Data-Driven Simulation for NARX Systems*

Vikas K. Mishra   Ivan Markovsky   Antonio Fazzi
Department ELEC
Vrije Universiteit Brussel (VUB)
Pleinlaan 2, 1050 Brussels, Belgium
{vmishra, imarkovs, afazzi}@vub.be

Philippe Dreesen
Dept. Electrical Engineering (ESAT), STADIUS
KU Leuven
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
philippe.dreesen@gmail.com

***Abstract*—Nonlinear phenomena can be represented as nonlinear autoregressive exogenous (NARX) systems. NARX systems can be seen as a nonlinear version of linear infinite impulse response filter. Data-driven approaches are witnessing considerable interests in recent times and they are well-flourished for linear time-invariant systems. However, for nonlinear systems, they are still limited and attempts have been made to generalize the results for linear systems to nonlinear systems. In this paper, we study the problem of data-driven simulation for NARX systems: compute the output trajectory from a given input trajectory and initial conditions without explicitly identifying the parametric model; the model is implicitly identified by observed trajectory. Next, we develop an algorithm for the implementation of our approach. Finally, we illustrate the developed algorithm by numerical experiments.**

***Index Terms*—Data-driven simulation, NARX, System identification**

I. Introduction

NARX (nonlinear autoregressive exogenous) systems are popular in modeling and simulation of nonlinear processes [1]. They can be seen as a nonlinear generalization of ARX (autoregressive exogenous), a technique for linear black-box system identification [2]. NARX models cover a large class of nonlinear dynamic systems, such as Gaussian processes [3], [4] and neural networks [5], [6], widely used in applications. The fact that the output depends on its previous values (the autoregressive property) makes the NARX model a good predictor of time series [7]–[11].

Here we are interested in data-driven simulation for NARX systems. This is opposed to the classical model-based simulation problem, where a model is first identified in order to predict the output. As argued in the literature [12], this intermediate step of identifying a model may return a suboptimal solution of the desired problem. Hence it would be better to avoid this intermediate step of identifying a model and work directly with the available data (which implicitly identify the model). This leads to the problem of data-driven simulation: given an observed trajectory $w_d := \text{col}(u_d, y_d) := \left[\begin{smallmatrix} u_d \\ y_d \end{smallmatrix}\right]$, initial condition $w_{\text{ini}} := \text{col}(u_{\text{ini}}, y_{\text{ini}})$, and input trajectory u_s; compute the output trajectory y_s.

In recent times, Theorem 1 of [13] or its variants have received considerable attention among the researchers in systems and control due to its applications in data-driven simulation and control [14]–[17]. This result is popularly known as the *fundamental lemma* in system theory [18]. We also adopt this nomenclature in our present paper. Originally, the fundamental lemma is for LTI (linear time-invariant) systems. It essentially provides conditions under which trajectories of an LTI system can be expressed as a linear combination of columns of the Hankel matrix, which is built from an observed trajectory. Thus, it gives a trajectory-based representation of a model, *i.e.* the model is described independent of the model parameters. This fact is the crux while utilizing the fundamental lemma in studying data-driven control problems. Recently, there has been a great deal of interest in generalizing the fundamental lemma; these generalizations are in two directions: 1) it has been shown in [19] that the fundamental lemma like result is possible even if we replace a (long) trajectory by multiple (short) trajectories by introducing the notion of *collectively persistency of excitation*; 2) it has been extended to a few classes of nonlinear systems: Hammerstein and Wiener systems [20], and second order Volterra systems [21]. Here, we are also interested in generalizing this in the second direction, particularly for NARX systems. It is notable that none of the previous extensions consider nonlinearity in the outputs.

This paper is motivated by a recent work [21], where the fundamental lemma is generalized to a second order Volterra system. It is then natural to ask if this result can be further extended to a more general class of systems. In this paper, we have extended the result for NARX systems by modifying the definition of the persistency of excitation, which not only involves the inputs, but also the observed outputs. Moreover, we have utilized this result in developing an algorithm for data-driven simulation. Specifically, our contributions are twofold: 1) we have extended the fundamental lemma for LTI systems to NARX systems, and 2) we have developed an algorithm for data-driven simulation for NARX systems.

