Bilinear Realization from I/O Data with NNs

D. S. Karachalios[1,2(✉)], I. V. Gosea[2], K. Kour[2], and A. C. Antoulas[2,3,4]

[1] Institute of Electrical Engineering in Medicine, University Luebeck, Luebeck, Germany
[2] Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany
dimitrios.karachalios@uni-luebeck.de,
{gosea,kour}@mpi-magdeburg.mpg.de
[3] Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005, USA
aca@rice.edu
[4] Baylor College of Medicine, Houston, TX 77030, USA

Abstract. We present a method that connects a well-established nonlinear (bilinear) identification method from data in the time domain with the advantages of neural networks (NNs). The main challenge for fitting bilinear systems is the accurate recovery of the corresponding Markov parameters from the input and output measurements. Afterward, a realization algorithm similar to that proposed by Isidori can be employed. The novel step is that NNs are used here as a surrogate data simulator to construct input-output (i/o) data sequences from a single experiment. Then, classical realization theory is used to build an interpretable bilinear model that can further optimize engineering processes through robust simulations and control design.

1 Introduction

Evolutionary phenomena can be formally described as continuous dynamical models with partial differential equations (PDE)s. The continuous nature of these physical models is equipped with analytical results for an efficient discrete approximation in space and in time. In particular, methods such as finite elements or finite differences bridge the continuous analytical laws of the physical world with computational science [1]. On the other hand, data science allows model discovery when the identification feature is considered [2]. Quantification of these equivalences, in combination with the stochastic nature that governs real-world applications, aims to explain the digital twin [3]. Spatial discretization of PDEs, in many cases, results in a continuous in-time system of ordinary differential equations (ODE)s that is described by the operators $(\mathbf{F},\ \mathbf{G})$ and can be approximated with Carleman linearization (e.g., bilinear system form) [4] where we present the single-input single-output (SISO) case with the continuous operators to be denoted with the subscript ”c.”

$$\boldsymbol{\Sigma}:\begin{cases}\dot{\mathbf{x}}(t)=\mathbf{F}(\mathbf{x}(t))+\mathbf{G}(\mathbf{x}(t))u(t)\\ y(t)=\mathbf{H}\mathbf{x}(t),\ \mathbf{x}_0=\mathbf{0},\ t\geq 0.\end{cases}\xrightarrow[\boldsymbol{\Sigma}\approx\boldsymbol{\Sigma}_{bil}]{\text{Carleman}}\boldsymbol{\Sigma}_{bil}:\begin{cases}\dot{\mathbf{x}}(t)=\mathbf{A}_c\mathbf{x}(t)+\mathbf{N}_c\mathbf{x}(t)u(t)+\mathbf{B}_cu(t)\\ y(t)=\mathbf{C}_c\mathbf{x}(t),\ \mathbf{x}_0=\mathbf{0},\ t\geq 0.\end{cases}\quad(1)$$

If the original system has dimension n, since Carleman linearization [4] preserves up to the quadratic term $\mathbf{x}(t)\otimes\mathbf{x}(t)$[1], the dimension of the resulting bilinear system $(\mathbf{A}_c,\ \mathbf{N}_c,\ \mathbf{B}_c,\ \mathbf{C}_c)$ increases to $N=n^2+n$.

[1] $\otimes$: Kronecker product.

M. van Beurden et al. (Eds.): SCEE 2022, MI 43, pp. 184–192, 2024.
https://doi.org/10.1007/978-3-031-54517-7_21

Data-driven methods can be classified into two general classes. The first provides prediction through regression techniques such as neural networks (NN)s from machine learning (ML). At the same time, the second has its roots in system theory and allows model discovery [2,5]. Generally, NNs are sensitive to parameter tuning and lack model interpretability due to the inherent "black-box" structure [6], while the latter construct interpretable models and can explain the hidden dynamics. ML models learn the features by composing non-linear activation functions and utilizing mainly the backpropagation algorithm to adjust the network weights during training. Therefore, by using data points for training, the prediction would be expressed as a function of these data points (finite memory). Until recently, the ML and system identification (SI) techniques were developed independently. But in recent years, great effort has been invested into establishing a common ground [7].

