTI system $\mathscr{B}$ is characterized by three integers: the order $\mathbf{n}(\mathscr{B})$, the lag $\mathbf{l}(\mathscr{B})$, and the number of inputs $\mathbf{m}(\mathscr{B})$, which are all three properties of the system [14]. We define the forward shift operator σ as $\sigma y(t) = y(t+1)$. The Kronecker product of two matrices A and B is denoted by $A \otimes B$. We define the concatenation operator as $w_1 \wedge w_2 = \begin{bmatrix} w_1 & w_2 \end{bmatrix}$, with $w_1 \in (\mathbb{R}^q)^{T_1}$ and $w_2 \in (\mathbb{R}^q)^{T_2}$. The block Hankel matrix of a trajectory $w \in (\mathbb{R}^q)^T$ with L block rows is denoted by $\mathscr{H}_L(w)$, with

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

*This work is supported by: the Research Foundation Flanders (FWO) research projects G081222N, G033822N, G0A0920N; European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 953348; Research Council KU Leuven C1 project No. C14/18/068; Fonds de la Recherche Scientifique – FNRS and the Fonds Wetenschappelijk Onderzoek – Vlaanderen under EOS project no 30468160 (SeLMA). FNRS–FWO EOS Project 30468160. IM is funded by the Catalan Institution for Research and Advanced Studies (ICREA).

[1,3]L. Hemelhof and P. Patrinos are with the Department of Electrical Engineering (ESAT-STADIUS), KU Leuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium. Email: {leander.hemelhof, panos.patrinos}@esat.kuleuven.be

[2]Ivan Markovsky is with the the International Centre for Numerical Methods in Engineering (CIMNE), Gran Capitàn, Barcelona, Spain and the Catalan Institution for Research and Advanced Studies (ICREA), Pg. Lluis Companys 23, Barcelona, Spain. Email: imarkovsky@cimne.upc.edu

The truncation of a trajectory $w \in (\mathbb{R}^q)^T$ written as $w_{[T_1:T_2]}$, with $T_1 \leq T_2 \leq T$ denotes sequence

$$(w(T_1), w(T_1+1), \ldots, w(T_2)) \in (\mathbb{R}^q)^{T_2-T_1+1}.$$

A missing value before or after the separator indicates the start or end of the sequence respectively. For a sequence $x \in (\mathbb{R}^q)^T$ we define its vectorized version as

$$\mathrm{col}(x) = \begin{bmatrix} x(1)^\top & x(2)^\top & \ldots & x(T)^\top \end{bmatrix}^\top \in \mathbb{R}^{qT}.$$

Formally, the data-driven output matching problem can be formulated in the behavioral framework as follows [13, Problem 13]:

Problem 1 (Data-driven output matching): For a system $\mathscr{B}$ and with

- a trajectory $w_\mathrm{d} \in (\mathbb{R}^q)^T$ of $\mathscr{B}|_T$,
- an input-output partitioning $Pw = \begin{bmatrix} u \\ y \end{bmatrix}$ with P a permutation matrix and $q = n_u + n_y$,
- a reference output $y_\mathrm{r} \in (\mathbb{R}^{n_y})^L$,
- an initial trajectory $w_\mathrm{ini} \in (\mathbb{R}^q)^{T_\mathrm{ini}}$ of $\mathscr{B}|_{T_\mathrm{ini}}$,

find an input $u \in (\mathbb{R}^{n_u})^L$ such that $w_\mathrm{ini} \wedge (u, y_\mathrm{r}) \in \mathscr{B}|_{T_\mathrm{ini}+L}$.

This problem does not always have a solution. Consider for example an LTI system, defined by $y(t) = 0.5y(t-1) + u(t-1)$. If we give a reference output $y_\mathrm{r} = \begin{bmatrix} 2 & 2 \end{bmatrix}$ while $u_\mathrm{ini} = 0$ and $y_\mathrm{ini} = 5$, no matter what input we give to the system, it will be impossible to match the reference output.

This work deals with the considered nonlinear systems by using LTI embedding, where we embed the nonlinearity into an LTI system. This allows us to use results meant for LTI systems. With the goal of designing a data-driven method in mind, we will make use of the data-driven representation of LTI systems. This representation is given by following lemma [4, Corollary 3].

Lemma 1 (Data-driven representation): Consider $\mathscr{B} \in \mathscr{L}^q$, data trajectory $w_\mathrm{d} \in \mathscr{B}|_T$ and $L \geq \mathbf{l}(\mathscr{B})$. Assuming

$$\mathrm{rank}\, \mathscr{H}_L(w_\mathrm{d}) = \mathbf{m}(\mathscr{B})L + \mathbf{n}(\mathscr{B}), \tag{1}$$

then $w \in \mathscr{B}|_L$ if and only if there exists a vector $g \in \mathbb{R}^{T-L+1}$ such that

$$\mathscr{H}_L(w_\mathrm{d})g = w. \tag{2}$$

Condition (1) is referred to as the *generalized persistency of excitation* (GPE) condition. A sufficient condition is given by following lemma [15].