A. Related Work

Recently, control theory has received considerable attention among researchers from different domains because of the new methods developed in view of data-driven analysis and control.

VKM, AF, and IM are funded by ERC Grant 258581, FWO projects G028015N and G090117N; and FNRS–FWO EOS Project 30468160. PD is affiliated to Leuven.AI-KU Leuven institute for AI, Leuven, Belgium. He is funded by KU Leuven Research fund; FWO Vlaanderen (FNRS–FWO EOS Project 30468160, SBO project S005319N, Infrastructure project I013218N, TBM project T001919N, SB/1SA1319N, SB/1S93918, SB/151622), Flemish Government (AI Research Program), and ERC-AdG grant 885682.

Perhaps the main ingredient in developing these methods is a seminal result by Willems et al. [13, Theorem 1] in system theory, which is also known as the fundamental lemma. The result has been recently generalized to the situations where a number of short system trajectories are given instead of a single long trajectory [19], Hammerstein and Wiener systems [20], and second order Volterra systems [21]. It seems like the first contribution towards data-driven simulation and control for LTI systems using the fundamental lemma is reported in [14]. Later, an algorithm is developed for closed-loop data-driven simulation in [22]. Recently, data-driven simulation problem for LTI systems has been studied as the Hankel matrix completion problem in [17]. Here, we are interested in extending the work of [21] for the purpose of data-driven simulation for NARX systems.

In the last years, data-driven problems are becoming more popular as they avoid identifying the explicit analytical model [12]. Looking at the literature in this framework, we can list as examples the solution of minimum energy control for discrete-time LTI systems by using experimental data [23], the notion of data informativity for control problems [24], [25], data-driven model predictive control [26], data-driven open and closed loop representations of LTI discrete-time systems [16], data-driven dissipativity [27], data-driven controllability and observability [28]–[30]. Further, data-driven approaches to control have already found applications in real-time control of electrical circuits [31] and real-time implementation of fault-tolerant control systems [32].

B. Problem Description

Consider the following NARX system

$$y(t)=\sum_{i=1}^{\mathtt{d}_1}\sum_{j=0}^{\ell_1}\alpha_{ij}u^i(t-j)+\sum_{k=1}^{\mathtt{d}_2}\sum_{l=1}^{\ell_2}\beta_{kp}y^k(t-l) \tag{1}$$

where all α_{ij}, β_{kl} are real parameters, $u(t)\in\mathbb{R}$ is the input, and $y(t)\in\mathbb{R}$ is the output. For convenience, the above system is rewritten in a more compact form as

$$y(t)=\sum_{i=1}^{\mathtt{d}_1}\Gamma_i^\top\eta^i(t)+\sum_{k=1}^{\mathtt{d}_2}\Lambda_k^\top\xi^k(t), \tag{2}$$

where

$$\begin{aligned}\eta^i(t)&=\begin{bmatrix}u^i(t) & u^i(t-1) & \dots & u^i(t-\ell_1)\end{bmatrix}^\top\\ \xi^k(t)&=\begin{bmatrix}y^k(t-1) & y^k(t-2) & \dots & y^k(t-\ell_2)\end{bmatrix}^\top\\ \Gamma_i&=\begin{bmatrix}\alpha_{i0} & \alpha_{i1} & \dots & \alpha_{i\ell_1}\end{bmatrix}^\top\\ \Lambda_k&=\begin{bmatrix}\beta_{k1} & \beta_{k2} & \dots & \beta_{k\ell_2}\end{bmatrix}^\top.\end{aligned} \tag{3}$$

Note that when we leave out the superscript in η^i or ξ^k, it should be understood as 1. Systems of the form (1) are known as additive NARX in the literature and have been efficiently applied to several industrial processes [1]. In order to keep our exposition simple, we do not involve the cross terms in (1). Nevertheless, the results of the paper can be extended to a system with cross terms analogously. See Appendix A for outline of the approach. To this end, define $\ell:=\max(\ell_1,\ell_2)$.