The authors in [8] have extended the subspace realization theory from linear to bilinear systems. For example, in applications that concern chemical processes, the controls are flow rates, and, from the first principles, e.g., mass and heat balances, these will appear in the system equations as products with the state variables. Therefore, the bilinear equation has the physical form $M\dot{\mathbf{x}} = \sum_i \mathbf{q}_i \mathbf{x}_i - \sum_m \mathbf{q}_m \mathbf{x}_m$, $\mathbf{q}$(inputs), $\mathbf{x}$(state). The authors of [9] could construct bilinear systems with white noise input based on an iterative deterministic-stochastic subspace approach. The author in [10] uses the properties of the linear model of the bilinear system when subjected to a constant input. Constant inputs can transform the bilinear model to an equivalent linear model [11].

In Sect. (2), we introduce the theory of bilinear realization by explaining in detail the data acquisition procedure to compute the bilinear Markov parameters that will enter the bilinear Hankel matrix. Further, we present a concise algorithm that can achieve bilinear identification, detailed by two examples. In Sect. (3), we train a neural network with a single i/o data sequence to mimic the unknown simulator and combine it with the bilinear realization theory. As a result, we could construct a bilinear model from a single i/o data with slightly better fit performance compared with another state-of-the-art bilinear SI approach. Finally, we provide the conclusion and the outlook in Sect. (4).

2 The Bilinear Realization Framework

In the case of linear systems, Ho and Kalman [12] have provided the mathematical foundations for realizing linear systems from i/o data. In the nonlinear case and towards the exact scope of identifying nonlinear systems, Isidori in [13] has extended these results for the bilinear case, and Al Baiyat in [14] has provided an SVD-based algorithm.

Time discretization as in [15] of the single-input single-output (SISO) bilinear system Eq. (1) with sampling time Δt, results in fully discrete models defined at time instances given by $0 < \Delta t < 2\Delta t < \cdots < k\Delta t$, with $\mathbf{x}_c(k\Delta t) = \mathbf{x}_k$ and $u(k\Delta t) = u_k$ for $k = 0, \ldots, m-1$

$$\mathbf{\Sigma}_{\text{disc}} : \begin{cases} \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{N}\mathbf{x}_k u_k + \mathbf{B}u_k, \\ y_k = \mathbf{C}\mathbf{x}_k, \ \mathbf{x}_0 = \mathbf{0}. \end{cases} \tag{2}$$

The discrete-time system in Eq. (2) has state dimension N, so, $\mathbf{x} \in \mathbb{R}^N$ and the operators have dimensions $\mathbf{A}, \mathbf{N} \in \mathbb{R}^{N \times N}$, $\mathbf{B}, \mathbf{C}^T \in \mathbb{R}^N$. We have assumed homogeneous initial conditions and a zero feed-forward term (e.g., $\mathbf{D} = 0$ term). As far as the authors are aware, the forward Euler scheme is the only numerical scheme that preserves the bilinear structure in a discrete set-up with the cost of conditional stability. Moreover, a more sophisticated scheme can exactly interpolate the continuous model at the sampling points in [16] but is restricted to only a subclass of bilinear systems. Therefore, a good choice in terms of stability is the backward Euler scheme from [15], which preserves the bilinear structure asymptotically, and the transformation in Eq. (3) that leads to the discrete system is

$$\begin{aligned} &\phi:\ \mathbf{A}=(\mathbf{I}-\Delta t\mathbf{A}_c)^{-1},\ \mathbf{N}=\Delta t(\mathbf{I}-\Delta t\mathbf{A}_c)^{-1}\mathbf{N}_c,\ \mathbf{B}=\Delta t(\mathbf{I}-\Delta t\mathbf{A}_c)^{-1}\mathbf{B}_c,\ \mathbf{C}=\mathbf{C}_c,\\ &\boldsymbol{\Sigma}_b^c:(\mathbf{A}_c,\mathbf{N}_c,\mathbf{B}_c,\mathbf{C}_c)\overset{\phi^{-1}}{\leftrightarrow}\boldsymbol{\Sigma}_b^d:(\mathbf{A},\mathbf{N},\mathbf{B},\mathbf{C}) \end{aligned} \tag{3}$$

Definition 1. The reachability matrix $\mathscr{R}_n=\begin{bmatrix}\mathbf{R}_1 & \cdots & \mathbf{R}_n\end{bmatrix}$ is defined recursively from the following relation: $\mathbf{R}_j=\begin{bmatrix}\mathbf{A}\mathbf{R}_{j-1} & \mathbf{N}\mathbf{R}_{j-1}\end{bmatrix},\ j=2,\ldots n,\ \mathbf{R}_1=\mathbf{B}$.

