Automatica 156 (2023) 111210

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Deep subspace encoders for nonlinear system identification☆,☆☆

Gerben I. Beintema [a,*], Maarten Schoukens [a], Roland Tóth [a,b]

[a] Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands
[b] Systems and Control Laboratory, Institute for Computer Science and Control, Budapest, Hungary

ARTICLE INFO

Article history:
Received 26 October 2022
Received in revised form 30 March 2023
Accepted 16 June 2023
Available online 31 July 2023

Keywords:
System identification
Nonlinear state–space modeling
Subspace identification
Deep learning

ABSTRACT

Using Artificial Neural Networks (ANN) for nonlinear system identification has proven to be a promising approach, but despite of all recent research efforts, many practical and theoretical problems still remain open. Specifically, noise handling and models, issues of consistency and reliable estimation under minimization of the prediction error are the most severe problems. The latter comes with numerous practical challenges such as explosion of the computational cost in terms of the number of data samples and the occurrence of instabilities during optimization. In this paper, we aim to overcome these issues by proposing a method which uses a truncated prediction loss and a subspace encoder for state estimation. The truncated prediction loss is computed by selecting multiple truncated subsections from the time series and computing the average prediction loss. To obtain a computationally efficient estimation method that minimizes the truncated prediction loss, a subspace encoder represented by an artificial neural network is introduced. This encoder aims to approximate the state reconstructability map of the estimated model to provide an initial state for each truncated subsection given past inputs and outputs. By theoretical analysis, we show that, under mild conditions, the proposed method is locally consistent, increases optimization stability, and achieves increased data efficiency by allowing for overlap between the subsections. Lastly, we provide practical insights and user guidelines employing a numerical example and state-of-the-art benchmark results.

1. Introduction

While linear system identification offers both a strongly developed theoretical framework and broadly applicable computational tools, identification of nonlinear systems remains challenging. The wide range of nonlinear behaviors that appear in engineering, reaching from mechatronic systems to chemical and biological systems, poses a challenge in developing generically applicable model structures and identification methods (Schoukens & Ljung, 2019). Hence, numerous nonlinear system identification methods have been proposed over the last decades. Amongst the most popular ones are, linear parameter-varying (Lee & Poolla, 1999; Tóth, 2010), Volterra (Birpoutsoukis, Marconato, Lataire, & Schoukens, 2017; Sliwiński, Marconato, Wachel, & Birpoutsoukis, 2017), NAR(MA)X (Billings, 2013), block-oriented (Giri & Bai, 2010; Schoukens & Tiels, 2017), and nonlinear state–space (Beintema, Tóth, & Schoukens, 2021a, 2021b; Forgione, Mejari, & Piga, 2022; Gedon, Wahlström, Schön, & Ljung, 2021; Masti & Bemporad, 2021; Paduart et al., 2010; Schön, Wills, & Ninness, 2011; Schoukens, 2021) approaches.

In this paper, we consider the problem of identifying nonlinear systems using *nonlinear state–space* (NL-SS) models since they can represent a broad range of dynamic behaviors and are well applicable for *multiple-input multiple-output* (MIMO) systems (Schoukens & Ljung, 2019). However, estimation of NL-SS models is rather challenging as the state-variables are often not measurable (hidden Markov model) and the associated optimization-based training process is prone to local minima and model/gradient instability (Decuyper, Runacres, Schoukens, & Tiels, 2020). Furthermore, the associated nonlinear state-transition and output functions rapidly grow in complexity with a growing number of states and inputs. If these are parametrized as a linear combination of basis functions, e.g., polynomials as in Decuyper, Dreesen, Schoukens, Runacres, and Tiels (2019) and Paduart et al. (2010), then this often leads to an explosion of parameters to be able to capture the system dynamics. Also,

☆ The research was partly funded by the Eötvös Loránd Research Network (Grant Number: SA-77/2021). The material in this paper was partially presented at 3rd Conference on Learning for Dynamics and Control, June 7 – 8, 2021, ETH Zurich, Switzerland, Virtual. This paper was recommended for publication in revised form by Associate Editor Dario Piga under the direction of Editor Alessandro Chiuso.

☆☆ Implementation of the proposed SUBNET method is available at https://github.com/GerbenBeintema/deepSI and the implementation of the simulation study is available at `GerbenBeintema/encoder-automatica-experiments`.