Lemma 2 (Sufficient condition for GPE): Consider an LTI system $\mathscr{B} \in \mathscr{L}^q$, a data trajectory with chosen input-output partitioning $w_\mathrm{d} = \begin{bmatrix} u_\mathrm{d} \\ y_\mathrm{d} \end{bmatrix} \in \mathscr{B}|_T$ and $L \geq \mathbf{l}(\mathscr{B})$. Under the assumption that

- $\mathscr{B}$ is controllable,
- u_d is persistently exciting of order $L + \mathbf{n}(\mathscr{B})$ ($\mathscr{H}_{L+\mathbf{n}(\mathscr{B})}(u_d)$ is full row rank),

rank condition (1) holds.

Note that since the number of inputs is a property of the system, it is independent of the chosen input-output partitioning.

Remark 1 (Minimal number of samples): The condition on the persistency of excitation of the input offers a lower bound on the minimal number of samples needed in the data trajectory. To fulfil the persistency of excitation condition, the number of columns given by $T - (L+\mathbf{n}(\mathscr{B}))+1$ needs to be at least $\mathbf{m}(\mathscr{B})(L+\mathbf{n}(\mathscr{B}))$, leading to

$$T \geq (\mathbf{m}(\mathscr{B})+1)(L+\mathbf{n}(\mathscr{B})) - 1.$$

It turns out that in almost all cases we only need the input to be persistently exciting of order L, except in the case of specific initial conditions that can be characterized in terms of the solution of a Sylvester equation [15]. In the case without these special initial conditions the true minimal number of samples is determined by the number of columns needed to fulfill rank condition (1), and is given by

$$T \geq (\mathbf{m}(\mathscr{B})+1)L + \mathbf{n}(\mathscr{B}) - 1.$$

Most of the time this distinction makes little difference. Only when dealing with very high order systems or with many inputs does this become relevant.

III. Output-generalized bilinear systems

A. Output-generalized bilinear model class

We consider a MIMO system $\mathscr{B}$ that can be written in the form

$$\begin{aligned} y(t) &= f(x_y(t), x_u(t), t) \\ &= \theta_\mathrm{lin}^\top x_w(t) + \theta_\mathrm{nl}^\top \phi_\mathrm{nl}(x_y(t), x_u(t), t), \end{aligned}$$

where

$$\begin{aligned} x_y(t) &= \begin{bmatrix} y(t-l)^\top & y(t-l+1)^\top & \ldots & y(t-1)^\top \end{bmatrix}^\top, \\ x_u(t) &= \begin{bmatrix} u(t-l)^\top & u(t-l+1)^\top & \ldots & u(t)^\top \end{bmatrix}^\top, \\ x_w(t) &= \begin{bmatrix} x_y(t)^\top & x_u(t)^\top \end{bmatrix}^\top, \end{aligned}$$

for all $t \in \mathbb{Z}$ and with $l \in \mathbb{N}_0$, $\theta_\mathrm{lin} \in \mathbb{R}^{ln_y+(l+1)n_u \times n_y}$, $\theta_\mathrm{nl} \in \mathbb{R}^{n_\mathrm{b} \times n_y}$, and with n_b the number of functions in

$$\phi_\mathrm{nl} : \mathbb{R}^{ln_y} \times \mathbb{R}^{(l+1)n_u} \times \mathbb{Z} \to \mathbb{R}^{n_\mathrm{b}}.$$

The only limitation put on the nonlinear term ϕ_nl is that it is affine in x_u given x_y and t. Throughout the remainder of this work, the past horizon of the nonlinear term will be denoted by l, not to be confused with the lag of an LTI system denoted by $\mathbf{l}(\mathscr{B}_\mathrm{LTI})$. Time is allowed to be negative to capture the influence of the infinite past of the system. In practice, we have signals such that $t \in \mathbb{N}$.

To ease notation, from this point on we will not write the time dependence explicitly unless relevant. We only have to remember that ϕ_nl can depend on time. This way we obtain

$$y = f(x_y, x_u) = \theta_\mathrm{lin}^\top x_w + \theta_\mathrm{nl}^\top \phi_\mathrm{nl}(x_y, x_u), \tag{3}$$

where

$$x_y = \begin{bmatrix} \sigma^{-l}y^\top & \sigma^{-l+1}y^\top & \dots & \sigma^{-1}y^\top \end{bmatrix}^\top,$$
$$x_u = \begin{bmatrix} \sigma^{-l}u^\top & \sigma^{-l+1}u^\top & \dots & u^\top \end{bmatrix}^\top,$$
$$x_w = \begin{bmatrix} x_y^\top & x_u^\top \end{bmatrix}^\top,$$

for all time $t \in \mathbb{Z}$, with σ the forward shift operator. The main use case of the explicit time dependence is to allow the nonlinear terms to depend on other known signals, which are interpreted here as time-dependent functions.

A similar *Single-Input Single-Output* (SISO) model class, called *generalized bilinear*, is introduced in [4]. It has the same form as used here, but the nonlinear term is given by

$$\phi_{\text{nl}}(x_u, x_y) = \phi_{\text{b},0} + \phi_{\text{b}}(x_u) \otimes x_y.$$

In contrast to this generalized bilinear model class, we refer to our model class as *output-generalized bilinear* (OGB). The main differences between the two are the role of input and output being swapped, the generalization to MIMO systems, and allowing for a time dependence in ϕ_{nl}.