Then, the state space of the bilinear system is spanned by the states reachable from the origin if and only if $\text{rank}(\mathscr{R}_n)=n$.

Definition 2. The observability matrix $\mathscr{O}_n=\begin{bmatrix}\mathbf{O}_1 & \cdots & \mathbf{O}_n\end{bmatrix}^T$ is defined recursively from the following relation: $\mathbf{O}_j^T=\begin{bmatrix}\mathbf{O}_{j-1}\mathbf{A} & \mathbf{O}_{j-1}\mathbf{N}\end{bmatrix}^T,\ j=2,\ldots n,\ \mathbf{O}_1=\mathbf{C}$.

Then the state space of the bilinear system is observable iff $\text{rank}(\mathscr{O}_n)=n$. The following Def. (3) will allow a concise representation of the i/o relation.

Definition 3. $\mathbf{u}_j(h)=\begin{bmatrix}\mathbf{u}_{j-1}(h)\\ \mathbf{u}_{j-1}(h)u(h+j-1)\end{bmatrix},\ j=2,\ldots,\ \mathbf{u}_1(h)=u(h)$.

Let $\{\mathbf{w}_1,\mathbf{w}_2,\ldots,\mathbf{w}_j,\ldots\}$ be an infinite sequence of row vectors, in which $\mathbf{w}_j\in\mathbb{R}^{1\times 2^{j-1}}$ and is defined recursively as follows $\mathbf{w}_j=\mathbf{C}\mathbf{R}_j,\ j=1,2,\ldots;$

The state response of system Eq. (2) from the state $\mathbf{x}_0=\mathbf{0}$ at time $k=0$, under a given input function, can be expressed as:

$$\begin{aligned} \mathbf{x}_1&=\mathbf{B}u_0\triangleq\mathbf{R}_1\mathbf{u}_1(0),\\ \mathbf{x}_2&=\mathbf{A}\mathbf{R}_1\mathbf{u}_1(0)+\mathbf{N}\mathbf{R}_1\mathbf{u}_1(0)u(1)+\mathbf{B}u(1)\triangleq\mathbf{R}_2\mathbf{u}_2(0)+\mathbf{R}_1\mathbf{u}_1(1),\\ &\vdots\\ \mathbf{x}_k&=\sum_{j=1}^{k}\mathbf{R}_j\mathbf{u}_j(k-j),\ k=1,2,\ldots; \end{aligned} \tag{4}$$

Finally, the zero-state input-output map of system Eq. (2) after multiplication with the vector $\mathbf{C}$ from the left can be written as:

$$y_k=\sum_{j=1}^{k}\mathbf{w}_j\mathbf{u}_j(k-j),\ k=1,2,\ldots; \tag{5}$$

2.1 The Bilinear Markov Parameters

The bilinear Markov (invariant) parameters are encoded in the $\{\mathbf{w}_j\}$ vectors for $j \in \mathbb{Z}_+$. These are invariant quantities of the bilinear system in connection with the input-output relation. After making use of Def. (3), we can write

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix}}_{\mathbf{Y}} = \underbrace{\begin{bmatrix} \mathbf{u}_1^T(0) & 0 & \cdots & 0 \\ \mathbf{u}_1^T(1) & \mathbf{u}_2^T(0) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_1^T(k-1) & \mathbf{u}_2^T(k-2) & \cdots & \mathbf{u}_k^T(0) \end{bmatrix}}_{\mathbf{U}} \cdot \underbrace{\begin{bmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \\ \vdots \\ \mathbf{w}_k^T \end{bmatrix}}_{\mathbf{W}}, \tag{6}$$

where the dimensions are: $\mathbf{Y} \in \mathbb{R}^{k\times 1}$, $\mathbf{U} \in \mathbb{R}^{k\times m}$, and $\mathbf{W} \in \mathbb{R}^{m\times 1}$.

The least squares problem filled out with k time steps will remain under-determined $\forall k \in \{2,3,\ldots\}$ as long as the $m = 2^k - 1$ bilinear Markov parameters are activated. Thus, we must deal with the k equations and the $2^k - 1$ unknowns. Solving an under-determined system is not impossible, but the solutions are infinite, and regularization schemes cannot easily lead to identification. Therefore, one way to uniquely identify bilinear Markov parameters and determine the solution vector $\mathbf{W}$ can be achieved by solving a coupled least squares system after applying several simulations to the original system.