* Corresponding author.
E-mail addresses: g.i.beintema@tue.nl (G.I. Beintema), m.schoukens@tue.nl (M. Schoukens), r.toth@tue.nl (R. Tóth).

https://doi.org/10.1016/j.automatica.2023.111210

probabilistic methods such as Schön et al. (2011) can become computationally burdensome with increasing numbers of states and inputs or training sequence lengths. Hence, an efficient representation approach for the nonlinearities and a novel estimation concept is required for NL-SS identification.

Deep learning and *artificial neural networks* (ANNs) are uniquely suited to approach the NL-SS identification challenges as they have been shown theoretically and practically to be able to model complex data relations while being computationally scalable to large datasets. Although these benefits inspired the use of state–space neural network models two decades ago (Suykens, Moor, & Vandewalle, 1995), fully exploiting these properties in NL-SS identification without major downsides is still an open problem. For instance, careful initialization of the neural network weights and biases partially mitigates the risk of local minima during optimization, but requires additional information, e.g., estimating of a linear approximate model of the system (Schoukens, 2021). Additionally, Ribeiro, Tiels, Umenberger, Schön, and Aguirre (2020) have shown that multiple shooting smooths the cost function, reducing the number of local minima and improving optimization stability, which has given rise to the use of truncated simulation error cost for ANN based NL-SS estimation (Forgione & Piga, 2021). However, the use of multiple shooting approaches comes with the challenge of estimating a potentially large number of unknown initial states for each subsection, resulting in a complexity increase of the optimization. To overcome this problem, auto-encoders have been investigated to jointly estimate the model state and the underlying state–space functions using one-step-ahead prediction cost (Masti & Bemporad, 2021). However, these approaches fall short of giving accurate long-term predictions due to incorrect noise handling, they need for tuning sensitive hyperparameters in the composite auto-encoder/prediction-error loss function, and they lack of consistency guarantees.

To overcome these challenges, this paper enhances the subspace encoder-based method for identification of *state–space* (SS) neural networks first introduced in Beintema et al. (2021b) with an innovation noise model and prove consistency properties. The nonlinear SS model is parametrized with ANNs for flexibility and efficiency in representing the often complex and high-dimensional state-transition and output functions. The model is estimated under a *truncated prediction loss*, evaluated on short subsections. Similarly to multiple shooting, these subsections further improve computational scalability and optimization stability, thereby reducing the importance of parameter initialization. The internal state at the start of each subsection is obtained using a nonlinear *subspace encoder* which approximates the reconstructability map of the SS model and further improves computational scalability and data efficiency. The state-transition and output functions of the SS model and the encoder are simultaneously estimated based on the aforementioned truncated prediction loss function. Finally, *batch optimization* and *early stopping* are employed to further improve the performance of the proposed identification scheme. We demonstrate that the resulting nonlinear state–space identification method is robust w.r.t. model and gradient instability during training, has a relatively small number of hyperparameters, and obtains state-of-the-art results on benchmark examples.

To summarize, our main contributions are

- A novel ANN-based NL-SS identification algorithm that even in the presence of innovation noise disturbances provides reliable and computationally efficient data-driven modeling;
- Efficient use of multiple-shooting based formulation of the prediction loss via co-estimation of an encoder function representing the reconstructability map of the nonlinear model (computational efficiency);
- Proving that the proposed estimator is consistent (statistical validity) and enhances smoothness of the costs function (optimization efficiency);
- Guidelines for the choice of hyperparameters and a detailed comparison of the proposed method to the state-of-the-art on a widely used identification benchmark.

The paper is structured as follows: Section 2 introduces the considered data-generating system and identification problem. Section 3 discusses the proposed subspace encoder method in detail and provides some user guidelines. We theoretically prove multiple key properties of the proposed method in Section 4, and demonstrate state-of-the-art performance of the method on a simulation example and the Wiener–Hammerstein benchmark in Sections 5–6, followed by the conclusions in Section 7.

