European Journal of Control 85 (2025) 101360

Contents lists available at ScienceDirect

European Journal of Control

journal homepage: www.sciencedirect.com/journal/european-journal-of-control

Are Artificial Neural Networks suitable for data-driven moment matching?☆

Matteo Scandella [a,*], Davide Previtali [a], Alessio Moreschini [b]

[a] Department of Management, Information and Production Engineering, University of Bergamo, via Marconi 5, Dalmine (BG), 24044, Italy
[b] Department of Electrical and Electronic Engineering, Imperial College London, London, SW72AZ, United Kingdom

ARTICLE INFO

Recommended by T. Parisini

Keywords:
Moment matching
Neural networks
Nonlinear systems
Estimation

ABSTRACT

We investigate the use of artificial neural networks in the context of data-driven moment matching for nonlinear systems, comparing it with state-of-the-art approaches that rely on regularized kernel methods or least squares. We propose a novel neural network model that shares the properties of the moment function of a nonlinear system, which can be learned by means of surrogate-based black-box optimization methods (such as Bayesian optimization). To validate the proposed approach, we conduct an extensive simulation analysis of the method on two benchmark model reduction problems, employing different settings and comparing with state-of-the-art methods. This investigation suggests that neural networks are a suitable and promising approach for data-driven moment matching, and they appear to show comparable performance to state-of-the-art methods based on regularized kernel methods.

1. Introduction

Model reduction methods are commonly used to address the challenges associated with large-scale dynamical models that involve a significant number of ordinary or Partial Differential Equations (PDEs) (Rowley & Dawson, 2017; Schilders, 2011; Snowden, van der Graaf, & Tindall, 2017). These methods aim to construct a simplified model that approximates the underlying large-scale system while preserving essential properties. A variety of paradigms and model reduction techniques have been developed, differing in the specific properties they preserve, the criteria used for approximation, and the assumptions made about the underlying system. For Linear Time-Invariant (LTI) systems, classical approaches include methods based on balanced truncation (Glover, 1984; Kawano & Scherpen, 2017; Safonov, Chiang, & Limebeer, 1990) and the interpolation techniques such as the Loewner framework or moment matching (Astolfi, 2010; Benner, Gugercin, & Willcox, 2015; Gosea & Antoulas, 2018; Ionescu & Astolfi, 2015; Mayo & Antoulas, 2007; Moreschini & Astolfi, 2025; Moreschini, Simard, & Astolfi, 2023, 2024; Padoan, 2023; Samari, Sandberg, Johansson, & Lavaei, 2023). See Antoulas (2005), Astolfi et al. (2024), Benner et al. (2015), Scarciotti and Astolfi (2024) for exhaustive overviews of the literature.

Moment matching is a model reduction method that devises a lower-complexity model matching the *moment* of the underlying model. In the case of LTI models, the moment is defined as the value of the transfer function and its derivatives evaluated at specific points in the complex plane where the function is defined (Antoulas, 2005, Sec11.1). Astolfi (2010) shows that the moment of an LTI system is in a one-to-one relation with the steady-state output of the system when it is driven by a signal generated by a specific autonomous LTI system, referred to as a *signal generator*. This new interpretation led to the extension of the moment matching paradigm to nonlinear systems, where the concept of moment relies on the invariant manifold, which exists under certain assumptions, of the system interconnected with the signal generator (Astolfi, 2010). For this reason, the reduction of a nonlinear model by moment matching requires the solution of the *invariant equation* (Carr, 2012; Isidori, 1995) that is a PDE whose complexity can escalate with both the dimension of the signal generator and the system. Additionally, in most practical applications, mathematical models are uncertain, imprecise, or even unavailable, creating additional challenges in finding a reliable reduced model using moment matching. Recently, both these problems have been tackled by employing measurements taken on the plant to directly estimate the model without resorting to the mathematical model of the system, leading to the so-called data-driven moment matching paradigm (Bhattacharjee, Moreschini, & Astolfi, 2025; Haasdonk, Hamzi, Santin, & Wittwar, 2021; Mao & Scarciotti, 2024, 2024; Moreschini, Scandella, & Parisini, 2024; Scarciotti & Astolfi, 2017).

Data-driven moment matching can be interpreted as a learning problem, where the goal is to estimate the moment function of the system to be reduced, modeled as a mapping from the states of a signal generator to the system's steady-state outputs obtained by an experiment in

☆ This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.
* Corresponding author.
E-mail addresses: matteo.scandella@unibg.it (M. Scandella), davide.previtali@unibg.it (D. Previtali), a.moreschini@imperial.ac.uk (A. Moreschini).

https://doi.org/10.1016/j.ejcon.2025.101360
Received 4 August 2025; Received in revised form 19 August 2025; Accepted 21 August 2025
Available online 25 August 2025

steady-state. This perspective has motivated a growing body of research over the past decade. The seminal paper (Scarciotti & Astolfi, 2017) assumes that the moment can be approximated with a linear combination of known functions, estimated using the least squares algorithm. Since then, this technique has been extended to two-sided moment matching (Mao & Scarciotti, 2024) and moment matching methods that are agnostic to the signal generator (Bhattacharjee et al., 2025). More recently, Moreschini, Scandella et al. (2024) proposes a new learning method that employs potentially high-dimensional reproducing kernel Hilbert spaces with a regularization term to handle the bias–variance trade-off.

1.1. Contributions

In this work, we explore the use of *Artificial Neural Networks (ANNs)* for the data-driven moment matching problem. In the past years, ANNs were employed as functional approximators for different tasks, from traditional supervised learning problems (*i.e.*, regression and classification) to image analysis and sequence modeling (Goodfellow, Bengio, & Courville, 2016, Ch 6–10). Furthermore, neural networks are gaining popularity in the system identification community for learning nonlinear dynamical systems (Pillonetto et al., 2025). Apart from the numerous success stories in fields such as speech recognition, computer vision, drug discovery and genomics (LeCun, Bengio, & Hinton, 2015; Liu et al., 2017), the widespread use of ANNs is also motivated by strong theoretical properties including the universal approximation theorem, which states that a sufficiently complex neural network can approximate any continuous function to arbitrary accuracy (Pillonetto et al., 2025). Despite the ever-growing usage of ANNs, their application to data-driven moment matching remains unexplored. Yet, their employment in this context could be particularly beneficial with respect to the literature reviewed in the previous paragraph, potentially overcoming the inability to model complex functions using linear-in-the-parameters models used by Mao and Scarciotti (2024), Scarciotti and Astolfi (2017), and mitigating the scalability issues related to kernel methods used by Moreschini, Scandella et al. (2024) (see Quiñonero-Candela & Rasmussen, 2005; Rudi, Carratino, & Rosasco, 2017; Scandella, Mazzoleni, Formentin, & Previdi, 2021 for more details and possible solutions to the problem). For these reasons, this work proposes a novel ANN-based data-driven moment matching method for nonlinear systems. In particular, we propose a novel ANN model that embodies the properties of the moment function of a nonlinear system. This model can be tuned using data collected from experiments on the plant or through simulation, employing *surrogate-based Black-Box Optimization (BBO)* techniques (Regis, 2020; Vu, D'Ambrosio, Hamadi, & Liberti, 2017), such as Bayesian optimization (Frazier, 2018). Given that the estimated moment is a valid moment function of a nonlinear system, it can be used to construct a reduced model from the collected data. To validate the ANN model, we test it on two model reduction benchmarks against the state-of-the-art data-driven moment matching methods based on linear-in-the-parameters models (Scarciotti & Astolfi, 2017) and regularized kernel methods (Moreschini, Scandella et al., 2024). We show that ANNs are a valid tool for these settings, achieving accuracies comparable with the aforementioned methods even in the presence of noise. Despite representing a substantial advancement over the linear-in-the-parameters models of Scarciotti and Astolfi (2017), ANNs have yet to demonstrate a significant performance improvement over the kernel-based approach of Moreschini, Scandella et al. (2024).