To uniquely determine the $(2^k - 1)$ parameters, the column rank of the matrix $\mathbf{U}$ should be complete. This can be accomplished by adding rows with more experiments to the matrix $\mathbf{U}$ until the new augmented matrix $\hat{\mathbf{U}}$ has more rows than columns. Thus, we need at least 2^{k-1} independent simulations of the original system. That is exactly the bottleneck expected for nonlinear identification frameworks that deal with time-domain data. Later, we will relax this condition in a novel way using NNs. Equation (7) describes the coupled linear least squares system with $d = 2^{k-1}$ independent simulations that can provide the unique solution $\mathscr{W}$ with bilinear Markov parameters.

$$\underbrace{\begin{bmatrix} \mathbf{Y}_1 \cdots \mathbf{Y}_d \end{bmatrix}^T}_{\hat{\mathbf{Y}}} = \underbrace{\begin{bmatrix} \mathbf{U}_1 \cdots \mathbf{U}_d \end{bmatrix}^T}_{\hat{\mathbf{U}}} \cdot \mathscr{W} \tag{7}$$

Hence, we repeat the simulation d times, and each time we get k equations, with the i^{th} simulation to be $\mathbf{Y}_i = \begin{bmatrix} y_1^{(i)} \; y_2^{(i)} \cdots y_k^{(i)} \end{bmatrix}^T$ and accordingly for the $\mathbf{U}_i$, the real matrix $\hat{\mathbf{U}}$ has dimension $2^k \times (2^k - 1)$. After concatenating all the lower triangular matrices with full column rank, the matrix $\hat{\mathbf{U}}$ results. To enforce that $\hat{\mathbf{U}}$ will also have full column rank, one choice is to use a white input (sampled from a Gaussian distribution) for the simulations. The use of a white input is widespread for SI. Still, in that case, a careful choice of deterministic inputs can make the inversion exact and recover the bilinear Markov parameters. The solution is as follows: $\text{rank}(\hat{\mathbf{U}}) = 2^k - 1$, so, the unique solution is: $\mathscr{W} = \hat{\mathbf{U}}^{-1}\hat{\mathbf{Y}} \in \mathbb{R}^{2^k-1}$. The vector $\mathscr{W}$ contains the $2^k - 1$ bilinear Markov parameters. A generalized Hankel matrix can be computed from the bilinear Markov parameters.

2.2 The Bilinear Hankel Matrix

The bilinear Hankel matrix is the product of the observability and reachability matrices. The bilinear Hankel matrix is denoted with $\mathscr{H}_b$ and is defined as the product of the following two infinite matrices $\mathscr{O}$, $\mathscr{R}$,

$$\mathscr{H}_b = \mathscr{O}\mathscr{R} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CN} \\ \vdots \end{bmatrix} \begin{bmatrix} \mathbf{B}\ \mathbf{AB}\ \mathbf{NB} \cdots \end{bmatrix} = \begin{bmatrix} \mathbf{CB} & \mathbf{CAB} & \mathbf{CNB} & \cdots \\ \mathbf{CAB} & \mathbf{CA}^2\mathbf{B} & \mathbf{CANB} & \cdots \\ \mathbf{CNB} & \mathbf{CNAB} & \mathbf{CN}^2\mathbf{B} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{8}$$

Equation (8) reveals the connection with the bilinear Markov parameters $\mathscr{W} = \mathbf{C}\mathscr{R}$ that appear in the first row of $\mathscr{H}_b$. In general, the construction of the bilinear Hankel matrix is described in [13] with the partial and completed realization theorems along with the partitions[2] $\mathscr{S}^A$, $\mathscr{S}^N$ [14].

2.3 Bilinear Realization Algorithm

Input: Input-output time-domain data from a system $u \rightarrow \boxed{\mathbf{\Sigma}?} \rightarrow y$.
Output: A minimal bilinear system $(\mathbf{A}_r, \mathbf{N}_r, \mathbf{B}_r, \mathbf{C}_r)$ of low dimension r that $\mathbf{\Sigma}_r \approx \mathbf{\Sigma}$.