2. Problem setting and preliminaries

2.1. Data-generating system

Consider a discrete-time system with innovation noise that can be represented by the state–space description:

$$x_{k+1} = f(x_k; u_k; e_k); \tag{1a}$$

$$y_k = h(x_k) + e_k; \tag{1b}$$

where $k \in \mathbb{Z}$ is the discrete-time, e is an i.i.d. white noise process with finite variance $_{\mathrm{e}} \in \mathrm{R}^{n_y \times n_y}$, and u is a quasi-stationary input process independent of e and taking values in R^{n_u} at each time moment k. Additionally, x and y are the state and output processes, taking values in R^{n_x} and R^{n_y} respectively. The functions $f : \mathrm{R}^{n_x \times n_u \times n_y} \to \mathrm{R}^{n_x}$ and $h : \mathrm{R}^{n_x} \to \mathrm{R}^{n_y}$, i.e. the state transition and output functions, are considered to be bounded, deterministic maps. Without loss of generality we can assume that h does not contain a direct feedthrough term. By assuming various structures for f and h, many well-known noise structures can be obtained such as *nonlinear output noise* (NOE), *nonlinear auto-regressive with exogenous input* (NARX), *nonlinear auto-regressive with moving average exogenous input* (NARMAX) and *nonlinear Box–Jenkins* (NBJ) (Jansson, 2003). For instance, if f does not depend on e_k, then a NL-SS model with an OE noise structure is obtained.

For a given sampled excitation sequence $\{u_k\}_{k=1}^N$ and potentially unknown initial state $x_1 \in \mathrm{R}^{n_x}$, the obtained response of the considered system (1) in terms of a sample path realization is collected into an ordered *input–output* (IO) data set $D_N = \{(u_k; y_k)\}_{k=1}^N$ used for identification. To avoid unnecessary clutter, we will not use different notation for random variables such as y_k defined by (10) and their sampled values, but at places where confusion might arise, we will specify which notion is used.

2.2. Identification problem

Based on the given data sequence D_N, our objective is to identify the dynamic relation (1), which boils down to the estimation of f and h. Note that these functions cannot be estimated directly as x and e are not measured.

To accomplish our objective, notice that $e_k = y_k - h(x_k)$ based on (1), hence, by substitution, we get

$$x_{k+1} = f(x_k; u_k; y_k - h(x_k)) = \tilde{f}(x_k; u_k; y_k); \tag{2}$$

Then, for $n \geq 1$, we can write

$$y_k = h(x_k) + e_k; \tag{3a}$$

$$y_{k+1} = (h \circ \tilde{f})(x_k; u_k^k; y_k^k) + e_{k+1}; \tag{3b}$$

$$\vdots$$

$$y_{k+n} = (h \circ_n \tilde{f})(x_k; u_k^{k+n-1}; y_k^{k+n-1}) + e_{k+n}; \tag{3c}$$

where $\circ$ stands for function concatenation on the state argument, $\circ_n$ means n-times recursive repetition of $\circ$ (e.g., $h \circ_2 \tilde{f} = h \circ \tilde{f} \circ \tilde{f}$), and $u_k^{k+n-1} = [u_k^\top \quad \cdots \quad u_{k+n-1}^\top]^\top$ with y_k^{k+n-1} similarly defined. More compactly:

$$y_k^{k+n} = \quad_n(x_k; u_k^{k+n-1}; y_k^{k+n-1}) + e_k^{k+n}. \tag{4}$$

Note that the noise sequence e_k^{k+n} is not available in practice, hence, Eq. (4) cannot be directly used in estimation. To overcome this problem, we can exploit the i.i.d. white noise assumption on e_k and calculate the expectation of (4) w.r.t. e conditioned on the available past data and the initial state x_k:

$$\hat{y}_k^{k+n} = \mathsf{E}_e[y_k^{k+n} \mid u_k^{k+n-1}; y_k^{k+n-1}; x_k] = \quad_n(x_k; u_k^{k+n-1}; y_k^{k+n-1}); \tag{5}$$