Remark 2 (Known bijective transformation): Stepping away from the data-driven setting for a moment, based on prior knowledge of the nonlinear terms it is allowed to introduce a bijective transformation on the current and past inputs as long as the inverse of this transformation is also known. This generalization could be useful when trying to solve the output matching problem of a known system that is input affine in a transformed input space. To keep the proposed results as data-driven as possible within the setting of this paper, this transformation is only mentioned, but not used further.

B. Illustrative examples

To illustrate this model class, we offer two examples of systems that can be interpreted as an OGB system.

- *Simple SISO nonlinear system.* Let

  $$y(t) = 0.3\sin(y(t-1))u(t) + 0.2y(t-1)^2 + 1.5u(t-1).$$

  Taking

  $$\phi_{\text{nl}}(x_y(t), x_u(t), t) = \begin{bmatrix} \sin(y(t-1))u(t) & y(t-1)^2 \end{bmatrix}^\top$$

  and $l = 1$, we obtain the requested form by taking $\theta_{\text{lin}} = \begin{bmatrix} 0 & 1.5 & 0 \end{bmatrix}^\top$ and $\theta_{\text{nl}} = \begin{bmatrix} 0.3 & 0.2 \end{bmatrix}^\top$.
- *Linear parameter-varying systems.* We limit ourselves to the case of discrete-time linear parameter-varying systems. Furthermore, we assume the parameter values at each time instance are known. In that case, the system can be written in the following form:

  $$y(t) + \textstyle\sum_{i=1}^{n_{\text{a}}} \theta_{\text{y}i} a_i(p(t-i))y(t-i) = \sum_{i=0}^{n_{\text{b}}} \theta_{\text{u}i} b_i(p(t-i))u(t-i), \tag{4}$$

  $$a_i : \mathbb{P}^{n_{\text{p}}} \to \mathbb{R}^{n_1 \times n_y} \quad i = 1, \dots, n_{\text{a}},$$
  $$b_i : \mathbb{P}^{n_{\text{p}}} \to \mathbb{R}^{n_2 \times n_u} \quad i = 0, \dots, n_{\text{b}},$$
  $$\theta_{\text{y}} = \begin{bmatrix} \theta_{\text{y}1} & \dots & \theta_{\text{y}n_{\text{a}}} \end{bmatrix}^\top \in \mathbb{R}^{n_{\text{a}} n_1 \times n_y},$$
  $$\theta_{\text{u}} = \begin{bmatrix} \theta_{\text{u}0} & \dots & \theta_{\text{u}n_{\text{b}}} \end{bmatrix}^\top \in \mathbb{R}^{(n_{\text{b}}+1) n_2 \times n_y},$$

  where $n_{\text{a}}, n_{\text{b}} \in \mathbb{N}$, $\theta_{\text{y}i} \in \mathbb{R}^{n_y \times n_1}$, $\theta_{\text{u}i} \in \mathbb{R}^{n_y \times n_2}$, $p(t) \in \mathbb{P}^{n_{\text{p}}}$ are the known parameter values for all $t \in \mathbb{N}$, and $\mathbb{P}$ is the set of possible parameter values. This class of systems is already studied in the literature for the case with only affine dependence on the parameters, e.g. in [2] and [3]. Since we know $p(t)$ at each time instance, we can interpret this sequence as a discrete time dependent function. Making use of this, it is easy to see that

  $$\phi_{\text{nl}}(x_y, x_u, t) = \begin{bmatrix} a_1(p(t-1))y(t-1) \\ \vdots \\ a_{n_{\text{a}}}(p(t-n_{\text{a}}))y(t-n_{\text{a}}) \\ b_0(p(t))u(t) \\ \vdots \\ b_{n_{\text{b}}}(p(t-n_{\text{b}}))u(t-n_{\text{b}}) \end{bmatrix},$$

  and $\theta_{\text{nl}} = \begin{bmatrix} -\theta_{\text{y}}^\top & \theta_{\text{u}}^\top \end{bmatrix}^\top$, with $l \geq \max(n_{\text{a}}, n_{\text{b}})$ fulfils all requirements and leads to the correct equations.

A few classes of nonlinear systems are easily shown to be included in this model class. Affine systems can be included by taking $\phi_{\text{nl}}(x_y) = 1$. Wiener systems can be included by taking $\phi_{\text{nl}}(x_y)$ to be the nonlinear output mapping. LPV systems with known parameter values can be included as shown in the example above. Bilinear systems are linear in x_u given x_y and are as such a special case of OGB systems. Note that the discussion regarding the model class up to this point was of theoretical interest, where it is shown that these other classes can be rewritten as an OGB system assuming complete knowledge of the nonlinear terms. The next sections will start taking practical usability into account as well, albeit still in the deterministic setting.