1.2. Organization

The remainder of the paper is organized as follows. Section 2 recalls the notion of moment matching for nonlinear systems. Section 3 describes the data-driven moment matching problem that we tackle in this article. Section 4 introduces a neural network specialized for the approximation of moments. Section 5 explains how to tune the neural network from data. Section 6 provides extensive simulation results that show the benefit of the proposed method with respect to the current literature. Finally, Section 7 concludes the paper with some concluding remarks.

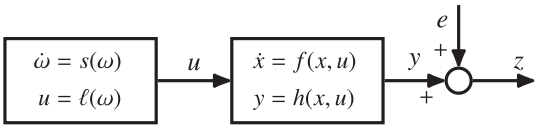


Fig. 1. Block diagram of the moment matching setup in which the measurement of the output y is affected by the additive noise e.

2. Preliminaries

2.1. Notation

The sets of real and natural numbers are denoted by $\mathbb{R}$ and $\mathbb{N}$, respectively. The set of nonnegative real numbers is denoted by $\mathbb{R}_{\geq 0}$. Given $n, m \in \mathbb{N}$, the sets of n-dimensional column vectors and $n \times m$ real matrices are denoted by $\mathbb{R}^n$ and $\mathbb{R}^{n\times m}$, respectively. Furthermore, $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ and $\|\cdot\|_{\mathrm{F}} : \mathbb{R}^{n\times m} \to \mathbb{R}_{\geq 0}$ denote the Euclidean norm and Frobenius norm, respectively. $A^\top$ is the transpose of the matrix A. The cardinality of a set S reads as $|S|$. We denote the ceiling function with $\lceil a \rceil$ for some $a \in \mathbb{R}$. Given two functions f and g, we denote with $f \circ g$ their composition (provided it is well-defined). All mappings are assumed to be sufficiently smooth, if not otherwise stated.

2.2. Moment matching for nonlinear systems

In this section, we recall the notion of *moment* of continuous-time nonlinear systems as first introduced in Astolfi (2010). Consider a continuous-time nonlinear system with input $u : \mathbb{R}_{\geq 0} \to \mathbb{R}^{d_{\mathrm{u}}}$ ($d_{\mathrm{u}} \in \mathbb{N}$), output $y : \mathbb{R}_{\geq 0} \to \mathbb{R}^{d_{\mathrm{y}}}$ ($d_{\mathrm{y}} \in \mathbb{N}$), and state $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^{d_{\mathrm{x}}}$ ($d_{\mathrm{x}} \in \mathbb{N}$) that satisfies the following equations

$$\dot{x}(t) = f(x(t), u(t)), \qquad x(0) = x_0, \tag{1a}$$

$$y(t) = h(x(t), u(t)), \tag{1b}$$

where $f : \mathbb{R}^{d_{\mathrm{x}}} \times \mathbb{R}^{d_{\mathrm{u}}} \to \mathbb{R}^{d_{\mathrm{x}}}$ and $h : \mathbb{R}^{d_{\mathrm{x}}} \times \mathbb{R}^{d_{\mathrm{u}}} \to \mathbb{R}^{d_{\mathrm{y}}}$ are smooth functions such that $f(0,0) = 0$ and $h(0,0) = 0$. The input signal u is generated by an autonomous system, called *signal generator*, described by the equations of the form

$$\dot{\omega}(t) = s(\omega(t)), \qquad \omega(0) = \omega_0, \tag{2a}$$

$$u(t) = \ell(\omega(t)), \tag{2b}$$

where $\omega : \mathbb{R}_{\geq 0} \to \Omega \subseteq \mathbb{R}^{d_\omega}$ ($d_\omega \in \mathbb{N}$) is the state of the signal generator, Ω is a sufficiently small, open, connected, invariant neighborhood of the origin, and the functions $s : \Omega \to \mathbb{R}^{d_\omega}$ and $\ell : \Omega \to \mathbb{R}^{d_{\mathrm{u}}}$ are such that $s(0) = 0$, $\ell(0) = 0$ and smooth. The interconnected model is summarized graphically in Fig. 1 and by the following equations

$$\dot{\omega}(t) = s(\omega(t)), \qquad \omega(0) = \omega_0, \tag{3a}$$

$$\dot{x}(t) = f(x(t), \ell(\omega(t))), \qquad x(0) = x_0, \tag{3b}$$

$$y(t) = h(x(t), \ell(\omega(t))). \tag{3c}$$

To define the notion of moment, we need the following two assumptions.

Assumption 1. The system (1) is minimal, *i.e.*, locally observable (Hermann & Krener, 1977) and locally accessible (Sussmann & Jurdjevic, 1972) around the origin. The origin of (1a), for $u = 0$, is locally exponentially stable.

Assumption 2. The signal generator in (2) is locally observable and neutrally stable around the origin.[1]

Assumption 2 guarantees that the input u is a periodic signal that does not vanish to zero as time goes to infinity for every initial condition $\omega_0 \in \Omega$.

Definition 1 (*Center Manifold*). A smooth function $\pi : \Omega \to \mathbb{R}^{d_x}$ such that $\pi(0) = 0$ is a center manifold of (3) if there exists an open neighborhood $\mathcal{B}_\pi$ of $(0,0)$ such that,

$$\forall (\omega, x) \in \mathcal{B}_\pi, \quad x = \pi(\omega),$$

and

$$(\omega_0, x_0) \in \mathcal{B}_\pi \implies \forall t \in \mathbb{R}_{\geq 0},\ (\omega(t), x(t)) \in \mathcal{B}_\pi.$$

Using Assumptions 1 and 2 and the Center Manifold Theorem (Carr, 2012, Thm 1), we can show that the interconnected system in (3) allows a center manifold π that is the locally unique solution of the PDE (Isidori, 1995, Prop 8.1.1)

$$\frac{\partial \pi}{\partial \omega}(\omega) s(\omega) = f\left(\pi(\omega), \ell(\omega)\right). \tag{4}$$

Moreover, $\mathcal{B}_\pi$ is locally attractive, guaranteeing the existence of a steady-state output $y_{ss} : \mathbb{R}_{\geq 0} \to \mathbb{R}^{d_y}$ such that $y_{ss}(t) = h(\pi(\omega(t)), \ell(\omega(t)))$ and

$$\lim_{t \to \infty} \tau(t) = 0, \tag{5}$$

where $\tau(t) := y(t) - y_{ss}(t)$, for every $t \in \mathbb{R}_{\geq 0}$.

Definition 2 (*Moment*). The (local) moment of (1) at (s, ℓ) is defined as the function $W : \mathbb{R}^{d_\omega} \to \mathbb{R}^{d_y}$ such that

$$\forall \omega \in \Omega, \qquad W(\omega) := h(\pi(\omega), \ell(\omega)). \tag{6}$$

Consider now a continuous-time model with input u, output $\tilde{y} : \mathbb{R}_{\geq 0} \to \mathbb{R}^{d_y}$, and state $\xi : \mathbb{R}_{\geq 0} \to \mathbb{R}^{d_\xi}$ ($d_\xi \in \mathbb{N}$) described by the following equations

$$\dot{\xi}(t) = \tilde{f}(\xi(t), u(t)), \qquad \xi(0) = \xi_0, \tag{7a}$$

$$\tilde{y}(t) = \tilde{h}(\xi(t), u(t)), \tag{7b}$$

where $\tilde{f} : \mathbb{R}^{d_\xi} \times \mathbb{R}^{d_u} \to \mathbb{R}^{d_\xi}$ and $\tilde{h} : \mathbb{R}^{d_\xi} \times \mathbb{R}^{d_u} \to \mathbb{R}^{d_y}$ are smooth functions such that $\tilde{f}(0,0) = 0$ and $\tilde{h}(0,0) = 0$. We say that (7) achieves moment matching at (s, ℓ) if its moment $\tilde{W}$ satisfies the matching condition

$$\tilde{W} = W.$$

Then, we say that (7) is a model of (1) and, if $d_\xi < d_x$, we say that (7) is a *model order reduction* of (1).