which is the so called *one-step-ahead* predictor associated with (1) and can be computed for the entire sample path realization in D_N, i.e., $\hat{y}_1^N = \quad_N(x_1; u_1^{N-1}; y_1^{N-1})$ or, for a specific sample, as $\hat{y}_n = \quad_n(x_1; u_1^{n-1}; y_1^{n-1})$ with $\quad_n = (h \circ_n \tilde{f})$. We can exploit (5) to define the estimator by introducing a parametrized form $\quad_{N;}$ of the predictor in terms of $f \ : \mathsf{R}^{n_x \times n_u \times n_y} \to \mathsf{R}^{n_x}$ and $h \ : \mathsf{R}^{n_x} \to \mathsf{R}^{n_y}$ defined by the parameters $\in \subseteq \mathsf{R}^n$. The classical way to estimate the parameter vector based on a given data set D_N and ensure that f and h accurately represent Eq. (1) is to minimize the $\grave{}_2$ loss of the prediction error $\hat{e}_k = y_k - \hat{y}_k$ between the measured samples y_k and the predicted response $\hat{y}_k$ by $\quad_{N;}$:

$$V_{D_N}^{\text{pred}}(\ ) = \frac{1}{N} \sum_{k=1} \quad y_k - \hat{y}_k \quad_2^2; \tag{6}$$

where the initial state x_1 is a parameter which is co-estimated with . In case f does not depend on $\hat{e}_k$, which corresponds to an OE noise structure, then (6) is equal to the well-known *simulation error* loss function.

The parametrized predictor $\quad_{N;}$, can also be written in a state–space form

$$\hat{x}_{k+1} = f\ (\hat{x}_k; u_k; \hat{e}_k); \tag{7a}$$

$$\hat{y}_k = h\ (\hat{x}_k); \tag{7b}$$

where $\hat{x}$ and $\hat{y}$ are the predicted state and predicted output taking values from R^{n_x} and R^{n_y} respectively, while $\hat{e}$ is the prediction error. In fact, (7) qualifies as the model structure used to estimate Eq. (1) through the minimization of the identification criterion (6).

In the sequel, we will consider f and h to be multi-layer *artificial neural networks* (ANNs), parametrized in , where each hidden layer is composed from m activation functions $\ : \mathsf{R} \to \mathsf{R}$ in the form of $z_{i;j} = \left(\sum_{l=1}^{m_{i-1}} \quad_{w;i;j;l} z_{i-1;l} + \quad_{b;i;j}\right)$ where $z_i = \text{col}(z_{i;1}; \vdots\ ; z_{i;m_i})$ is the latent variable representing the output of layer $1 \le i \le q$. Here, col( ) denotes composition of a column vector. For f with q hidden-layers and linear input and output layers, this means $f\ (\hat{x}_k; u_k; \hat{e}_k) = \quad_{w;q+1} z_q(k) + \quad_{b;q+1}$ and $z_0(k) = \text{col}(\hat{x}_k; u_k; \hat{e}_k)$. The parameters of the state transition and output functions of (7) are collected in . Furthermore, for the remainder of this paper we will assume that f and h are Lipschitz continuous. Note that assumption is not restrictive for commonly used neural network structures since the activation functions (ReLu, tanh, sigmoid, etc.) used for are Lipschitz continuous. Under these considerations, model structure (7) represents a recurrent neural network and it is also called *state–space* (SS) ANN in the literature (Schoukens, 2021; Suykens et al., 1995).

By using the ANNs f and h, one can directly compose the feedforward predictor network $\quad_{N;}$ and attempt to solve minimization of (6) directly. However, this blunt approach can meet with considerable difficulties. In ANN-based identification, minimizing the simulation error, which is a special case of (6) under an OE noise structure, has been observed to result in accurate models (Schoukens & Ljung, 2019), but its major shortcoming is that the computational cost scales at least linearly with N. Furthermore, optimization of this cost function is sensitive to local minima and gradient-based methods commonly display unstable behavior (Ribeiro et al., 2020). Hence, the problem that we aim to solve in this paper is twofold: *(i)* achieve consistent estimation of (1) under innovation noise conditions using the parametrized SS-ANN model (7) and one-step-ahead prediction (6) and *(ii)* to provide a consistent estimator that drastically reduces the involved computational cost and ensures implementability.

3. The subspace encoder method

This section introduces the proposed subspace encoder method that addresses many of the challenges encountered when using classical prediction or simulation error identification approaches for nonlinear state–space models. The proposed approach builds on the introduction of two main ingredients: a truncated prediction loss based cost function and a subspace encoder which is linked to the concept of state reconstructability.