C. LTI Embedding

Similar to [4], we introduce additional inputs $u_{\text{nl}}(t) \in \mathbb{R}^{n_{\text{nl}}}$ based on the nonlinear mapping $\phi_{\text{nl}}(x_y, x_u)$. To make use of the fact that the input-output representation of an LTI system with lag l depends on x_y and x_u, and as such on inputs up to l samples in the past, we choose the additional inputs such that

$$\mathscr{B}_{\text{ext}} := \left\{ w_{\text{ext}} = \begin{bmatrix} w \\ u_{\text{nl}} \end{bmatrix} \;\middle|\; y = \theta_{\text{lin}}^\top x_w + \theta_{\text{nl}}^\top x_{\text{nl}} \right\}, \tag{5}$$

where

$$x_{\text{nl}} = \begin{bmatrix} u_{\text{nl}}(t-l)^\top & u_{\text{nl}}(t-l+1)^\top & \dots & u_{\text{nl}}(t)^\top \end{bmatrix}^\top, \tag{6}$$
$$x_w = \begin{bmatrix} x_u^\top & x_y^\top \end{bmatrix}^\top,$$
$$\theta_{\text{nl}} \in \mathbb{R}^{(l+1)n_{\text{nl}} \times n_{\text{y}}},$$

is an LTI system with $n_u + n_{\text{nl}}$ inputs. From this point on we will refer to this LTI system as the *extended system*. To be consistent with (3) we re-scale θ_{nl} and form $u_{\text{nl}}(t)$ using the minimal nuber of functions such that $x_{\text{nl}}(t)$ contains all functions in $\phi_{\text{nl}}(x_y, x_u)$, such that $n_{\text{b}} \geq n_{\text{nl}}$. In practice we do not know the nonlinear terms exactly, making it difficult to use this model class. The solution is to choose a set of candidate basis functions, assumed to be linearly independent, such that linear combinations of these basis functions can

best represent the unknown nonlinear terms. The choice of this set allows the user to insert prior knowledge regarding the nonlinear terms, leading to a smaller problem to solve. Alternatively, a truncated set of universal basis functions, can be used in exchange for loosing exact representation in most cases. For the remainder of this work we will assume perfect representation of the nonlinear terms:

Assumption 1: Given an output-generalized bilinear system $\mathscr{B}$ and a proposed set of n_{nl} basis functions used as additional inputs $u_{\mathrm{nl}}(x_y,x_u)$, there exist parameter matrices θ_{lin} and θ_{nl} such that for each trajectory length $L \geq l$ and each trajectory with chosen input-output partitioning

$$w = \begin{bmatrix} u \\ y \end{bmatrix} \in \mathscr{B}|_L,$$

the equality

$$y = \theta_{\mathrm{lin}}^\top x_w + \theta_{\mathrm{nl}}^\top x_{\mathrm{nl}}(x_y,x_u),$$

with x_{nl} defined by (6) holds.

The solution quality in the case of non-perfect representation depends on how well the nonlinear terms can be represented given the set of basis functions. Depending on the choice of universal basis function there will be literature available discussing the maximal approximation error when truncating the number of basis functions.

The extended LTI system can make use of the results previously obtained for LTI systems. Of special interest is the data-driven representation, for which Lemma 2 is a sufficient condition. A point of attention is the persistency of excitation of the extended input. Since we no longer have complete freedom to choose these inputs, this limits which basis functions are allowed. A few guidelines for choosing a set of basis functions are given:

- The basis functions are linearly independent.
- No basis function is a linear combination of time-shifted (up to l times) versions of themselves or other basis functions.

The second guideline makes sure that no basis function is related to other basis functions in a way that would lead to rank deficiency in the extended input block Hankel matrix.

To formulate the main result of this section in a more readable fashion we need to formalize a partitioning of the matrices and vectors. Define the following partitioning of the extended trajectory block Hankel matrix:

$$\mathscr{H}_{L+T_{\mathrm{ini}}}(w_{\mathrm{ext}}) = \begin{bmatrix} \mathscr{H}_{L+T_{\mathrm{ini}}}(u_d) \\ \mathscr{H}_{L+T_{\mathrm{ini}}}(y_d) \\ \mathscr{H}_{L+T_{\mathrm{ini}}}(u_{\mathrm{nl}}) \end{bmatrix} = \begin{bmatrix} U \\ Y \\ U_{\mathrm{nl}} \end{bmatrix}.$$

Here $U \in \mathbb{R}^{n_u(L+T_{\mathrm{ini}})\times c}$, $Y \in \mathbb{R}^{n_y(L+T_{\mathrm{ini}})\times c}$, $U_{\mathrm{nl}} \in \mathbb{R}^{n_{\mathrm{b}}(L+T_{\mathrm{ini}})\times c}$, and $c = T-(L+T_{\mathrm{ini}})+1$. For ease of notation, these matrices will be further partitioned into past and future, which gives by making use of Lemma 1:

$$\mathscr{H}_{L+T_{\mathrm{ini}}}(w_{\mathrm{ext}})g = \begin{bmatrix} U \\ Y \\ U_{\mathrm{nl}} \end{bmatrix} g = \begin{bmatrix} U_p \\ U_f \\ Y_p \\ Y_f \\ U_{\mathrm{nl}} \end{bmatrix} g = \begin{bmatrix} \mathrm{col}(u_{\mathrm{ini}}) \\ \mathrm{col}(u_f) \\ \mathrm{col}(y_{\mathrm{ini}}) \\ \mathrm{col}(y_f) \\ \mathrm{col}(u_{\mathrm{nl}}) \end{bmatrix}. \quad (7)$$