1. Excite the system $\mathbf{\Sigma}$ k times with $\mathbf{u}_m \sim \mathbf{N}(\mu, \sigma)$ and collect $\mathbf{y}_m$, where $k = 2^{m-1}$.

$$\begin{array}{l} \text{1st simulation } [u_1(1)\cdots u_1(m)] \rightarrow \boxed{\mathbf{\Sigma}} \rightarrow [y_1(1)\cdots y_1(m)] = \mathbf{Y}_1, \text{ and } \mathbf{U}_1 \text{ as in Definition 3} \\ \qquad\vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ \text{kth simulation } [u_k(1)\cdots u_k(m)] \rightarrow \boxed{\mathbf{\Sigma}} \rightarrow [y_k(1)\cdots y_k(m)] = \mathbf{Y}_k, \text{ and } \mathbf{U}_k \text{ as in Definition 3.} \end{array}$$

2. Identify the $(2^m - 1)$ bilinear Markov parameters by solving the system in (7).
3. Construct the bilinear Hankel matrix $\mathscr{H}_b$ and the sub-matrices $\mathscr{S}^{\mathbf{A}}$, $\mathscr{S}^{\mathbf{N}}$.
4. Compute $[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \text{SVD}(\mathscr{H}_b)$ and truncate w.r.t the singular values decay ($r \ll n$) - the reduced/identified bilinear model $(\mathbf{A}_r, \mathbf{N}_r, \mathbf{B}_r, \mathbf{C}_r)$ is constructed

$$\mathbf{A}_r = \mathbf{\Sigma}^{-1/2}\mathbf{U}^T\mathscr{S}^{\mathbf{A}}\mathbf{V}\mathbf{\Sigma}^{-1/2} \tag{9}$$

$$\mathbf{N}_r = \mathbf{\Sigma}^{-1/2}\mathbf{U}^T\mathscr{S}^{\mathbf{N}}\mathbf{V}\mathbf{\Sigma}^{-1/2} \tag{10}$$

$$\mathbf{B}_r = \mathbf{\Sigma}^{1/2}\mathbf{V}^T \quad \rightarrow \quad \text{1st column} \tag{11}$$

$$\mathbf{C}_r = \mathbf{U}\mathbf{\Sigma}^{1/2} \quad \rightarrow \quad \text{1st row} \tag{12}$$

Example 1. (A toy system) Let the following bilinear system of order 2 be

$$\mathbf{A} = \begin{bmatrix} 0.9 & 0.0 \\ 0.0 & 0.8 \end{bmatrix}, \mathbf{N} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 1.0 \\ 0.0 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}^T. \tag{13}$$

Applying the algorithm in Sect. (2.3), by choosing $m = 4$, we can recover $2^m - 1 = 15$ bilinear Markov parameters. The solution of the system in Eq. (7) is:

$$\mathbf{W} = \begin{bmatrix} 1.0\ 0.9\ 0.4\ 0.81\ 0.33\ 0.36\ 0.22\ 0.729\ 0.273\ 0.297\ 0.183\ 0.324\ 0.18\ 0.198\ 0.118 \end{bmatrix}$$

[2] $\mathscr{S}^A = \{\text{set of } \mathscr{H}_b\text{columns : from } 2^m \text{ to } (3 \cdot 2^{m-1} - 1),\ m = 1, 2, \ldots\}$,
$\mathscr{S}^N = \{\text{set of } \mathscr{H}_b\text{columns : from } 3 \cdot 2^{m-1} \text{ to } (2^{m+1} - 1),\ m = 1, 2, \ldots\}$.

By reshuffling the vector $\mathscr{W}$, we can form the $\mathscr{H}_b$ matrix and the shifted versions $\mathscr{S}^{\mathbf{A}}$, $\mathscr{S}^{\mathbf{N}}$ as described above. The Hankel matrix (3 rows & 7 columns are displayed) along with the shifted versions are $\left\{\mathscr{H}_b,\ \mathscr{S}^{A},\ \mathscr{S}^{N}\right\} :=$

$$\left\{ \begin{bmatrix} 1.0 & 0.9 & 0.4 & 0.81 & 0.33 & 0.36 & 0.22 \\ 0.9 & 0.81 & 0.33 & 0.729 & 0.273 & 0.297 & 0.183 \\ 0.4 & 0.36 & 0.22 & 0.324 & 0.18 & 0.198 & 0.118 \end{bmatrix}, \begin{bmatrix} 0.9 & 0.81 & 0.33 \\ 0.81 & 0.729 & 0.273 \\ 0.36 & 0.324 & 0.18 \end{bmatrix}, \begin{bmatrix} 0.4 & 0.36 & 0.22 \\ 0.33 & 0.297 & 0.183 \\ 0.22 & 0.198 & 0.118 \end{bmatrix} \right\}.$$