3.1. Truncated prediction loss

In order to overcome the computational difficulties in the minimization of (6), it is an important observation that the main difficulty comes from forward propagation of the state over the entire length of the data set. Hence in the proposed method, which is an extension of our previous work (Beintema et al., 2021b), a truncated form of the $\grave{}_2$ prediction loss is considered that emulates well the total prediction loss. This truncated form aims to reduce the computational cost by the utilization of parallel computing and to increase optimization stability (Ribeiro et al., 2020). By selecting subsections of length T (called the truncation length) in the overall time sequence, the prediction loss is calculated on the selected sections:

$$V_{D_N}^{\text{sub}}(\ ) = \frac{1}{C} \sum_{t=1}^{N-T+1} \sum_{k=0}^{T-1} \|y_{t+k} - \hat{y}_{t+k|t}\|_2^2; \tag{8a}$$

$$\hat{x}_{t+k+1|t} = f\ (\hat{x}_{t+k|t}; u_{t+k}; \hat{e}_{t+k|t}); \tag{8b}$$

$$\hat{y}_{t+k|t} = h\ (\hat{x}_{t+k|t}); \tag{8c}$$

$$\hat{e}_{t+k|t} = y_{t+k} - \hat{y}_{t+k|t}; \tag{8d}$$

where the pipe (|) notation is introduced to distinguish between subsections as (current index|start index), and $C = (N - T + 1)T$. If the truncation length is set to $T = N$, then the prediction loss (6) is recovered.

Formulation (8) addresses both shortcomings of the prediction loss mentioned in Section 1. Based on the fact that the predictions can be computed in parallel, only T computations are required to be performed in series hence providing $O(T)$ computational scaling, which can be considerably smaller than the initial $O(N)$. Moreover, as is shown in Section 4.2, the use of truncated sections increases the loss function smoothness (Ribeiro et al., 2020), which both makes gradient-based optimization methods more stable and reduces the effect of parameter initialization on the optimization, making the estimation process more reproducible and less varied (Ribeiro et al., 2020).

The computational cost of the proposed loss function (8) can be further decreased by not summing over all available subsections of the complete data set D_N for each optimization step,

but only over a subset of subsections. This results in a batch formulation of the loss:

$$V_{D_N}^{(\text{sub;batch})}(\ ) = \frac{1}{N_{\text{batch}}} \sum_{t \in I} v_t; \tag{9a}$$

$$v_t \ , \ \frac{1}{T} \sum_{k=0}^{T-1} \|y_{t+k} - \hat{y}_{t+k|t}\|_2^2; \tag{9b}$$

$$I \subset \mathbb{I}_{n+1}^{N-T+1} = \{n+1; n+2; \dots; N-T+1\} \tag{9c}$$

$$\text{s.t. } |I| = N_{\text{batch}};$$

which allows for the utilization of powerful batch optimization algorithms such as the Adam optimizer (Kingma & Ba, 2015). Moreover, it is also not necessary to have the complete data set in memory, see Beintema et al. (2021a), which is a significant advantage in case of large data sets.

An important problem in the minimization of (8) is that there is no expression for the initial state $\hat{x}_{t|t}$ of each subsection. Considering the initial state of each section to be an optimization parameter in the minimization of (8) trades new optimization parameters for the scalability of the cost function (i.e. number of parameters would scale $O(N)$). This quickly outweighs the benefits of (8). Hence, to preserve the advantages of the cost function reformulation, an appropriate estimator of the initial state $\hat{x}_{t|t}$ is required. The next section introduces an encoder-based state estimator based on the concept of state reconstructability.

3.2. Subspace encoder

To introduce the proposed encoder, we first require some preliminary notions from nonlinear system theory. Due to causality of (1), it is a fundamental property of the state that x_k with $k > 1$ is completely determined by the past sequence of inputs $\{u_l\}_{l=1}^{k-1}$ and disturbances $\{e_l\}_{l=1}^{k-1}$ together with an initial state x_1. In case of state observability of (1), this initial state x_1 can be determined based on a future IO sequence (Isidori, 1985). The complementary notion of state reconstructability considers the determination of x_k based on a purely past IO sequence (Isidori, 1985). The concepts of state observability and reconstructability and the realization theory that builds upon them both for deterministic and stochastic systems form the cornerstones of subspace identification of linear systems and led to many powerful estimation algorithms, see Katayama et al. (2005) for an overview.