Here

$$U_p \in \mathbb{R}^{n_u T_{\mathrm{ini}}\times c}, \qquad U_f \in \mathbb{R}^{n_u L\times c},$$
$$Y_p \in \mathbb{R}^{n_y T_{\mathrm{ini}}\times c}, \qquad Y_f \in \mathbb{R}^{n_y L\times c},$$

and

$$\begin{bmatrix} u \\ y \end{bmatrix} \in \mathscr{B}|_{L+T_{\mathrm{ini}}+l},$$

$$w = w_{\mathrm{ini}} \wedge w_f = \begin{bmatrix} u_{[l+1:]} \\ y_{[l+1:]} \\ u_{\mathrm{nl}} \end{bmatrix} = \begin{bmatrix} u_{\mathrm{ini}} \wedge u_f \\ y_{\mathrm{ini}} \wedge y_f \\ u_{\mathrm{nl}} \end{bmatrix} \in \mathscr{B}_{ext}|_{L+T_{\mathrm{ini}}}.$$

Due to $u_{\mathrm{nl}}(t)$ depending on $x_y(t)$ and $x_u(t)$, which require l samples in the past of both u and y, we require l additional samples in the past that are not used in the data-driven representation directly. For convenience we will start counting after the first l samples, such that for example $u(1)$ is the first input in (7), while the samples only used to calculate u_{nl} have indices $-l+1,\ldots,0$.

In the proof of the main theorem we will exploit the affine dependence on x_u of $\phi_{\mathrm{nl}}(x_y,x_u)$ given x_y and t. For this we offer a constructive proof similar to the proof of [4, Lemma 6] that allows us to write this nonlinear additional input in matrix form:

$$\mathrm{col}(u_{\mathrm{nl}}) = \Phi_{\mathrm{b},0}(y) + \Phi_{\mathrm{b},p}(y)\mathrm{col}(u_{\mathrm{ini}}) + \Phi_{\mathrm{b},f}(y)\mathrm{col}(u). \quad (8)$$

Lemma 3 (Matrix representation of u_{nl}): For an output-generalized bilinear system $\mathscr{B}$, and a finite signal $w \in (\mathbb{R}^q)^{T+l}$ indexed from $-l+1$ to T, with a given input-output partitioning and $T \in \mathbb{N}$, (8) holds with

$$\Phi_{\mathrm{b},0}(y) = \begin{bmatrix} u_{\mathrm{nl}}(x_y(1),\mathbf{0})^\top & \ldots & u_{\mathrm{nl}}(x_y(T),\mathbf{0})^\top \end{bmatrix}^\top,$$

$$\Phi_i(y) = \begin{bmatrix} u_{\mathrm{nl}}(x_y(1),x(\delta_i,1))^\top & \ldots & u_{\mathrm{nl}}(x_y(T),x(\delta_i,T))^\top \end{bmatrix}^\top,$$

$$\Phi_{\mathrm{b}}(y) = \begin{bmatrix} \Phi_1(y)-\Phi_{\mathrm{b},0}(y) & \ldots & \Phi_{n_u(T+l)}(y)-\Phi_{\mathrm{b},0}(y) \end{bmatrix},$$

with $\delta_i \in \mathbb{R}^{n_u(T+l)}$ the vector with all zeros except for a 1 at index i, $\Phi_{\mathrm{b},0} \in \mathbb{R}^{n_{\mathrm{b}}T}$, $\Phi_{\mathrm{i}} \in \mathbb{R}^{n_{\mathrm{b}}T}$, $\Phi_{\mathrm{b}} \in \mathbb{R}^{n_{\mathrm{b}}T\times n_u(T+l)}$, and

$$x(\delta_i,t) = \begin{bmatrix} \delta_i(t-l)^\top & \delta_i(t-l+1)^\top & \ldots & \delta_i(t)^\top \end{bmatrix}^\top.$$

With some abuse of notation we interpret $\delta_i \in \mathbb{R}^{n_u(T+l)}$ in the formula above as $\delta_i \in (\mathbb{R}^{n_u})^{T+l}$ (using the inverse transformation of col), allowing the vector to take the place of sequence u where needed.

Proof: Making use of the definition of $\mathrm{col}(u_{\mathrm{nl}})$ we can look at each time instance separately and stack the resulting matrices on top of each other. The most general form of u_{nl} that fulfills the affine dependence on x_u is

$$u_{\mathrm{nl}}(x_y(t),x_u(t),t) = \phi_{\mathrm{b},0}(x_y(t),t) + \theta^\top(\phi_{\mathrm{b}}(x_y(t),t) \otimes x_u(t)),$$

where

$$\phi_{\mathrm{b},0} : \mathbb{R}^{ln_y}\times\mathbb{Z} \to \mathbb{R}^{n_{\mathrm{b}}},$$
$$\phi_{\mathrm{b}} : \mathbb{R}^{ln_y}\times\mathbb{Z} \to \mathbb{R}^{n_{\mathrm{b}2}},$$

with $\theta \in \mathbb{R}^{n_{\mathrm{b}2}\times n_{\mathrm{b}}}$ and $n_{\mathrm{b}2} \in \mathbb{N}$. Fixing time t, it is evident to see that taking all $u=0$ results in $\phi_{\mathrm{nl}}(x_y,\mathbf{0}) = \phi_{\mathrm{b},0}(x_y)$, since that is the only term remaining in that case. Stacking these