In Astolfi (2010), the authors describe a model achieving moment matching with $d_\xi = d_\omega$ in which

$$\tilde{f}(\xi, u) := s(\xi) - \delta(\xi)\ell(\xi) + \delta(\xi)u, \qquad \tilde{h}(\xi, u) := W(\xi), \tag{8}$$

where $\delta : \mathbb{R}^{d_\xi} \to \mathbb{R}^{d_\xi}$ is a function such that (7) satisfies Assumption 1. More recently, this result was generalized in Simard, Moreschini, and Astolfi (2025), where the authors define the family of all the models that achieve moment matching with a generic order $d_\xi \geq d_\omega$.

3. Problem statement

As illustrated in Section 2.2, to define a model that achieves moment matching, we can use the system (7) with $\tilde{f}$ and $\tilde{h}$ as in (8) that, however, requires the knowledge of the moment W at (s, ℓ).[2] In turn, W is defined using the center manifold π of the interconnected system in (3) that is defined as the unique solution of the PDE in (4). In many practical applications, retrieving π by solving (4) is not possible without relying on numerical methods that generate inevitable computational errors. Furthermore, PDE (4) is defined using the knowledge of the functions f and h that characterize the model under analysis, which are not always perfectly known without uncertainties. To address these problems, we rely on data collected either from experiments on the plant or from simulations of the interconnected system described in (3). To model the uncertainties on the output, we assume that only noisy measurements of the output y are available. In particular, we consider the noisy output $z : \mathbb{R}_{\geq 0} \to \mathbb{R}^{d_y}$ defined by

$$\forall t \in \mathbb{R}_{\geq 0}, \qquad z(t) = y(t) + e(t),$$

where $e : \mathbb{R}_{\geq 0} \to \mathbb{R}^{d_y}$ is an unknown signal that describes the noise affecting the output. Fig. 1 shows the relationship of z and e with the interconnected system in (3).

Since the signal generator is a design choice of the practitioner, we assume that s, ℓ, ω_0, and the simulated input u are completely known without uncertainty. Then, we define the available dataset as

$$\mathcal{D} := \{(\bar{t}_i, \bar{\omega}_i, \bar{z}_i)\}_{i=1}^{N} \subseteq \mathbb{R}_{\geq 0} \times \Omega \times \mathbb{R}^{d_y}, \tag{9}$$

where, for all $i \in \{1, \dots, N\}$, $N \in \mathbb{N}$, $\bar{t}_i \in \mathbb{R}_{\geq 0}$ denotes the ith sampling time, $\bar{\omega}_i = \omega(\bar{t}_i)$, and $\bar{z}_i = z(\bar{t}_i)$. Furthermore, we assume that $\bar{t}_i$ is a non-decreasing divergent sequence.

If the system (1) is not strictly proper, the moment W is defined by the functions h, π and ℓ (see Definition 2). Since ℓ is known, it is more convenient to learn the function $\mu : \mathbb{R}^{d_\omega} \times \mathbb{R}^{d_u} \to \mathbb{R}^{d_u}$ such that $\mu(\omega, \ell(\omega)) = W(\omega)$ instead of learning W directly. For strictly proper systems, this is not possible, so we identify W directly. To cover both cases, we define $d_q := d_\omega$ if the system is strictly proper, or $d_q := d_\omega + d_u$ otherwise. We also introduce the function $q : \mathbb{R}^{d_\omega} \to \mathbb{R}^{d_q}$ such that

$$q(\omega) = \begin{cases} \omega & \text{if } d_q = d_\omega, \\ (\omega, \ell(\omega)) & \text{if } d_q = d_\omega + d_u. \end{cases} \tag{10}$$

Then, we define the function $\mu : \mathbb{R}^{d_q} \to \mathbb{R}^{d_y}$ such that $W(\omega) = \mu(q(\omega))$. For compactness, for all $i \in \{1, \dots, N\}$, we also define $\bar{q}_i = q\left(\bar{\omega}_i\right) \in \mathbb{R}^{d_q}$.

The objective of this work is to construct an algorithm that, given the signal generator and the dataset $\mathcal{D}$, automatically approximates the function μ from which we can reconstruct the moment W of the system in (1). In turn, the estimated μ can be used to define a model that achieves moment matching as in (7) with $\tilde{f}$ and $\tilde{h}$ as in (8). In particular, we propose to model the function μ using an ANN that can be trained using the available dataset $\mathcal{D}$. We stress that the functions f, h, π, and W, and the initial condition x_0 are assumed unknown.

4. Neural networks for moment estimation

Given that we are interested in learning a static mapping μ with domain $\mathbb{R}^{d_q}$ and codomain $\mathbb{R}^{d_y}$, we resort to *Feed Forward Neural Networks (FFNNs)*. A deep FFNN with $L \in \mathbb{N}$ layers amounts to a sequence of compositions (Pillonetto et al., 2025)

$$\eta = \eta^{(L)} \circ \dots \circ \eta^{(1)}, \tag{11}$$

where $\eta^{(l)} : \mathbb{R}^{n_{l-1}} \to \mathbb{R}^{n_l}$, $l \in \{1, \dots, L\}$, is the mapping from the inputs of the lth layer to its outputs, $n_l \in \mathbb{N}$ being its number of units. Notationwise, $\zeta_l \in \mathbb{R}^{n_l}$ represents the outputs of the lth layer (or, analogously, the inputs of the $(l+1)$-th layer). Further, we also introduce the *input layer*, which is such that ζ_0 is the input of the network and $n_0 = d_q$. Instead, the last layer is referred to as the *output layer* and is such that

[1] The equilibrium in the origin is a stable equilibrium (in the sense of Lyapunov) and each initial state is stable in the sense of Poisson, see Isidori (1995, Ch 8.1).

[2] We remark that there exist other possible ways to define $\tilde{f}$ and $\tilde{h}$ for (7) using the moment W, see Simard et al. (2025) for more details.

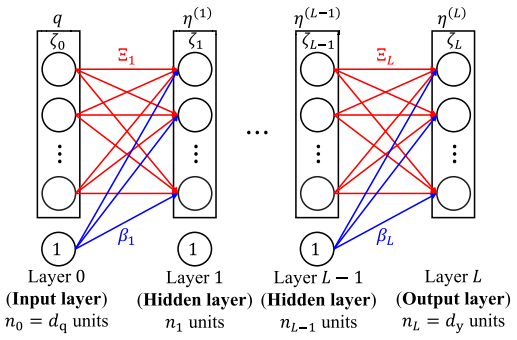


Fig. 2. Deep feed forward neural network architecture.

Table 1
Examples of activation functions ϕ_l in (12) (Hendrycks & Gimpel, 2023; Pillonetto et al., 2025). The reported formulas are to be interpreted as applied element-wise. In the Table, $\Phi(p)$ is the standard cumulative Gaussian distribution evaluated at $p \in \mathbb{R}$.

Activation function	Formula
Rectified linear unit (ReLU)	$\max(p, 0)$
Gaussian error linear unit (GeLU)	$p\,\Phi(p)$
Hyperbolic tangent (tanh)	$\frac{\exp(p)-\exp(-p)}{\exp(p)+\exp(-p)}$
Sigmoid	$\frac{1}{1+\exp(-p)}$

ζ_L is the output of the network and $n_L = d_y$. Lastly, the remaining layers for $l \in \{1, \dots, L-1\}$ are typically referred to as *hidden layers* to distinguish them from the input and output layers. Fig. 2 depicts the architecture of the considered deep feed forward neural network.