In Fig. (1), the 3rd normalized singular value has reached machine precision $\sigma_3/\sigma_1 = 5.2501e-17$, that is the criterion for choosing the order of the fitted system (which is minimal, in this case) of the underlying bilinear system. Therefore, we construct a bilinear model of order $r=2$, and the realization obtained is equivalent to the original (minimal) one, up to a coordinate (similarity) transformation. Other ways of constructing reduced models from Hankel⊂Loewner matrices can be obtained with the CUR (cross approximations based) decomposition scheme as in [17] (Fig. 1).

$$\mathbf{A}_r = \begin{bmatrix} 0.89394 & 0.11305 \\ 0.0050328 & 0.80606 \end{bmatrix},\ \mathbf{N}_r = \begin{bmatrix} 0.41116 & -0.2281 \\ -0.24782 & 0.088841 \end{bmatrix},\ \mathbf{B}_r = \begin{bmatrix} -1.0001 \\ -0.053577 \end{bmatrix},\ \mathbf{C}_r = \begin{bmatrix} -1.0001 \\ 0.0040101 \end{bmatrix}^T. \tag{14}$$

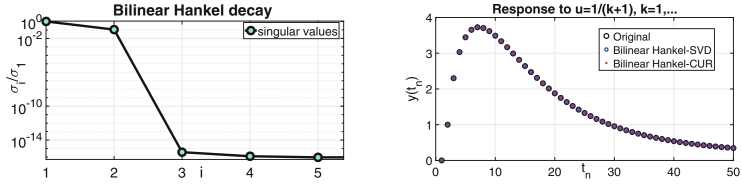

Fig. 1. On the left figure, the singular value decay of the bilinear Hankel matrix is depicted. On the right figure, the input response $u_k = 1/(k+1),\ k = 0,1,\ldots$ certifies that all models are equivalent.

Example 2. (The viscous Burgers' equation example) Following [15] after spatial semi-discretization and the Carleman linearization technique, yields a bilinear system of dimension $N = 30^2 + 30 = 930$. The viscosity parameter is $\nu = 0.1$; the sampling time is $\Delta t = 0.1$ and with $2^{m-1} = 512$ independent random inputs of length $m = 10$ each, we construct a database of $5,120$ points. Solving Eq. (7), we get the bilinear Markov parameters, and the bilinear Hankel matrix is constructed. On the left pane of Fig. (2), the decay of bilinear Hankel singular values captures the nonlinear nature of the Burgers' equation, while, on the other hand, the linear Hankel framework captures only the linear minimal response. It is evident in the right pane of Fig. (2) that after using the inverse transformation ϕ from (3), the reduced continuous-time bilinear model of order $r = 18$ performs well, producing an error $O(10^{-5})$ where at the same time the linear fit is off (Fig. 2).

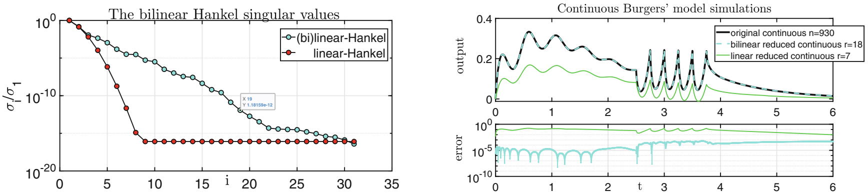

Fig. 2. Left pane: The recommended reduced bilinear model with $\Delta t = 0.1$ is of order $r = 18$ where $\sigma_{19}/\sigma_1 = 1.18 \cdot 10^{-12}$. **Right pane**: $u_1 = (1 + \cos(2\pi t))e^{-t}, t \in [0, 2.5], u_2 = 2\text{sawtooth}(8\pi t), t \in [2.5, 3.75], u_3 = 0$, it is compared with a continuous bilinear identification method based on the Loewner framework in both frequency and time domain approaches [5, 18].