$\phi_{\mathrm{b},0}$ for the different time values results in $\Phi_{\mathrm{b},0}(y)$. Taking a look at $\theta^\top(\phi_{\mathrm{b}}(x_y)\otimes x_u)$, we see that this term is linear in x_u given θ and x_y. We can extract the coefficients of each input for each row by setting that specific input to one and the rest to zero, in other words using δ_i as input. We find Φ_{b} by noticing that $\theta^\top(\phi_{\mathrm{b}}(x_y)\otimes x_u) = u_{\mathrm{nl}}(x_y,x_u) - \phi_{\mathrm{b},0}(x_y)$ and stacking the resulting matrices. Partitioning Φ_{b} into past (corresponding to the initial input) and future (corresponding to the future input) immediately leads to (8). ■

IV. Output matching

We now have the following theorem:

Theorem 1 (data-driven output matching): Consider an output-generalized bilinear system $\mathscr{B}$, proposed set of n_{nl} linearly independent basis functions used as additional inputs $u_{\mathrm{nl}}(x_y,x_u)$, data $w_{\mathrm{d}} \in \mathscr{B}|_{T+l}$, initial trajectory $w_{\mathrm{ini}} = \begin{bmatrix} u_{\mathrm{ini}}^\top & y_{\mathrm{ini}}^\top \end{bmatrix}^\top \in (\mathbb{R}^{n_{\mathrm{u}}+n_{\mathrm{y}}})^{T_{\mathrm{ini}}+l}$, and a reference output $y_{\mathrm{r}} \in (\mathbb{R}^{n_y})^L$ generated by the system from w_{ini}. Write the truncated trajectories $u'_{\mathrm{ini}} = u_{\mathrm{ini}[l+1:]}$, $y'_{\mathrm{ini}} = y_{\mathrm{ini}[l+1:]}$, and $w'_{\mathrm{ini}} = w_{\mathrm{ini}[l+1:]}$, and write $w_{\mathrm{d,ext}} \in (\mathbb{R}^{n_y+n_u+n_{\mathrm{nl}}})^T$ as the data trajectory of the extended system from w_{d}. Under the assumption that

$$\operatorname{rank} \mathscr{H}_{L+T_{\mathrm{ini}}}(w_{\mathrm{d,ext}}) = (n_u+n_{\mathrm{nl}})(L+T_{\mathrm{ini}})+n, \tag{9}$$

with n the order of the extended system, and Assumption 1, a solution to Problem 1 is given by

$$\operatorname{col}(u) = U_{\mathrm{f}} g, \tag{10}$$

where g is a solution to the linear system of equations

$$\mathscr{A}(w_d)g = \begin{bmatrix} \operatorname{col}(u'_{\mathrm{ini}}) \\ \operatorname{col}(y'_{\mathrm{ini}} \wedge y_{\mathrm{r}}) \\ \Phi_{\mathrm{b,p}}(x_y)\operatorname{col}(u'_{\mathrm{ini}}) + \Phi_{\mathrm{b},0}(x_y) \end{bmatrix}$$

with

$$\mathscr{A}(w_{\mathrm{d}}) = \begin{bmatrix} U_{\mathrm{p}} \\ Y \\ U_{\mathrm{nl}} - \Phi_{\mathrm{b,f}}(x_y)U_{\mathrm{f}} \end{bmatrix}.$$

Proof: Making use of the set of proposed basis functions we can construct U_{nl}, $\Phi_{\mathrm{b},0}$, $\Phi_{\mathrm{b,p}}$, and $\Phi_{\mathrm{b,f}}$. Under rank condition (9) we have, according to Lemma 1, that there exists a g such that

$$\mathscr{H}_{L+T_{\mathrm{ini}}}(w_{\mathrm{d,ext}})g = \begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_p \\ Y_f \\ U_{\mathrm{nl}} \end{bmatrix} g = \begin{bmatrix} \operatorname{col}(u'_{\mathrm{ini}}) \\ \operatorname{col}(u) \\ \operatorname{col}(y'_{\mathrm{ini}}) \\ \operatorname{col}(y_{\mathrm{r}}) \\ \operatorname{col}(u_{\mathrm{nl}}) \end{bmatrix}. \tag{11}$$

Plugging the equation $\operatorname{col}(u) = U_{\mathrm{f}} g$ into (8), and plugging in the resulting expression for u_{nl} in (11), we can move the term depending on g to the other side, which gives, putting aside the equations with unknown solutions:

$$\begin{bmatrix} U_{\mathrm{p}} \\ Y \\ U_{\mathrm{nl}} - \Phi_{\mathrm{b},f}(x_y)U_{\mathrm{f}} \end{bmatrix} g = \begin{bmatrix} \operatorname{col}(u_{\mathrm{ini}}) \\ \operatorname{col}(y'_{\mathrm{ini}} \wedge y_{\mathrm{r}}) \\ \Phi_{\mathrm{b},p}(x_y)\operatorname{col}(u'_{\mathrm{ini}}) + \Phi_{\mathrm{b},0}(x_y) \end{bmatrix}. \tag{12}$$