Each layer (except the input layer) applies an affine transformation to its inputs, followed by a nonlinear transformation $\phi_l : \mathbb{R}^{n_l} \to \mathbb{R}^{n_l}$ (the so-called *activation function*), *i.e.*, for any $l \in \{1, \dots, L-1\}$,

$$\zeta_l = \eta^{(l)}(\zeta_{l-1}) = \phi_l\left(\Xi_l \zeta_{l-1} + \beta_l\right), \tag{12}$$

where $\Xi_l \in \mathbb{R}^{n_l \times n_{l-1}}$ is the *weight matrix* of the lth layer while $\beta_l \in \mathbb{R}^{n_l}$ is its *bias vector*. Common activation functions amount to the element-wise application of a nonlinear mapping, such as the ones reported in Table 1. The only exception is the output layer, for which no nonlinear transformation is employed, *i.e.*, $\zeta_L = \eta^{(L)}(\zeta_{L-1}) = \Xi_L \zeta_{L-1} + \beta_L$.

To tune the neural network model η in (11), we need to select the *learnable parameters* $\theta := \{(\Xi_l, \beta_l) : l \in \{1, \dots, L\}\}$ and several *hyperparameters*, denoted as γ, among which there are the number of layers L, and the number of units per hidden layer, *i.e.*, $n_1, \dots, n_{L-1}$. The approach used to estimate the parameters θ is explained in Section 5.1 while the selection and formal definition of γ is explained in Section 5.2. Whenever needed, to highlight the dependency of η in (11) on θ and γ, we will write $\eta(\zeta_0; \theta, \gamma)$.

In general, the just-described FFNN cannot be used for moment matching purposes because, for some parameters and activation function choices, η in (11) is not a valid moment that can be used to correctly reconstruct a reduced model about the considered equilibrium of the system. For this reason, we introduce the following Proposition to formalize the conditions on η that ensure that it is a valid moment function.

Proposition 1. *The FFNN η in (11) is a valid moment for the system (1) only if, for all layers $l \in \{1, \dots, L\}$, the following conditions hold:* ***C1*** *the activation function ϕ_l is such that $\phi_l(0) = 0$,* ***C2*** *the activation function ϕ_l is smooth, and* ***C3*** *the bias vector β_l is the zero vector.*

Proof. First, recall that $\pi(0) = 0$, $\ell(0) = 0$ and $h(0, 0) = 0$ as assumed in Section 2.2. Therefore, from Definition 2, we have that $W(0) = h(\pi(0), \ell(0)) = \mu(0) = 0$. Hence, η in (11) is a valid moment only if $\eta(0) = 0$. We proceed by induction. Assuming that $\zeta_{l-1} = 0$ for any $l \in \{2, \dots, L\}$, from Conditions **C1** and **C3**,

$$\zeta_l = \phi_l\left(\Xi_l \zeta_{l-1} + \beta_l\right) = \phi_l\left(\Xi_l \cdot 0 + 0\right) = \phi_l(0) = 0.$$

By noting that $\zeta_0 = 0$ because ζ_0 is the input of the neural network η, we conclude that $\eta(0) = 0$ by induction.

Since the functions f, ℓ and s are smooth by definition, the solution π of (4) is also smooth. Then, the functions W and μ are also smooth because the function h is smooth by definition. Thus, η in (11) is a valid moment only if it is smooth. Since η is defined as the composition of linear affine functions (smooth by definition) and activation functions (smooth for Condition **C2**), we finish the proof by stating that η is smooth. □

To satisfy the conditions of Proposition 1, we can notice that the ReLU, GeLU, and tanh functions in Table 1 are all such that $\phi_l(0) = 0$ (Condition **C1**). However, among those three, only the GeLU and the tanh are smooth, making them viable alternatives to satisfy Condition **C2**. Finally, the satisfaction of Condition **C3** is explained in Section 5.1.

5. Neural network training

Before explaining the training procedure, we note that the noisy output z in Fig. 1 can be rewritten using the moment W of the system and the transient of the output y. In particular, by noting that $y_{ss}(t) = W(\omega(t))$, for any $t \in \mathbb{R}_{\geq 0}$, we have $z(t) - W(\omega(t)) = e(t) + \tau(t)$, where $\tau(t)$ converges to 0 as shown in (5). Therefore, the idea is to train the FFNN to minimize the difference between the measurements $\bar{z}_i$ and $\eta(\bar{q}_i) \approx W(\bar{\omega}_i)$. Ideally, we are interested in selecting θ and γ for (11) such that the *Mean Squared Error (MSE)* defined as

$$\mathrm{MSE}(\theta; \gamma, \mathcal{D}) := \frac{1}{|\mathcal{D}|} \sum_{(\bar{\imath}_i, \bar{\omega}_i, \bar{z}_i) \in \mathcal{D}} \|\bar{z}_i - \eta(\bar{q}_i; \theta, \gamma)\|^2,$$

is small for every possible dataset $\mathcal{D}$.

5.1. Learnable parameters estimation

First, we consider the problem of estimating the parameters θ of the neural network in (11) assuming to know the value of the hyperparameters γ. For this purpose, we partition the dataset $\mathcal{D}$ into two subsets: the training set $\mathcal{D}_{\mathrm{tr}}$, and the validation set $\mathcal{D}_{\mathrm{val}}$. The parameters θ are found by minimizing the cost function

$$J'(\theta; \gamma, \mathcal{D}_{\mathrm{tr}}) := \mathrm{MSE}(\theta; \gamma, \mathcal{D}_{\mathrm{tr}}) + \lambda \sum_{l=1}^{L} \|\Xi_l\|_{\mathrm{F}}^2, \tag{13}$$

that is the sum of the MSE computed on $\mathcal{D}_{\mathrm{tr}}$ and a *regularization term* on the weight matrices; $\lambda \in \mathbb{R}_{\geq 0}$ is a hyperparameter that weighs the two terms.

This optimization problem is typically solved through *mini-batch gradient methods* such as Stochastic Gradient Descent (SGD), AdaGrad, RMSProp, or Adam (Goodfellow et al., 2016, Ch 8). All these methods are variants of the classic gradient descent algorithm where the gradient is computed using only a subset of the data in $\mathcal{D}_{\mathrm{tr}}$. In particular, $\mathcal{D}_{\mathrm{tr}}$ is randomly partitioned into $N_{\mathrm{mb}} \in \mathbb{N}$ subsets (the so-called *mini-batches*) of roughly the same size and denoted as $\mathcal{D}_{\mathrm{tr}}^{(b)}$, $b \in \{1, \dots, N_{\mathrm{mb}}\}$. Then, starting from an initial set of parameters $\theta^{(0)}$, θ is updated iteratively following

$$\theta^{(k)} = \theta^{(k-1)} - \Psi\left(\nabla_\theta J'\left(\theta^{(k-1)}; \gamma, \mathcal{D}_{\mathrm{tr}}^{(b)}\right), \varepsilon\right), \tag{14}$$

where $k \in \mathbb{N}$ is the iteration number, $\varepsilon \in \mathbb{R}_{\geq 0}$ is the *learning rate*,[3] and Ψ is an update rule that depends on the method used.

[3] We omit other possible hyperparameters (see Goodfellow et al., 2016, Ch 8) since the learning rate is usually the most impactful.

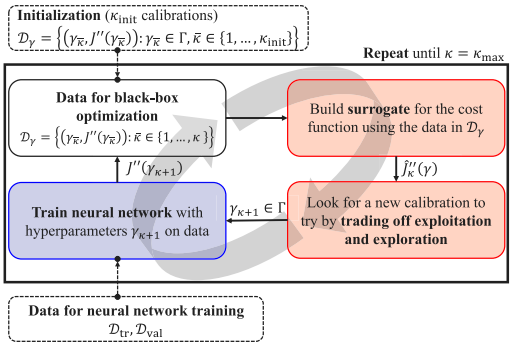


Fig. 3. Summary of the steps taken by a surrogate-based black-box optimization procedure for neural network hyperparameters calibration. White boxes represent data, red boxes are operations related to the BBO method, and blue boxes concern FFNN training.