To exploit the concept of observability and reconstructability in deep learning-based identification of (1), consider the result of our derivations in (4). If for an $n \geq 1$, $\ _n$ is partially invertible w.r.t x_k on the open sets $\mathsf{X}_0 \subseteq \mathbb{R}^{n_x}$, $\mathsf{U}_0 \subseteq \mathbb{R}^{n_u}$, $\mathsf{Y}_0 \subseteq \mathbb{R}^{n_y}$, $\mathsf{E}_0 \subseteq \mathbb{R}^{n_y}$, i.e., there exists a $\ _n : \mathsf{U}_0^n \times \mathsf{Y}_0^{n+1} \times \mathsf{E}_0^{n+1} \to \mathbb{R}^{n_x}$ such that $x_k = \ _n(u_k^{k+n-1}; y_k^{k+n}; e_k^{k+n})$ with $x_k \in \mathsf{X}_0$ and IO signals in these sets, then (1) is called *locally observable* on $(\mathsf{X}_0; \mathsf{U}_0; \mathsf{Y}_0; \mathsf{E}_0)$ and $\ _n$ is called the *observability map* of (1) (Isidori, Sontag, & Thoma, 1995). Note that if there exists a $(x_*; w_*) \in \mathbb{R}^{n_x} \times \mathbb{R}^{nn_u \times (n+1)n_y \times (n+1)n_y}$ that $\nabla_{x_*} \ _{n_x-1}(x_*; w_*)$ is full row rank, then there are open sets $x_* \in \mathsf{X}_0$ and $w_* \in \mathsf{U}_0^n \times \mathsf{Y}_0^{n+1} \times \mathsf{E}_0^{n+1}$ such that the partial inverse of $\ _n$ exists in terms of an analytic function $\ _n$ (Isidori et al., 1995). Furthermore, if (1) is locally observable on $(\mathsf{X}_0; \mathsf{U}_0; \mathsf{Y}_0; \mathsf{E}_0)$, then $\ _n$ with $n \geq n_x - 1$ has to be partially invertible in the above defined sense.

Let $\circ_n \tilde{f}$ be a shorthand for $\tilde{f}$ when $n = 1$ and $\tilde{f} \circ_{n-1} \tilde{f}$ for $n > 1$. Consider

$$x_k = (\circ_n \tilde{f})(x_{k-n}; u_{k-n}^{k-1}; y_{k-n}^{k-1}) \tag{10a}$$

$$= (\circ_n \tilde{f}) \ _n(u_{k-n}^{k-1}; y_{k-n}^{k}; e_{k-n}^{k}); u_{k-n}^{k-1}; y_{k-n}^{k-1} \tag{10b}$$

$$= \ _n(u_{k-n}^{k-1}; y_{k-n}^{k}; e_{k-n}^{k}) \tag{10c}$$

which is called the *reconstructability map* (Isidori et al., 1995) of (1) as it allows to recover x_k from past measured IO data. Note that the noise sequence e_{k-n}^k is not directly available in practice to compute this recovery based on (10c), but again we can exploit the i.i.d. white noise property of e_k to arrive at:

$$\bar{x}_k = \mathbb{E}_e[x_k \mid u_{k-n}^{k-1}; y_{k-n}^{k}] = \ _n(u_{k-n}^{k-1}; y_{k-n}^{k}); \tag{11}$$

giving an efficient estimator of x_k. In the sequel, we will exploit this concept to formulate an encoder that approximates $\ _n$.

As shown in (11), there exists a state-estimator in the conditional expectation sense for the original system and also the same estimator can be derived for the model structure (7). However, the exact calculation of this estimator for a given ANN parametrization of f and h is practically infeasible due to the required analytic inversion in terms of $\ _n$ and the computation of the conditional expectation of $\ _n$ under a given $\ _e$. Hence, we aim to approximate $\ _n$ by introducing a nonlinear function $\ $ which is co-estimated with f and h. Since $\ $ aims to approximate the subspace reconstructability map (10c) we call it the subspace encoder. Similarly to f and h it is also assumed to be Lipschitz continuous:

$$\hat{x}_{t|t} \ , \ \ (u_{t-n}^{t-1}; y_{t-n}^{t}); \tag{12}$$

Here, n corresponds to the number of past inputs and outputs, i.e. *lag window*, considered to estimate the initial state, while $\in \ \subseteq \mathbb{R}^n$ is the collection of the parameters associated with $\ $ in terms of a corresponding ANN with multiple hidden layers. In order to provide an estimator for the initial state of the considered model structure (7), the encoder function $\ $ is co-estimated with f and h by adding the parameters $\ $ and the estimated initial state using $\ $ to the loss function (8).