This linear system of equations has at least one solution, the g vector from (11). Any solution g to the linear system of equations (12) will lead to an input u according to (10) that is a solution to Problem 1 for the extended system. Under Assumption 1 and with noise-free data, the extended system is equivalent to the original OGB system $\mathscr{B}$ by construction, such that u is also a solution to Problem 1 for $\mathscr{B}$. ■

With this theorem, it is shown that it is possible to solve the output matching problem exactly for this specific class of nonlinear systems in the deterministic case. It should be emphasized that this theorem offers *a* solution, not *the* solution. The solution to the output matching problem is not unique in general, for example in the case of non-invertible systems. It is possible to obtain a parameterization of the solution space, allowing the user to choose the best solution according to a chosen criterion. The conditions for Theorem 1 are sufficient, but necessity is not shown.

V. Simulation examples

This section discusses the results obtained by simulating a nonlinear system obtained by discretizing differential equations. A data trajectory w_{d} of length T is generated by applying a random input signal taken from a given distribution. This data trajectory is used to create the matrices needed by the method. Since we created the system ourselves we know its properties, including the lag and order, which we use to verify if the rank condition (9) is fulfilled. The quality of the result is quantified by applying the estimated input obtained by Theorem 1 and comparing the realized output obtained by applying the estimated input to the system with the reference output. The quantification is done using the realized normalized root mean squared error (RRMSE)

$$\mathrm{RRMSE}(y_{\mathrm{realised}}) = \frac{\|y_{\mathrm{realized}} - y_{\mathrm{r}}\|_2}{\|y_{\mathrm{r}}\|_2}. \tag{13}$$

A. Simulation example: Four-tank system

This example is taken from [16]. It consists of four tanks with two tanks each flowing into the one below. Two pumps, one for each top tank, pump water from the bottom basin to the tank. We take the water level in each tank as the outputs and the pump speeds as the inputs. Discretized using a first-order Euler approximation of the derivative with sample time T_s we obtain the following system:

$$\begin{aligned}
y_1(t) &= y_1(t-1) - T_s\Big(\frac{a_1}{A_1}\sqrt{2gy_1(t-1)} + \frac{a_3}{A_1}\sqrt{2gy_3(t-1)} \\
&\quad + \frac{\gamma_1 k_1}{A_1}u_1(t-1)\Big) \\
y_2(t) &= y_2(t-1) - T_s\Big(\frac{a_2}{A_2}\sqrt{2gy_2(t-1)} + \frac{a_4}{A_2}\sqrt{2gy_4(t-1)} \\
&\quad + \frac{\gamma_2 k_2}{A_2}u_2(t-1)\Big) \\
y_3(t) &= y_3(t-1) - T_s\Big(\tfrac{a_3}{A_3}\sqrt{2gy_3(t-1)} + \tfrac{(1-\gamma_2)k_2}{A_3}u_2(t-1)\Big) \\
y_4(t) &= y_4(t-1) - T_s\Big(\tfrac{a_4}{A_4}\sqrt{2gy_4(t-1)} + \tfrac{(1-\gamma_1)k_1}{A_4}u_1(t-1)\Big).
\end{aligned}$$

Here we take

$$u_{\text{nl}}(t) = \sqrt{y(t)},$$

applied to each element, which leads to 4 additional inputs. The parameter values used in the simulation are $a_i = 2.3\text{cm}^2$, $A_i = 730\ \text{cm}^2$, $k_1 = 5.51\ \text{cm}^3/\text{s}$, $k_2 = 6.58\ \text{cm}^3/\text{s}$, $g = 981\ \text{cm}/\text{s}^2$, $\gamma_1 = 0.333$ and $\gamma_2 = 0.307$. We use inputs $u \in (\mathbb{R}^2)^{1410}$ with the first two inputs zero and the remainder iid taken uniformly from the interval $]0, 0.05[$ to generate the data trajectory. The input estimation is done by applying Theorem 1 once. We take $L = 200$ and $T_{\text{ini}} = 1$. Taking a data trajectory length of 1410 is done because this is the minimal amount of data needed to fulfill rank condition (9). This system will not have a unique solution to the output matching problem. Since the output at the current time does not depend on the current input, the final inputs do not influence the output within our horizon, and can as such take any values while still being a solution to the output matching problem. We calculate the RRMSE value using different lengths of data to show that we can do with fewer samples due to this. Looking at the system definition, it is expected that we can do with 6 samples less, corresponding to $u(L)$ and $u_{\text{nl}}(L)$, since we do not care about the last 6 unknowns and as such need 6 equations less. The result can be seen in Figure 1. Up to the expected number of samples, the proposed method gives exact reconstruction of the output. This does not change by increasing the past horizon T_{ini} or L. This shows that even without fulfilling all conditions for Theorem 1 (in this case the generalized persistency of excitation condition is no longer met due to a lack of columns) it is still possible to find a solution, meaning that the sufficient conditions are not all necessary conditions for the existence of a correct solution using the given method.