For example, in the case of SGD, the update rule simply amounts to $\Psi\left(\nabla_\theta J'\left(\theta^{(k-1)}; \gamma, \mathcal{D}_{\text{tr}}^{(b)}\right), \varepsilon\right) = \varepsilon \nabla_\theta J'\left(\theta^{(k-1)}; \gamma, \mathcal{D}_{\text{tr}}^{(b)}\right)$, *i.e.*, the parameters are updated by taking a step from $\theta^{(k-1)}$ in the opposite direction of the gradient of the cost function J' (approximated using only a subset of the data in $\mathcal{D}_{\text{tr}}$) and with magnitude scaled by the learning rate ε. The mini-batches $\mathcal{D}_{\text{tr}}^{(b)}$, $b \in \{1, \dots, N_{\text{mb}}\}$, are sequentially processed so that going through the whole data in $\mathcal{D}_{\text{tr}}$ takes N_{mb} iterations. When that happens, we say that a training *epoch* has passed. The optimization procedure is stopped once a maximum number of iterations $K_{\max} \in \mathbb{N}$ is reached. As an implicit regularization mechanism, we also employ *early stopping* (Pillonetto et al., 2025) by computing the MSE on the validation set every $K_{\text{val}} \in \mathbb{N}$ iterations, *i.e.*, $\text{MSE}\left(\theta^{(k)}; \gamma, \mathcal{D}_{\text{val}}\right)$ for $k \in \{K_{\text{val}}, 2K_{\text{val}}, \dots, K_{\max}\}$.[4] Then, the optimization solver returns the set of parameters that performs best on $\mathcal{D}_{\text{val}}$, namely

$$\theta^*(\gamma, \mathcal{D}) := \underset{k \in \{K_{\text{val}}, 2K_{\text{val}}, \dots, K_{\max}\}}{\text{argmin}} \text{MSE}\left(\theta^{(k)}; \gamma, \mathcal{D}_{\text{val}}\right). \tag{15}$$

We point out that, due to the nested nonlinearities in (11), the cost function $J'(\theta; \gamma, \mathcal{D}_{\text{tr}})$ in (13) is both nonconvex and multimodal (Pillonetto et al., 2025). However, given the *overparametrized* nature of neural networks, most often exhibiting more parameters than training data points, we are not necessarily interested in finding a local minimum of $J'(\theta; \gamma, \mathcal{D}_{\text{tr}})$ with a high degree of accuracy. Rather, we aim to achieve a low generalization error (as approximated by $\text{MSE}\left(\theta; \gamma, \mathcal{D}_{\text{val}}\right)$), which might not necessarily be associated with a minimizer of (13) that, instead, could overfit the training data but perform badly out-of-sample.

Concerning Proposition 1, Condition **C3** can be satisfied by initializing the biases $\beta_l = 0, \forall l \in \{1, \dots, L\}$ (which is a common choice (Goodfellow et al., 2016, Ch 8)), and avoiding their update entirely in (14). This can easily be done in most deep learning libraries, such as MATLAB's Deep Learning Toolbox.

5.2. Hyperparameters calibration

A correct choice of the hyperparameters γ significantly impacts the performance of the neural network model η in (11). Apart from the hyperparameters mentioned in Section 4, *i.e.*, L and n_l, $l \in \{1, \dots, L-1\}$, the regularization weight λ in (13) and the learning rate ε in (14) also play a key role. For this reason, we propose an optimization strategy for their calibration.

In what follows, rather than selecting the number of units for each hidden layer, we resort to a *scaling hyperparameter* $\rho \in (0, 1]$ and define $n_l := \lceil \rho n_{l-1} \rceil$ for every layer $l \in \{2, \dots, L-1\}$ allowing to consider only n_1 and ρ as hyperparameters instead of $n_1, \dots, n_{L-1}$. Therefore, the set of hyperparameters γ is defined as $\gamma := \{L, n_1, \rho, \lambda, \varepsilon\}$.

In this work, we propose to tune γ by solving the following optimization problem:

$$\underset{\gamma}{\text{argmin}}\ J''(\gamma), \tag{16a}$$

$$\text{s.t.}\quad \gamma \in \Gamma, \tag{16b}$$

$$J''(\gamma) := \text{MSE}\left(\theta^*(\gamma, \mathcal{D}); \gamma, \mathcal{D}_{\text{val}}\right), \tag{16c}$$

where $\Gamma = \mathcal{L} \times \mathcal{N} \times \text{P} \times \Lambda \times \mathcal{E}$ is a set of constraints on the hyperparameters, each of which has its set of feasible values $\mathcal{L} \subset \mathbb{N}$, $\mathcal{N} \subset \mathbb{N}$, $\text{P} \subset (0, 1]$, $\Lambda \subset \mathbb{R}_{\geq 0}$, and $\mathcal{E} \subset \mathbb{R}_{\geq 0}$ for L, n_1, ρ, λ, and ϵ, respectively. The cost function J'' is the best validation MSE obtained during neural network training with a certain set of hyperparameters γ, see (15).

Given the *time-consuming* nature of the evaluation of J'' in (16c), requiring training a FFNN from scratch, we suggest resorting to *surrogate-based black-box optimization algorithms* (Regis, 2020; Vu et al., 2017) (Previtali, 2024, Ch 3), such as those belonging to the Bayesian optimization framework (Frazier, 2018), to solve the optimization problem. The goal of BBO methods is to find a sufficiently accurate global solution of Problem (16) within a limited number of cost function evaluations $\kappa_{\max} \in \mathbb{N}$ (the so-called *budget*). In short, these algorithms start from a set of $\kappa_{\text{init}} \in \mathbb{N}$, $\kappa_{\text{init}} < \kappa_{\max}$, feasible calibrations for γ, usually found via Latin Hypercube Designs (LHD) (McKay, Beckman, & Conover, 2000), coupled with their respective cost

$$\mathcal{D}_\gamma := \{(\gamma_\kappa, J''(\gamma_\kappa)) \ : \ \gamma_\kappa \in \Gamma, \kappa \in \{1, \dots, \kappa_{\text{init}}\}\}.$$

Then, at each iteration $\kappa \in \{\kappa_{\text{init}}, \dots, \kappa_{\max} - 1\}$, BBO methods build an approximation of J'' in (16c) that is cheap to evaluate, *i.e.*, the so-called *surrogate*, which is denoted as $\hat J''_\kappa$. The surrogate is subsequently used to find a new feasible calibration $\gamma_{\kappa+1} \in \Gamma$ with a relatively cheap global optimization problem that trades off *exploitation*, *i.e.*, minimizing $\hat J''_\kappa(\gamma)$ as a proxy for $J''(\gamma)$ in (16c), and *exploration*, *i.e.*, probing those regions of Γ where relatively few hyperparameters calibrations have been tested. The iteration ends by adding the couple $(\gamma_{\kappa+1}, J''(\gamma_{\kappa+1}))$ to $\mathcal{D}_\gamma$. This whole process is repeated until $\kappa_{\max}$ hyperparameter calibrations have been tried and the best tuning γ^* in $\mathcal{D}_\gamma$ is selected as the hyperparameters of the neural network. For further clarity, Fig. 3 summarizes this workflow. The interested reader is referred to Previtali (2024, Ch 3) for a detailed dissertation on the topic.