3 From a Single Data Sequence to Bilinear Realization

A repetitive data assimilation simulation in the time domain is required to achieve bilinear realization as in [13]. In many cases, the data from a simulated system are available as a single i/o sequence [9]. Using the NARX-net-based model, in the case of a single experiment, the expensive, repetitive simulations can be avoided in a real engineering environment. These models learn from a unique data sequence and can predict the output behavior under different excitations. That is precisely where the NARX-net model architecture will play the role of a surrogate simulator. Then, by constructing an NN-based model [19] and combining the realization theory in [13], a state-space bilinear model can be constructed as in (2). Using a state-space model, which relies on the classical nonlinear realization theory with many known results (especially on bilinear systems and in the study direction of stability, approximation, and control), is beneficial compared to the NARX.

Example 3. (Heat exchanger) The process is a liquid-saturated steam heat exchanger, where water is heated by pressurized saturated steam through a copper tube. The input variable is the liquid flow rate, and the output variable is the outlet liquid temperature. The sampling time is 1(s), and the number of samples is 4,000. More details can be found in [20], and the data set can be downloaded from the database to identify systems (DaISy): https://homes.esat.kuleuven.be/~tokka/daisydata.html.

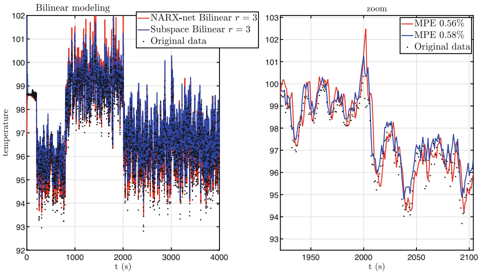

Fig. 3. Comparison and model fit of the proposed NARX-net bilinear model (15) with the subspace method from [9] for the same reduced order $(r = 3)$.

$$\begin{cases} \dot{\mathbf{x}}(t) = \begin{bmatrix} 0.9164 & 0.09167 & -0.1847 \\ -0.2663 & -0.1515 & 0.1232 \\ -0.07227 & 0.4778 & 0.3571 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0.02717 & 0.5169 & 0.5555 \\ -0.09674 & 0.5467 & 0.5696 \\ 0.1878 & -0.06846 & -1.981 \end{bmatrix} \mathbf{x}(t)u(t) + \\ \quad + \begin{bmatrix} 2.9063 \\ 2.909 \\ -0.16088 \end{bmatrix} u(t) + \begin{bmatrix} -1.073 \\ -1.074 \\ 0.05938 \end{bmatrix}, \\ y(t) = \begin{bmatrix} -0.7852 & 0.7794 & -0.05203 \end{bmatrix} \mathbf{x}(t) + 96.9358, \; \mathbf{x}(0) = \mathbf{0}, \; t \geq 0. \end{cases} \tag{15}$$

Figure 3 illustrates the superiority of the proposed method in terms of accuracy. From the single i/o data sequence, a neural network NN with 3-layers and 20-lags was trained using the same training data[3] as in [9] (1000 points). The trained NN was used in the bilinear realization algorithm to generate more data, and a stable reduced bilinear model of order $r = 3$ shown in Eq. (15) was successfully constructed. The original noisy data were explained with a lower mean percentage error MPE $= 0.56\%$ compared to the subspace method for the entire data set. Another NN architecture, s.a., the NARMAX[4] belongs to a subclass of bilinear systems and will filter some nonlinear features without achieving such a good MPE.

4 Conclusion

In conclusion, NN architectures are a superclass of NARMAX models used in the classical robust identification theory. Consequently, NN models share the same strong argument with the Carleman linearization scheme that can approximate general nonlinear systems. Finally, NN and realization theory successfully bridge data science with computational science to build reliable, interpretable nonlinear models. Different NN architectures (s.a., recurrent NNs, DeepOnets, etc.) in combination with other realization frameworks (s.a., the Loewner framework) and for other types of nonlinearities (s.a., quadratic-bilinear) are left for future research endeavors.

References

1. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. SIAM, Philadelphia (2005)
2. Antoulas, A.C., Beattie, C.A., Gugercin, S.: Interpolatory Methods for Model Reduction. SIAM, Philadelphia (2020)
3. Kapteyn, M., Knezevic, D., Huynh, D., Tran, M., Willcox, K.: Data-driven physics-based digital twins via a library of component-based reduced-order models. Int. J. Numer. Meth. Eng. **123**(13), 2986–3003 (2022). https://doi.org/10.1002/nme.6423

[3] **Data detrend**: $u_n = (u - \bar{u})/\sigma_u, \; y_n = (y - \bar{y})/\sigma_y$; **zero-response**: data were doubled in size for learning the zero-response, i.e., $u_n = 0 \rightarrow \boxed{\boldsymbol{\Sigma}} \rightarrow y_n = 0$.