Recall the problem of simulation: computing the outputs from inputs and initial conditions. A naive approach to deal with this problem is to use the linear regression and compute all the parameters α_{ij}, β_{kl} and then use system (1) to recursively compute the future outputs from inputs, and previous (delayed) inputs and outputs (initial conditions). This is what we call model-based simulation. However, as discussed earlier, here we are interested in an alternative approach, *viz.* data-driven simulation, which is typically defined as follows: given an observed input/output trajectory or data $w_d=\mathrm{col}(u_d,y_d)$, an input trajectory u_s, and initial conditions w_{ini}; find the output trajectory y_s. Note that the system is identified implicitly by the observed input/output trajectory $w_d=\mathrm{col}(u_d,y_d)$. Note moreover that in data-driven simulation, the output trajectory y_s is computed in one-shot contrary to the model-based simulation, where it is computed recursively. In order to tackle this problem, we first extend the fundamental lemma for LTI systems to NARX systems (Theorem 1) and then develop an algorithm based on this result (Algorithm 1).

C. Outline of the Paper

The next section collects our notation and develops preliminary concepts needed for the subsequent development of the paper. The extension of the fundamental lemma to NARX system is provided in Section III and an algorithm for data-driven simulation based on this is then developed in Section IV. Numerical experiments demonstrating the effectiveness of the developed algorithm are performed in Section V. Finally, conclusions of the paper are offered in the last section.

II. Notation and Preliminaries

For any matrix A, $A^\top$ denotes its transpose. For any two matrices A and B, their Kronecker product is denoted by $A\otimes B$. If matrices $A_1,A_2,\dots,A_r$ have the same number of columns, we define

$$\mathrm{col}(A_1,A_2,\dots,A_r):=\begin{bmatrix}A_1^\top & A_2^\top & \dots & A_r^\top\end{bmatrix}^\top.$$

By w_d (subscript 'd' is used to denote the observed data throughout the paper) we denote an observed vector time series of length T defined as

$$w_d:=\mathrm{col}\big(w_d(1),w_d(2),\dots,w_d(T)\big)\in\mathbb{R}^{qT},$$

where $w_d(i)\in\mathbb{R}^q$ for $i=1,2,\dots,T$.

Also, for any general trajectory $f:[1,T]\to\mathbb{R}^q$, we use $f|_T$ to denote

$$\mathrm{col}\big(f(1),f(2),\dots,f(T)\big).$$

Definition 1. *A time series w_d is persistently exciting of order $L\in\mathbb{N}$ if it is the maximum natural number for which the associated Hankel matrix*

$$\mathcal{H}_L(w_d)=\begin{bmatrix}w_d(1) & w_d(2) & \cdots & w_d(T-L+1)\\ w_d(2) & w_d(3) & \cdots & w_d(T-L+2)\\ \vdots & \vdots & \ddots & \vdots\\ w_d(L) & w_d(L+1) & \cdots & w_d(T)\end{bmatrix}$$

is of full row rank, i.e. qL.

We now generalize this definition to the case of the NARX system (1).

Definition 2. *A sequence $\{w_d, w_d^2, \ldots, w_d^{\mathtt{d}}\}$, where each element is a time series, of degree $\mathtt{d}$ is persistently exciting of order $L \in \mathbb{N}$ if it is the maximum natural number for which the mosaic-Hankel matrix (each block is the Hankel matrix)*

$$\mathcal{H}_L^{\mathtt{d}}(w_d) := \mathit{col}\left(\mathcal{H}_L(w_d), \mathcal{H}_L(w_d^2), \ldots, \mathcal{H}_L(w_d^{\mathtt{d}})\right)$$

is of full row rank, i.e. $\mathtt{d}qL$.

III. Generalization of the Fundamental Lemma to NARX Systems

Following [21], we state and prove the following result for NARX systems of the form (1).