6. Numerical validation

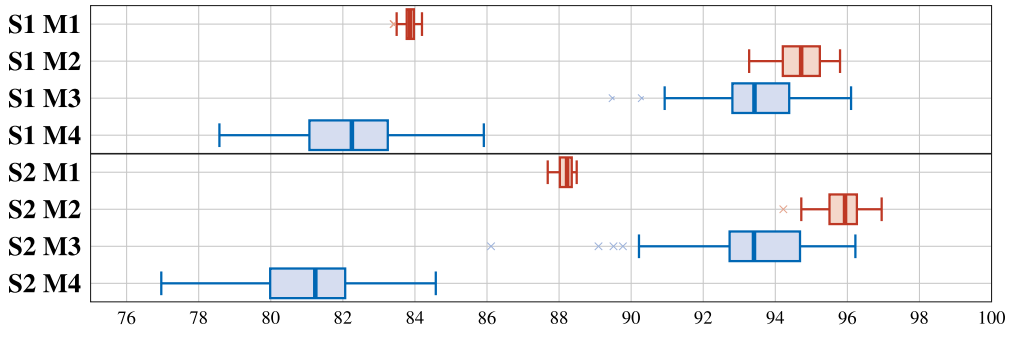
In this section, we validate the approach based on neural networks by comparing it with the current state of the art. In particular, we consider two different benchmark models to be reduced:

S1. The nonlinear RC ladder benchmark (The MORwiki Community, 2018, Model 1) or (Rewienski & White, 2003). The model is described by the equations

$$\begin{aligned}
\dot x_1 &= -g(x_1) - g(x_1 - x_2) + u,\\
\dot x_i &= g(x_{i-1} - x_i) - g(x_i - x_{i+1}),\ \forall i \in \{2, \dots, 24\},\\
\dot x_{25} &= g(x_{24} - x_{25}),
\end{aligned}$$

where, for all $i \in \{1, \dots, 25\}$, $x_i \in \mathbb{R}$ is the ith unmeasured state (the voltage in a certain node of the electrical circuit), $y(t) = x_1(t) \in \mathbb{R}$ is the output voltage, and $u(t) \in \mathbb{R}$ is the input voltage. The function $g : \mathbb{R} \to \mathbb{R}$ is such that $g(x) := x + \exp(40x) - 1$. In the simulations, the initial conditions of the states are mutually independent normal random variables with mean 0 and variance 0.004. The noise e is zero-mean, normally distributed, and stationary, with variance $2.5 \cdot 10^{-4}$ (signal-to-noise ratio of around 10).

[4] Without loss of generality, we assume that $K_{\max}$ is a multiple of K_{val}.



	M1	M2	M3	M4
S1	83.863	**94.717**	93.422	82.251
S2	88.222	**95.935**	93.408	81.236

Fig. 4. Box plots of the performance indices for all considered cases. The state-of-the-art methods are depicted in red while the proposed neural network models are shown in blue. The Fig. also reports the median Fit for each case in the table, highlighting the best results with a bold font.

S2. The nonlinear spring-mass-damper system described in Ionescu and Astolfi (2013). The model is described by the equations

$$\dot{x}_1 = x_3, \qquad \dot{x}_2 = x_4 + u,$$
$$\dot{x}_3 = x_2 - 2x_1, \qquad \dot{x}_4 = x_1 - x_2 + x_4(u - x_4^2 - 1),$$
$$y = x_4^2 + x_2 - x_1.$$

In the simulations, the initial conditions of the states are mutually independent normal random variables with mean 0 and variance 2. The noise e is zero-mean, normally distributed, and stationary, with variance $1.5{\cdot}10^{-2}$ (signal-to-noise ratio of around 10).

Regarding **S1**, we consider the problem of estimating the moment at (s_1, ℓ_1) where

$$s_1(\omega_1, \omega_2) = \begin{bmatrix} \omega_2 + \omega_2\omega_1 \cos(\omega_2)\sin(\omega_1^2) \\ -\omega_1 - \omega_1^2 \cos(\omega_2)\sin(\omega_1^2) \end{bmatrix},$$
$$\ell_1(\omega_1, \omega_2) = 5\omega_2,$$

with initial condition $\omega_1(0) = 0.8$ and $\omega_2(0) = 0.7$. Instead, for **S2**, we consider the problem of estimating the moment at (s_2, ℓ_2) where

$$s_2(\omega_1, \omega_2) = \begin{bmatrix} \frac{4}{3}\omega_1 - \frac{5}{3}\omega_1\omega_2 \\ -\frac{1}{5}\omega_2 + \frac{3}{4}\omega_1\omega_2 \end{bmatrix},$$
$$\ell_2(\omega_1, \omega_2) = \omega_1,$$

with initial condition $\omega_1(0) = \omega_2(0) = 1$. After computing the estimate of the moment $\tilde{W}$, we define the reduced model as in (7) with $\tilde{h}(\omega, u) = \tilde{W}(\omega)$ and $\delta(\omega_1, \omega_2) = [10\omega_1^2, 10]^\top$ for **S1**, or $\delta(\omega_1, \omega_2) = [\frac{7}{3}, 1]^\top$ for **S2** (see (8)).

In all the experiments, we collect $N = 1500$ uniformly sampled data points in a time window Δ, leading to the dataset $\mathcal{D}$ in (9). In particular, $\Delta = [2, 7]$ for **S1** or $\Delta = [400, 450]$ for **S2**. We remark that, in both cases, we only consider steady-state data, making the output of the systems dependent only on the moment and the additive noise (see Section 5).

We compare the moments estimated with the following algorithms:

M1. The algorithm proposed in Scarciotti and Astolfi (2017) for nonlinear systems with polynomial features up to degree 5;

M2. The algorithm proposed in Moreschini, Scandella et al. (2024) with the kernel introduced by the authors. The hyperparameters of the method are selected using Monte Carlo k-fold cross-validation (Xu & Liang, 2001) with 10 folds and 5 repetitions.

M3. The neural network model proposed in Section 4 with the GeLU, defined in Table 1, as activation function for each hidden layer, and $\beta_l = 0$, $\forall l \in \{1, \dots, L\}$, to satisfy all the conditions of Proposition 1. For FFNN parameters estimation purposes (Section 5), 20% of the data (300 points) is reserved for $\mathcal{D}_{\text{val}}$, while the remaining 80% (1200 points) is used for training. As customary, the inputs are standardized beforehand. The FFNN is trained for $K_{\max} = 5000$ iterations with a mini-batch size of 256 ($N_{\text{mb}} = 5$ mini-batches in total) using the RMSProp optimizer (Goodfellow et al., 2016, Ch 8). The performances on $\mathcal{D}_{\text{val}}$ are checked every $K_{\text{val}} = 5$ iterations (*i.e.*, every epoch). The constraint sets for Problem (16) amount to $\mathcal{L} = \{2, 3, 4, 5\}$, $\mathcal{N} = \{20, 30, 40\}$, $\mathrm{P} = [0.1, 1]$, $\Lambda = [10^{-6}, 10^{-1}]$, and $\mathcal{E} = [10^{-5}, 10^{-1}]$. We employ the GLIS-r (Previtali, 2024, Ch 5) BBO algorithm[5] to select γ with a budget $\kappa_{\max} = 100$ and $\kappa_{\text{init}} = 50$ initial calibrations generated by a LHD (McKay et al., 2000).

M4. The neural network model proposed in Section 4 with the tanh, defined in Table 1, as activation function for each hidden layer, and $\beta_l = 0$, $\forall l \in \{1, \dots, L\}$, in order to satisfy all the conditions of Proposition 1. The remaining settings are the same as **M3**.

To statistically compare the four methods, we have performed 51 Monte Carlo experiments in which all the aforementioned random variables have different realizations. The results are reported in Fig. 4, where we assess the performance with the index

$$\text{Fit} = 1 - \frac{\sum_{i=1}^{N_{\text{tst}}} \|y(\bar{t}_i) - \tilde{y}(\bar{t}_i)\|}{\sum_{i=1}^{N_{\text{tst}}} \|y(\bar{t}_i) - \frac{1}{N_{\text{tst}}}\sum_{j=1}^{N_{\text{tst}}} y(\bar{t}_j)\|} \tag{17}$$

with y the steady-state output of the benchmark model and $\tilde{y}$ the steady-state output of the estimated reduced model, $N_{\text{tst}} = 10^4$, and $t_i = 10 + (i-1)\frac{5}{N_{\text{tst}}-1}$ for **S1**, or $t_i = 600 + (i-1)\frac{110}{N_{\text{tst}}-1}$ for **S2**, $i \in \{1, \dots, N_{\text{tst}}\}$. From the Fig., we can note that the most accurate models on both benchmarks are the state-of-the-art kernel method in Moreschini, Scandella et al. (2024), *i.e.* **M2**, and the proposed neural network approach with the GeLU activation function, *i.e.* **M3**, achieving Fits above 90% in most Monte Carlo experiments and with a difference of at most 2.5% median-wise. In contrast, **M1** and **M4** exhibit noticeably lower performance than the other two methods, with differences in median Fit that can even exceed 10%. We can conclude that the activation function is a key

[5] In practice, GLIS-r (Previtali, 2024, Ch 5), like many BBO methods, only handles real optimization variables. Consequently, we solve Problem (16) with $L \in \mathcal{L}$ and $n_1 \in \mathcal{N}$ fixed, optimizing only with respect to ρ, λ, and ϵ. This process is repeated for each considered L and n_1, essentially carrying out an exhaustive search with respect to the integer variables.