[4] **NARMAX**: The nonlinear auto-regressive moving average model with exogenous input [14, 19].

4. Carleman, T.: Application de la théorie des équations intégrales linéaires aux systèmes d'équations différentielles non linéaires. Acta Math. **59**, 63–87 (1932)
5. Antoulas, A.C., Gosea, I.V., Ionita, A.C.: Model reduction of bilinear systems in the Loewner framework. SIAM J. Sci. Comput. **38**(5), B889–B916 (2016)
6. Schilders, W., Meijer, P., Ciggaar, E.: Behavioural modelling using the MOESP algorithm, dynamic neural networks and the Bartels-Stewart algorithm. Appl. Numer. Math. **58**(12), 1972–1993 (2008). https://doi.org/10.1016/j.apnum.2007.11.013
7. Ljung, L., Hjalmarsson, H., Ohlsson, H.: Four encounters with system identification. Eur. J. Control. **17**(5), 449–471 (2011). https://doi.org/10.3166/ejc.17.449-471
8. Favoreel, W., De Moor, B., Van Overschee, P.: Subspace identification of bilinear systems subject to white inputs. IEEE Trans. Autom. Control **44**(6), 1157–1165 (1999). https://doi.org/10.1109/9.769370
9. dos Santos, P.L., Ramos, J.A., de Carvalho, J.L.M.: Identification of bilinear systems with white noise inputs: an iterative deterministic-stochastic subspace approach. IEEE Trans. Control Syst. Technol. **17**(5), 1145–1153 (2009). https://doi.org/10.1109/TCST.2008.2002041
10. Juang, J.N.: Continuous-time bilinear system identification. Nonlinear Dyn. **39**(1), 79–94 (2005). https://doi.org/10.1007/s11071-005-1915-z
11. Gosea, I.V., Karachalios, D.S., Antoulas, A.C.: On computing reduced-order bilinear models from time-domain data. PAMM **21**(1), e202100,254 (2021). https://doi.org/10.1002/pamm.202100254
12. Ho, B.L., Kalman, R.E.: Editorial: effective construction of linear state-variable models from input/output functions. at - Automatisierungstechnik 14(1–12), 545–548 (1966). https://doi.org/10.1524/auto.1966.14.112.545
13. Isidori, A.: Direct construction of minimal bilinear realizations from nonlinear input-output maps. IEEE Trans. Autom. Control **18**(6), 626–631 (1973). https://doi.org/10.1109/TAC.1973.1100424
14. Al-Baiyat, S.A.: Model reduction of bilinear systems described by input-output difference equation. Int. J. Syst. Sci. **35**(9), 503–510 (2004). https://doi.org/10.1080/00207720410001734237
15. Benner, P., Breiten, T., Damm, T.: Generalised tangential interpolation for model reduction of discrete-time mimo bilinear systems. Int. J. Control **84**(8), 1398–1407 (2011). https://doi.org/10.1080/00207179.2011.601761
16. Dunoyer, A., Balmer, L., Burnham, K.J., James, D.J.G.: On the discretization of single-input single-output bilinear systems. Int. J. Control **68**(2), 361–372 (1997). https://doi.org/10.1080/002071797223668
17. Karachalios, D.S., Gosea, I.V., Antoulas, A.C.: The Loewner framework for system identification and reduction, pp. 181–228. De Gruyter (2021). https://doi.org/10.1515/9783110498967-006
18. Karachalios, D.S., Gosea, I.V., Antoulas, A.C.: On bilinear time-domain identification and reduction in the loewner framework. In: Benner, P., Breiten, T., Fabender, H., Hinze, M., Stykel, T., Zimmermann, R. (eds.) Model Reduction of Complex Dynamical Systems, vol. 171, pp. 3–30. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72983-7_1
19. Billings, S.A.: Neural Networks for Nonlinear System Identification, chap. 8, pp. 261–287. John Wiley & Sons, Ltd., Hoboken (2013). https://doi.org/10.1002/9781118535561.ch8
20. Bittanti, S., Piroddi, L.: Nonlinear identification and control of a heat exchanger: a neural network approach. J. Franklin Inst. **334**(1), 135–153 (1997). https://doi.org/10.1016/S0016-0032(96)00059-2