Theorem 1. *Let $w_d = \mathit{col}(u_d, y_d)$ be an observed trajectory of system* (1)*. Let $\{\eta_d, \eta_d^2, \ldots, \eta_d^{\mathtt{d}_1}, \xi_d, \xi_d^2, \ldots, \xi_d^{\mathtt{d}_2}\}$ be persistently exciting of order L. Then, any $w|_L = \mathit{col}(u, y)|_L$ is a trajectory of* (1) *if and only if there exists $g \in \mathbb{R}^{T-L+1}$ such that*

$$\begin{bmatrix} \eta|_L \\ \eta^2|_L \\ \vdots \\ \eta^{\mathtt{d}_1}|_L \\ \xi|_L \\ \xi^2|_L \\ \vdots \\ \xi^{\mathtt{d}_2}|_L \\ y|_L \end{bmatrix} = \begin{bmatrix} \mathcal{H}_L(\eta_d) \\ \mathcal{H}_L(\eta_d^2) \\ \vdots \\ \mathcal{H}_L(\eta_d^{\mathtt{d}_1}) \\ \mathcal{H}_L(\xi_d) \\ \mathcal{H}_L(\xi_d^2) \\ \vdots \\ \mathcal{H}_L(\xi_d^{\mathtt{d}_2}) \\ \mathcal{H}_L(y_d) \end{bmatrix} g. \tag{4}$$

Proof. From (2) and (3), we have

$$\begin{bmatrix} \eta|_L \\ \eta^2|_L \\ \vdots \\ \eta^{\mathtt{d}_1}|_L \\ \xi|_L \\ \xi^2|_L \\ \vdots \\ \xi^{\mathtt{d}_2}|_L \\ y|_L \end{bmatrix} = \begin{bmatrix} \mathrm{diag}(I, I, \ldots, I) & 0 \\ 0 & \mathrm{diag}(I, I, \ldots, I) \\ \mathcal{P} & \mathcal{Q} \end{bmatrix} \begin{bmatrix} \eta|_L \\ \eta^2|_L \\ \vdots \\ \eta^{\mathtt{d}_1}|_L \\ \hline \xi|_L \\ \xi^2|_L \\ \vdots \\ \xi^{\mathtt{d}_2}|_L \end{bmatrix} \tag{5}$$

with

$$\begin{aligned} \mathcal{P} &= \begin{bmatrix} I \otimes \Gamma_1^\top & I \otimes \Gamma_2^\top & \ldots & I \otimes \Gamma_{\mathtt{d}_1}^\top \end{bmatrix} \quad \text{and} \\ \mathcal{Q} &= \begin{bmatrix} I \otimes \Lambda_1^\top & I \otimes \Lambda_2^\top & \ldots & I \otimes \Lambda_{\mathtt{d}_2}^\top \end{bmatrix}. \end{aligned}$$

Because $\{\eta_d, \eta_d^2, \ldots, \eta_d^{\mathtt{d}_1}, \xi_d, \xi_d^2, \ldots, \xi_d^{\mathtt{d}_2}\}$ is persistently exciting of order L, matrix $\mathrm{col}\big(\mathcal{H}_L^{\mathtt{d}_1}(\eta_d), \mathcal{H}_L^{\mathtt{d}_2}(\xi_d)\big)$ is of full row rank. That is,

$$\begin{aligned} \mathrm{rank} \begin{bmatrix} \mathcal{H}_L(\eta_d) \\ \mathcal{H}_L(\eta_d^2) \\ \vdots \\ \mathcal{H}_L(\eta_d^{\mathtt{d}_1}) \\ \mathcal{H}_L(\xi_d) \\ \mathcal{H}_L(\xi_d^2) \\ \vdots \\ \mathcal{H}_L(\xi_d^{\mathtt{d}_2}) \end{bmatrix} &= \mathrm{rank} \left[\begin{array}{c|c} \mathcal{H}_L(\eta_d) & \eta|_L \\ \mathcal{H}_L(\eta_d^2) & \eta^2|_L \\ \vdots & \vdots \\ \mathcal{H}_L(\eta_d^{\mathtt{d}_1}) & \eta^{\mathtt{d}_1}|_L \\ \mathcal{H}_L(\xi_d) & \xi|_L \\ \mathcal{H}_L(\xi_d^2) & \xi^2|_L \\ \vdots & \vdots \\ \mathcal{H}_L(\xi_d^{\mathtt{d}_2}) & \xi^{\mathtt{d}_2}|_L \end{array}\right] \\ &= L\mathtt{d}_1(\ell_1 + 1) + L\mathtt{d}_2\ell_2. \end{aligned}$$