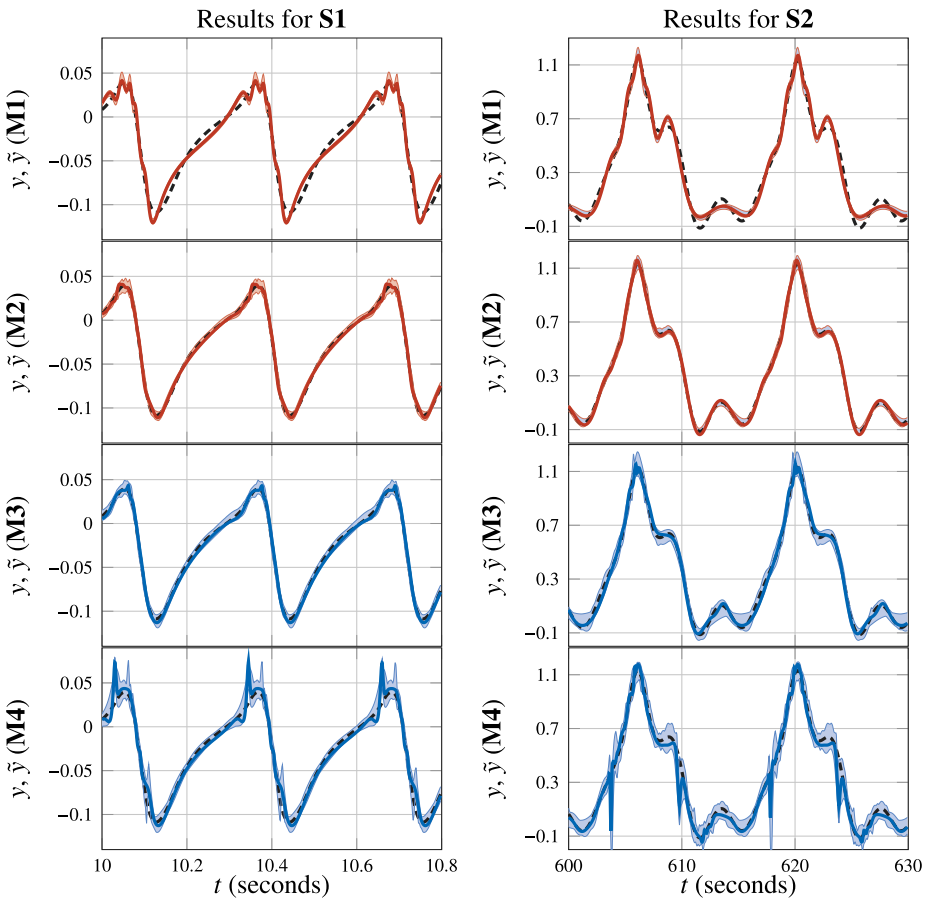


Fig. 5. Time history of the output of the benchmark models (dashed lines), the output of the estimated models with the median performance within the Monte Carlo experiments (solid lines), and the range of all the Monte Carlo experiments (shaded areas). The state-of-the-art methods are depicted in red while the proposed neural network models are shown in blue.

choice for our proposal, given that the GeLU significantly outperforms the tanh in our experiments.

As further validation, in Fig. 5, we show the outputs $\tilde{y}$ of the reduced model in (7) using the estimated moment against the output y of the benchmark model. In particular, we display the trajectories of the reduced models that achieve the median Fit in (17), along with the minimum and maximum of $\tilde{y}(\bar{t}_i)$ at each $i \in \{1, \dots, N_{\text{tst}}\}$. Fig. 5 confirms the high Fits of **M2** and **M3**, highlighting how both methods closely follow the true outputs in all Monte Carlo experiments. Instead, **M1** is unable to track the y's accurately in some portions of the considered benchmarks, while **M4** exhibits undesirable peaks at some sampling times.

7. Concluding remarks

In this preliminary work, we investigate the suitability and effectiveness of using neural networks in the context of data-driven moment matching for nonlinear systems. In particular, we introduce a neural network architecture capable of modeling the moments of nonlinear systems, demonstrating its suitability for this application. To evaluate the effectiveness of our approach, we compare the accuracy of the reduced model defined using the proposed neural network architecture with state-of-the-art data-driven moment matching methods on two benchmark models. The results confirm the capability of the neural network approach to approximate the moments of nonlinear systems, achieving performance comparable to that of the regularized kernel method proposed in Moreschini, Scandella et al. (2024).

Future work includes testing the proposed approach on additional benchmarks and exploring alternative neural network architectures. Moreover, it is worth investigating the theoretical properties of the neural-network-based reduced model and the estimated center manifold.

CRediT authorship contribution statement

Matteo Scandella: Writing – review & editing, Writing – original draft, Supervision, Software, Methodology, Formal analysis, Conceptualization. **Davide Previtali:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology. **Alessio Moreschini:** Writing – review & editing, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The author is an Editorial Board Member/Editor-in-Chief/Associate Editor/Guest Editor for this journal and was not involved in the editorial review or the decision to publish this article.

References

Antoulas, A. C. (2005). *Approximation of large-scale dynamical systems*. SIAM, http://dx.doi.org/10.1137/1.9780898718713.

Astolfi, A. (2010). Model reduction by moment matching for linear and nonlinear systems. *IEEE Transactions on Automatic Control*, *55*(10), 2321–2336. http://dx.doi.org/10.1109/TAC.2010.2046044.

Astolfi, A., Beck, C., Bhattacharjee, D., Kawano, Y., Moreschini, A., Sandberg, H., et al. (2024). Forty plus years of model reduction and still learning. In *Proc. 63rd IEEE Conference on Decision and Control (CDC)* (pp. 4480–4493). http://dx.doi.org/10.1109/CDC56724.2024.10886060.

Benner, P., Gugercin, S., & Willcox, K. (2015). A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, *57*(4), 483–531. http://dx.doi.org/10.1137/130932715.

Bhattacharjee, D., Moreschini, A., & Astolfi, A. (2025). Signal generator agnostic moment matching. *IEEE Transactions on Automatic Control*, 1–16. http://dx.doi.org/10.1109/TAC.2025.3576063.

Carr, J. (2012). *Applications of centre manifold theory*. Springer, http://dx.doi.org/10.1007/978-1-4612-5929-9.

Frazier, P. I. (2018). A tutorial on Bayesian optimization. arXiv preprint arXiv:1807.02811.

Glover, K. (1984). All optimal Hankel-norm approximations of linear multivariable systems and their L^∞-error bounds. *International Journal of Control*, *39*(6), 1115–1193. http://dx.doi.org/10.1080/00207178408933239.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Gosea, I. V., & Antoulas, A. C. (2018). Data-driven model order reduction of quadratic-bilinear systems. *Numerical Linear Algebra with Applications*, *25*(6), Article e2200. http://dx.doi.org/10.1002/nla.2200.