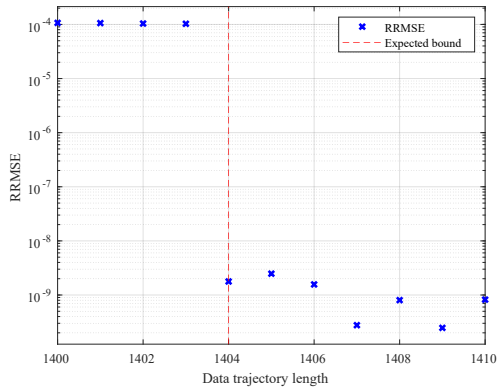

Fig. 1. Logarithmic plot of the RRMSE for different data trajectory lengths. Perfect reconstruction is predicted to last until a length of $1410 - 6 = 1404$.

VI. Conclusion and Future Work

This work has shown the feasibility of using data-driven simulation techniques to solve the output matching problem. It turns out that no constraints have to be added compared to the simulation case. It is shown that linear parameter-varying systems are a specific case of output-generalized bilinear systems, allowing the results of this work to also be applied to this class of systems.

Preliminary work has been done, but improvements are still possible. The next step is making the proposed method more robust to measurement noise on the trajectories. Other than that, the feasibility of adapting this method for the reference tracking control problem in the general case has to be analyzed. An analysis of the noise distributions arising from popular choices of nonlinear basis functions is interesting to look at, offering insights that could be used to develop more optimal noise solutions. Something else to look at is the complexity of solving the structure estimation problem.

References

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

[2] C. Verhoek, R. Tóth, S. Haesaert, and A. Koch, "Fundamental Lemma for Data-Driven Analysis of Linear Parameter-Varying Systems," in *2021 60th IEEE Conference on Decision and Control (CDC)*, Dec. 2021, pp. 5040–5046, iSSN: 2576-2370.

[3] C. Verhoek, H. S. Abbas, R. Tóth, and S. Haesaert, "Data-Driven Predictive Control for Linear Parameter-Varying Systems," *IFAC-PapersOnLine*, vol. 54, no. 8, pp. 101–108, Jan. 2021.

[4] I. Markovsky, "Data-driven simulation of nonlinear systems via linear time-invariant embedding," *IEEE Trans. Automat. Contr.*, 2023.

[5] J. G. Rueda-Escobedo and J. Schiffer, "Data-Driven Internal Model Control of Second-Order Discrete Volterra Systems," in *2020 59th IEEE Conference on Decision and Control (CDC)*, Dec. 2020, pp. 4572–4579, iSSN: 2576-2370.

[6] M. Alsalti, V. G. Lopez, J. Berberich, F. Allgöwer, and M. A. Müller, "Data-based Control of Feedback Linearizable Systems," Apr. 2022, arXiv:2204.01148 [cs, eess, math].

[7] M. Alsalti, J. Berberich, V. G. Lopez, F. Allgöwer, and M. A. Müller, "Data-Based System Analysis and Control of Flat Nonlinear Systems," in *2021 60th IEEE Conference on Decision and Control (CDC)*, Dec. 2021, pp. 1484–1489, iSSN: 2576-2370.

[8] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Linear Tracking MPC for Nonlinear Systems—Part II: The Data-Driven Case," *IEEE Transactions on Automatic Control*, vol. 67, no. 9, pp. 4406–4421, Sept. 2022, conference Name: IEEE Transactions on Automatic Control.

[9] L. Huang, J. Lygeros, and F. Dörfler, "Robust and Kernelized Data-Enabled Predictive Control for Nonlinear Systems," June 2022, arXiv:2206.01866 [cs, eess].

[10] M. Tanaskovic, L. Fagiano, C. Novara, and M. Morari, "Data-driven control of nonlinear systems: An on-line direct approach," *Automatica*, vol. 75, pp. 1–10, Jan. 2017.

[11] H. Arbabi, M. Korda, and I. Mezić, "A Data-Driven Koopman Model Predictive Control Framework for Nonlinear Partial Differential Equations," in *2018 IEEE Conference on Decision and Control (CDC)*, Dec. 2018, pp. 6409–6414, iSSN: 2576-2370.

[12] Y. Lian, R. Wang, and C. N. Jones, "Koopman based data-driven predictive control," Feb. 2021, arXiv:2102.05122 [cs, eess, math].

[13] I. Markovsky and P. Rapisarda, "Data-driven simulation and control," *International Journal of Control*, vol. 81, no. 12, pp. 1946–1959, Dec. 2008.

[14] J. C. Willems, "Paradigms and puzzles in the theory of dynamical systems," *IEEE Transactions on automatic control*, vol. 36, no. 3, pp. 259–294, 1991.

[15] I. Markovsky, E. Prieto-Araujo, and F. Dörfler, "On the persistency of excitation," *Automatica*, vol. 147, p. 110657, Jan. 2023.

[16] E. P. Gatzke, E. S. Meadows, C. Wang, and F. J. Doyle, "Model based control of a four-tank system," *Computers & Chemical Engineering*, vol. 24, no. 2, pp. 1503–1509, July 2000.