This is equivalent to saying that there exists $g \in \mathbb{R}^{T-L+1}$ such that

$$\begin{bmatrix} \eta|_L \\ \eta^2|_L \\ \vdots \\ \eta^{\mathtt{d}_1}|_L \\ \xi|_L \\ \xi^2|_L \\ \vdots \\ \xi^{\mathtt{d}_2}|_L \end{bmatrix} = \begin{bmatrix} \mathcal{H}_L(\eta_d) \\ \mathcal{H}_L(\eta_d^2) \\ \vdots \\ \mathcal{H}_L(\eta_d^{\mathtt{d}_1}) \\ \mathcal{H}_L(\xi_d) \\ \mathcal{H}_L(\xi_d^2) \\ \vdots \\ \mathcal{H}_L(\xi_d^{\mathtt{d}_2}) \end{bmatrix} g. \tag{6}$$

Substituting (6) in (5), we obtain (4). This completes the proof of the theorem. □

IV. Data-Driven Simulation

Based on Theorem 1 and the approach discussed in [14] for LTI systems, we develop the following algorithm for data-driven simulation.

Algorithm 1. Data-driven simulation for NARX system (1)

Input: *Observed trajectory $w_d = \mathit{col}(u_d, y_d)$, initial condition $w_{ini} = \mathit{col}(u_{ini}, y_{ini})$, input trajectory $u|_\tau = \mathit{col}\big(u(1), \ldots, u(\tau)\big)$, and integers $\mathtt{d}_1, \mathtt{d}_2, \ell_1, \ell_2$.*

1: *Define: $U^i := \mathcal{H}_{t_{ini}+\tau}(\eta_d^i)$ for $1 \leq i \leq \mathtt{d}_1$, $V^k := \mathcal{H}_{t_{ini}+\tau}(\xi_d^k)$ for $1 \leq k \leq \mathtt{d}_2$, and $Y := \mathcal{H}_{t_{ini}+\tau}(y_d)$.*

2: *Partition: $U^i := \begin{bmatrix} U_p^i \\ U_f^i \end{bmatrix}$, $V^k := \begin{bmatrix} V_p^k \\ V_f^k \end{bmatrix}$, and $Y := \begin{bmatrix} Y_p \\ Y_f \end{bmatrix}$.*

3: *Construct: η_{ini}^i and ξ_{ini}^k from w_{ini}, and $\eta^i|_\tau$ from $u|_\tau$ compatible with Step 2.*

4: *Compute g and $y|_\tau$ from*

$$\begin{bmatrix} U_p^1 \\ \vdots \\ U_p^{\mathsf{d}_1} \\ V_p^1 \\ \vdots \\ V_p^{\mathsf{d}_2} \\ Y_p \\ U_f^1 \\ \vdots \\ U_f^{\mathsf{d}_1} \\ V_f^1 \\ \vdots \\ V_f^{\mathsf{d}_2} \\ Y_f \end{bmatrix} g = \begin{array}{c} \begin{bmatrix} \eta_{ini} \\ \vdots \\ \eta_{ini}^{\mathsf{d}_1} \\ \xi_{ini} \\ \vdots \\ \xi_{ini}^{\mathsf{d}_2} \\ y_{ini} \\ \eta|_\tau \\ \vdots \\ \eta^{\mathsf{d}_1}|_\tau \\ \xi|_\tau \\ \vdots \\ \xi^{\mathsf{d}_2}|_\tau \\ y|_\tau \end{bmatrix} \end{array} \begin{array}{c} (\ell_1+1)t_{ini} \\ \\ (\ell_1+1)t_{ini} \\ \ell_2 t_{ini} \\ \\ \ell_2 t_{ini} \\ t_{ini} \\ (\ell_1+1)\tau \\ \\ (\ell_1+1)\tau \\ \ell_2\tau \\ \\ \ell_2\tau \\ \tau \end{array} . \tag{7}$$

Output: *Output trajectory $y|_\tau = col\big(y(1), \dots, y(\tau)\big)$.*

Proposition 1. *Under the assumptions of Theorem 1, Algorithm 1 computes the correct output trajectory y_s for given input trajectory u_s and initial condition w_{ini}.*