Haasdonk, B., Hamzi, B., Santin, G., & Wittwar, D. (2021). Kernel methods for center manifold approximation and a weak data-based version of the center manifold theorem. *Physica D. Nonlinear Phenomena*, *427*, Article 133007. http://dx.doi.org/10.1016/j.physd.2021.133007.

Hendrycks, D., & Gimpel, K. (2023). Gaussian error linear units (GELUs). arXiv preprint arXiv:1606.08415.

Hermann, R., & Krener, A. (1977). Nonlinear controllability and observability. *IEEE Transactions on Automatic Control*, *22*(5), 728–740. http://dx.doi.org/10.1109/TAC.1977.1101601.

Ionescu, T. C., & Astolfi, A. (2013). Moment matching for nonlinear port Hamiltonian and gradient systems. In *Proc. 9th IFAC Symposium on Nonlinear Control Systems, Vol. 46* (pp. 395–399). http://dx.doi.org/10.3182/20130904-3-FR-2041.00160.

Ionescu, T. C., & Astolfi, A. (2015). Nonlinear moment matching-based model order reduction. *IEEE Transactions on Automatic Control*, *61*(10), 2837–2847. http://dx.doi.org/10.1109/TAC.2015.2502187.

Isidori, A. (1995). *Nonlinear control systems* (3rd ed.). Springer, http://dx.doi.org/10.1007/978-1-84628-615-5.

Kawano, Y., & Scherpen, J. M. A. (2017). Model reduction by differential balancing based on nonlinear Hankel operators. *IEEE Transactions on Automatic Control*, *62*(7), 3293–3308. http://dx.doi.org/10.1109/TAC.2016.2628201.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. http://dx.doi.org/10.1038/nature14539.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, *234*, 11–26. http://dx.doi.org/10.1016/j.neucom.2016.12.038.

Mao, J., & Scarciotti, G. (2024). Data-driven model reduction by two-sided moment matching. *Automatica*, *166*, Article 111702. http://dx.doi.org/10.1016/j.automatica.2024.111702.

Mayo, A. J., & Antoulas, A. C. (2007). A framework for the solution of the generalized realization problem. *Linear Algebra and its Applications*, *425*(2), 634–662. http://dx.doi.org/10.1016/j.laa.2007.03.008.

McKay, M. D., Beckman, R. J., & Conover, W. J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, *42*(1), 55–61. http://dx.doi.org/10.1080/00401706.2000.10485979.

Moreschini, A., & Astolfi, A. (2025). Closed-loop interpolation by moment matching for linear and nonlinear systems. *IEEE Transactions on Automatic Control*, *70*(5), 2918–2933. http://dx.doi.org/10.1109/TAC.2024.3484309.

Moreschini, A., Scandella, M., & Parisini, T. (2024). Nonlinear data-driven moment matching in reproducing kernel Hilbert spaces. In *Proc. European Control Conference (ECC)* (pp. 3440–3445). http://dx.doi.org/10.23919/ECC64448.2024.10590737.

Moreschini, A., Simard, J. D., & Astolfi, A. (2023). Model reduction in the Loewner framework for second-order network systems on graphs. In *Proc. 62nd IEEE Conference on Decision and Control (CDC)* (pp. 6713–6718). http://dx.doi.org/10.1109/CDC49753.2023.10383794.

Moreschini, A., Simard, J. D., & Astolfi, A. (2024). Data-driven model reduction for port-Hamiltonian and network systems in the Loewner framework. *Automatica*, *169*, Article 111836. http://dx.doi.org/10.1016/j.automatica.2024.111836.

Padoan, A. (2023). Model reduction by least squares moment matching for linear and nonlinear systems. *IEEE Transactions on Automatic Control*, *68*(12), 8267–8274. http://dx.doi.org/10.1109/TAC.2023.3294869.

Pillonetto, G., Aravkin, A., Gedon, D., Ljung, L., Ribeiro, A. H., & Schön, T. B. (2025). Deep networks for system identification: a survey. *Automatica*, *171*, Article 111907. http://dx.doi.org/10.1016/j.automatica.2024.111907.

Previtali, D. (2024). *Surrogate-based methods for black-box and preference-based optimization in control systems* (Ph.D. thesis), University of Bergamo, http://dx.doi.org/10.13122/978-88-97413-93-6.

Quiñonero-Candela, J., & Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, *6*(65), 1939–1959.

Regis, R. G. (2020). A survey of surrogate approaches for expensive constrained black-box optimization. In *Optimization of complex systems: theory, models, algorithms and applications* (pp. 37–47). Springer, http://dx.doi.org/10.1007/978-3-030-21803-4_4.

Rewienski, M., & White, J. (2003). A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *22*(2), 155–170. http://dx.doi.org/10.1109/TCAD.2002.806601.

Rowley, C. W., & Dawson, S. T. (2017). Model reduction for flow analysis and control. *Annual Review of Fluid Mechanics*, *49*, 387–417. http://dx.doi.org/10.1146/annurev-fluid-010816-060042.

Rudi, A., Carratino, L., & Rosasco, L. (2017). FALKON: An optimal large scale kernel method. In *Advances in Neural Information Processing Systems, Vol. 30* (pp. 3888–3898).

Safonov, M. G., Chiang, R. Y., & Limebeer, D. J. N. (1990). Optimal Hankel model reduction for nonminimal systems. *IEEE Transactions on Automatic Control*, *35*(4), 496–502. http://dx.doi.org/10.1109/9.52314.

Samari, B., Sandberg, H., Johansson, K. H., & Lavaei, A. (2023). Data-driven model order reduction for continuous- and discrete-time nonlinear systems. arXiv preprint arXiv:2507.18131.

Scandella, M., Mazzoleni, M., Formentin, S., & Previdi, F. (2021). A note on the numerical solutions of kernel-based learning problems. *IEEE Transactions on Automatic Control*, *66*(2), 940–947. http://dx.doi.org/10.1109/TAC.2020.2989769.

Scarciotti, G., & Astolfi, A. (2017). Data-driven model reduction by moment matching for linear and nonlinear systems. *Automatica*, *79*, 340–351. http://dx.doi.org/10.1016/j.automatica.2017.01.014.

Scarciotti, G., & Astolfi, A. (2024). Interconnection-based model order reduction – a survey. *European Journal of Control*, *75*, Article 100929. http://dx.doi.org/10.1016/j.ejcon.2023.100929.

Schilders, W. H. A. (2011). The need for novel model order reduction techniques in the electronics industry. In *Model reduction for circuit simulation* (pp. 3–23). Springer, http://dx.doi.org/10.1007/978-94-007-0089-5_1.

Simard, J. D., Moreschini, A., & Astolfi, A. (2025). Parameterization of all differential-algebraic moment matching interpolants. *IEEE Transactions on Automatic Control*, *70*(3), 1875–1882. http://dx.doi.org/10.1109/TAC.2024.3469247.

Snowden, T. J., van der Graaf, P. H., & Tindall, M. J. (2017). Methods of model reduction for large-scale biological systems: A survey of current methods and trends. *Bulletin of Mathematical Biology*, *79*(7), 1449–1486. http://dx.doi.org/10.1007/s11538-017-0277-2.

Sussmann, H. J., & Jurdjevic, V. (1972). Controllability of nonlinear systems. *Journal of Differential Equations*, *12*(1), 95–116. http://dx.doi.org/10.1016/0022-0396(72)90007-1.

The MORwiki Community (2018). Nonlinear RC Ladder. MORwiki – Model Order Reduction Wiki. http://modelreduction.org/index.php/Nonlinear_RC_Ladder [Accessed: 2025-08-25].

Vu, K. K., D'Ambrosio, C., Hamadi, Y., & Liberti, L. (2017). Surrogate-based methods for black-box optimization. *International Transactions in Operational Research*, *24*(3), 393–424. http://dx.doi.org/10.1111/itor.12292.

Xu, Q.-S., & Liang, Y.-Z. (2001). Monte Carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, *56*(1), 1–11. http://dx.doi.org/10.1016/S0169-7439(00)00122-2.