Proof. Let $L = t_{\text{ini}} + \tau$ in Eq. (4). Assume that an initial trajectory w_{ini} is given and we want to compute a $\tau-$samples long output trajectory y_s for a given input trajectory u_s. From Theorem 1, the concatenation $w_{\text{ini}} \wedge \text{col}(u_s, y_s) =: \bar{w}$ is a trajectory of system (1) if and only if there exists a g such that Eq. (4) holds. In view of Steps 1–3 of Algorithm 1, Eq. (7) is equivalent to Eq. (4). This completes the proof. ☐

Remark 1 (On initial conditions)**.** *The initial condition $w_{ini} = col(u_{ini}, y_{ini})$ is specified by $(\ell+1)-$long sequences of inputs and outputs. Thereby, we can determine η_{ini}^i and ξ_{ini}^k for each i and for each k. Note that in the instances where w_{ini} is not prescribed, it is by default specified by zero initial condition.*

Note that Step 4 of Algorithm 1 is practically intractable. Indeed, due to the presence of monomials in $\xi|_\tau$, we could not be able to implement the algorithm and is a topic of future research. Nevertheless, we are able to implement the algorithm for an important particular case in which monomials in $\xi|_\tau$ and thus ξ_{ini} are not present, which is to say that current output does not depend on previous outputs. Remark that this particular case is a subject of [21]; however, the problem of data-driven simulation has not been studied therein. In this case (after removing the block equations corresponding to monomials in $\xi|_\tau$ and ξ_{ini}), the remaining block equations, except the last block equation $Y_f g = y|_\tau$, put restrictions on g, *i.e.* are used to determine g, and the last block equation is used to compute the output trajectory $y|_\tau$. Typically, g is nonunique. However, taking a particular solution, say g_p (obtained by Matlab's backslash operator), and substituting it in the equation $y|_\tau = Y_f g_p$, we obtain the unique output trajectory. (This can be seen as an additional requirement to obtain unique solution for an underdetermined system.) Conditions under which g is unique are not known and it would be interesting to uncover the natural assumptions, which guarantees the existence of unique g. For LTI systems, if the length of the initial trajectory w_{ini} is greater than or equal to the lag of the system, output is uniquely determined from the input and initial condition [14, Lemma 1].

V. Numerical Experiments

Example 1. Consider the system

$$y(t) = u(t) + 2u(t-1) + 3u^2(t) + 4u^2(t-1). \tag{8}$$

We generate two sets of 500 samples of outputs by applying a uniform random input time series with zero initial conditions. We assume one set of input/output time series as the observed data $(u_d, y_d) = (u|_{500}, y|_{500})$. We then compute first 50 samples of outputs of the other dataset by using Algorithm 1 with the given input and zero initial condition. The results are shown in Fig. 1. Because there is no noise in the data, we obtain perfect match between computed and true outputs.

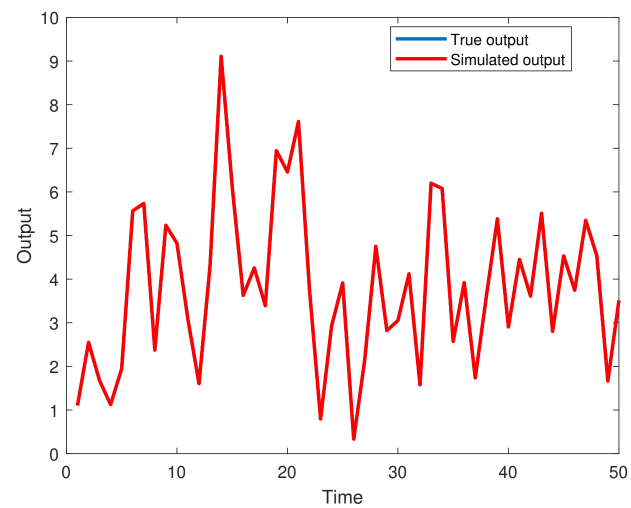

Fig. 1. True outputs and the ones computed by Algorithm 1. Because the data are exact (noise-free), the true and computed outputs coincide.

We now explore the noisy case. We adapt the previous setup with the assumption that the outputs are corrupted by normally distributed additive noise. Clearly, as we increase the noise level, we can expect increase in the relative error, which is defined as $\|\bar{y} - \tilde{y}\|/\|\bar{y}\|$, where $\bar{y}$ and $\tilde{y}$ are the true and computed outputs, respectively. To show the effectiveness of our approach, we add increasing level of noise (by increasing the standard deviation) to the true outputs and then by using our algorithm we compute the outputs and subsequently compute the relative errors. The results are displayed in Table I.

VI. Conclusions

We have generalized the fundamental lemma for LTI systems to NARX systems. It has been seen that the result requires a persistency of excitation condition that involves

TABLE I
RELATIVE ERRORS CORRESPONDING TO VARIOUS NOISE LEVELS

Standard deviation	0.01	0.05	0.1	0.15	0.2
Relative error	0.0059	0.0318	0.0645	0.0813	0.1087

not only the inputs, but also the outputs. This result is then used to develop an algorithm that implements the data-driven simulation. Numerical experiments have been performed to illustrate the effectiveness of the developed algorithm. It would be interesting to see how the relative error in the noisy case is affected by the length of the trajectory or the degree of the term of the polynomial. Further, extending these results to data-driven control problems such as output tracking and stabilization are the future research topics.

REFERENCES

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

[2] M. Jansson, "Subspace identification and arx modeling," *IFAC Proceedings Volumes*, vol. 36, no. 16, pp. 1585 – 1590, 2003.

[3] K. Worden, G. Manson, and E. Cross, "On gaussian process narx models and their higher-order frequancy response functions," in *Solving Computationally Expensive Engineering Problems*, 2014.

[4] K. Worden, W. Becker, T. Rogers, and E. Cross, "On the confidence bounds of gaussian process narx models and their higher-order frequency response functions," *Mech. Syst. Signal Process.*, vol. 104, pp. 188 – 223, 2018.

[5] Q. Liu, W. Chen, H. Hu, Q. Zhu, and Z. Xie, "An optimal narx neural network identification model for a magnetorheological damper with force-distortion behavior," *Frontiers in Materials*, vol. 7, pp. 1–12, 2020.

[6] L. C. B. da Silva et al., "Narx neural network model for strong resolution improvement in a distributed temperature sensor," *Appl. Opt.*, vol. 57, pp. 5859–5864, 2018.

[7] S. Mohanty, P. K. Patra, and S. S. Sahoo, "Prediction of global solar radiation using nonlinear auto regressive network with exogenous inputs (narx)," *2015 39th National Systems Conference (NSC)*, pp. 1–6, 2015.

[8] E. Pisoni, M. Farina, C. Carnevale, and L. Piroddi, "Forecasting peak air pollution levels using narx models," *Eng. Appl. Artif. Intell.*, vol. 22, no. 4, pp. 593–602, 2009.

[9] L. Ruiz, M. Cuéllar, M. Calvo-Flores, and M. Jiménez, "An application of non-linear autoregressive neural networks to predict energy consumption in public buildings," *Energies*, vol. 9, pp. 1–21, 2016.

[10] E. Diaconescu, "The use of narx neural networks to predict chaotic time series," *WSEAS Transactions on Computer Research*, vol. 3, pp. 182–191, 2008.

[11] Hang Xie, Hao Tang, and Yu-He Liao, "Time series prediction based on narx neural networks: An advanced approach," in *2009 International Conference on Machine Learning and Cybernetics*, vol. 3, 2009, pp. 1275–1279.

[12] I. Markovsky, "A missing data approach to data-driven filtering and control," *IEEE Trans. Autom. Control*, vol. 62, pp. 1972–1978, April 2017.

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

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

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

[16] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality and robustness," *IEEE Trans. Autom. Control*, 2019.

[17] P. Dreesen and I. Markovsky, "Data-driven simulation using the nuclear norm heuristic," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2019)*, Brighton, UK, 2019, pp. 8207–8211.

[18] I. Markovsky, J. C. Willems, S. Van Huffel, and B. De Moor, *Exact and Approximate Modeling of Linear Systems: A Behavioral Approach*. SIAM, 2006.

[19] H. J. van Waarde, C. D. Persis, M. Camlibel, and P. Tesi, "Willems' fundamental lemma for state-space systems and its extension to multiple datasets," *IEEE Control Syst. Lett.*, vol. 4, pp. 602–607, 2020.

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

[21] J. G. Rueda-Escobedo and J. Schiffer, "Data-driven internal model control of second-order discrete volterra systems," in *IEEE Conf. Decision Control*. IEEE, 2020, pp. 4572–4579.

[22] I. Markovsky, "An algorithm for closed-loop data-driven simulation," in *15th IFAC Symposium on System Identification*, Saint-Malo, France, July 2009, pp. 114–115.

[23] G. Baggio, V. Katewa, and F. Pasqualetti, "Data-driven minimum-energy controls for linear systems," *IEEE Control Syst. Lett.*, vol. 3, no. 3, pp. 589–594, 2019.

[24] H. Van Waarde, J. Eising, H. Trentelman, and K. Camlibel, "Data informativity: a new perspective on data-driven analysis and control," *IEEE Trans. Autom. Control*, 2020.

[25] V. K. Mishra and I. Markovsky, "The set of linear time-invariant unfalsified models with bounded complexity is affine," *IEEE Trans. Autom. Control*, 2020.

[26] J. Berberich, J. Köhler, M. A. Muller, and F. Allgower, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Trans. Autom. Control*, 2020.

[27] A. Romer, J. Berberich, J. Köhler, and F. Allgöwer, "One-shot verification of dissipativity properties from input–output data," *IEEE Control Syst. Lett.*, vol. 3, no. 3, pp. 709–714, 2019.

[28] Z. Wang and D. Liu, "Data-based controllability and observability analysis of linear discrete-time systems," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2388–2392, 2011.

[29] D. Liu, P. Yan, and Q. Wei, "Data-based analysis of discrete-time linear systems in noisy environment: Controllability and observability," *Inf. Sci.*, vol. 288, pp. 314–329, 2014.

[30] V. K. Mishra, I. Markovsky, and B. Grossmann, "Data-driven tests for controllability," *IEEE Control Syst. Lett.*, vol. 5, no. 2, pp. 517–522, 2020.

[31] T. Jain, J. J. Yamé, and D. Sauter, "Trajectory-based real-time control of an electrical circuit against unknown faults," *J. Franklin Inst.*, vol. 351, no. 2, pp. 986–1000, 2014.

[32] S. Yin, H. Luo, and S. X. Ding, "Real-time implementation of fault-tolerant control systems with performance optimization," *IEEE Trans. Ind. Electron.*, vol. 61, no. 5, pp. 2402–2411, 2013.

APPENDIX

Consider the following NARX system

$$y(t) = \sum_{i=1}^{\mathtt{d}_1}\sum_{j=0}^{\ell_1}\alpha_{ij}u^i(t-j) + \sum_{k=1}^{\mathtt{d}_2}\sum_{l=1}^{\ell_2}\beta_{kp}y^k(t-l) + \sum_{m=1}^{\mathtt{d}_3}\sum_{j=1}^{\ell_1}\sum_{l=1}^{\ell_2}\delta_{mjp}u^m(t-j)y^m(t-l) \tag{9}$$

where α_{ij}, α_{kl}, and δ_{mjl} are parameters. The above system can be rewritten as follows

$$y(t) = \sum_{i=1}^{\mathtt{d}_1}\Gamma_i^\top\eta^i(t) + \sum_{k=1}^{\mathtt{d}_2}\Lambda_k^\top\xi^k(t) + \sum_{m=1}^{\mathtt{d}_3}\Delta_m^\top\zeta^m(t), \tag{10}$$

where in addition to (3), we define

$$\begin{aligned}
\zeta^m(t) &= Z^m(t)\otimes\xi^m(t)\\
Z^m(t) &= \begin{bmatrix}u^m(t-1) & u^m(t-2) & \dots & u^m(t-\ell_1)\end{bmatrix}^\top\\
\Delta_m &= \begin{bmatrix}\delta_{m11} & \dots & \delta_{m1\ell_2} & \dots & \delta_{m\ell_1 1} & \dots & \delta_{m\ell_1\ell_2}\end{bmatrix}^\top.
\end{aligned}$$

Now, Theorem 1 and Algorithm 1 may be analogously developed for system (9).