$$V_{D_N}^{\text{enc}}(\ ; \ ) = \frac{1}{C} \sum_{t=n+1}^{N-T+1} \sum_{k=0}^{T-1} \|y_{t+k} - \hat{y}_{t+k|t}\|_2^2; \tag{13a}$$

$$\hat{x}_{t|t} = \ (u_{t-n}^{t-1}; y_{t-n}^{t}); \tag{13b}$$

$$\hat{x}_{t+k+1|t} = f\ (\hat{x}_{t+k|t}; u_{t+k}; \hat{e}_{t+k|t}); \tag{13c}$$

$$\hat{y}_{t+k|t} = h\ (\hat{x}_{t+k|t}); \tag{13d}$$

$$\hat{e}_{t+k|t} = y_{t+k} - \hat{y}_{t+k|t}; \tag{13e}$$

where now $C = (N - T - n + 1)T$ and which again can be formulated as a batch loss function similar to (9). The used truncated prediction loss and the introduced subspace encoder lead to a model with a deep network structure for estimation, which we call the *subspace-encoder network* (SUBNET). It is graphically summarized in Fig. 1.

The derivation of the reconstructability map has shown that based on $n = n_x - 1$ past inputs and outputs, an effective unbiased estimator of the initial state $x_{t|t}$ can be achieved. While $n = n_x - 1$ is often the minimal required number of past IO samples to obtain an unbiased estimator, the variance of estimate $\hat{x}_{t|t}$ can be rather significant and further reduced by increasing n. The underlying mechanism is similar to the concept of minimum variance observers (Darouach & Zasadzinski, 1997) that provide statically efficient state estimation using $n > n_x - 1$ input and outputs lags. Besides of showing empirically this effect in Section 6, a deeper theoretical exploration of the variance optimal choice of n is not within the scope of this paper.

3.3. Parameter estimation

To obtain a model estimate in terms of the SUBNET structure through the minimization of the loss (13), the following steps are executed: *(i)* random initialization of all networks in Fig. 1 by an efficient approach such as the Xavier method (Glorot & Bengio,

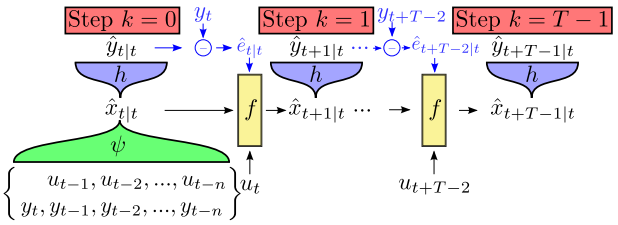


Fig. 1. Overall SUBNET structure: the subspace encoder estimates the initial state at time index t based on past inputs and outputs, then the state is propagated through f and h multiple times until the truncation length T. The parts marked in blue constitute the innovation noise process. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2010), *(ii)* for the given normalized data (or batches of data) the loss is computed while the computation graph with intermediate values are saved in memory (this uses $O(N_{\text{batch}}T)$ memory), *(iii)* the gradient of the loss is computed by back-propagation using the computation graph obtained in Step *(ii)*, *(iv)* the network parameters are updated by a stochastic gradient optimization method like Adam (Kingma & Ba, 2015), *(v)* iteration is continued till convergence or cross-validation based early stopping.

3.4. User guidelines

The subspace encoder has a number of hyper-parameters that need to be chosen based on the to-be-identified system at hand. Hence, a few guidelines are provided based on insights obtained from theoretical analysis, numerical analysis and practical experience.

- Choose T to be a few times the largest characteristic time scale for stable data-generating systems. For such a choice, the truncated prediction loss (13) provides a close approximation of the 'regular' prediction error at a low computational cost.
- n_x and n need to be chosen as at least the effective order and lag (minimal reconstruction order) of the system, respectively. Furthermore, one can choose different lags n_a for past y and n_b for past u and increase n to reduce the variance of the initial state estimate.
- The choice of the ANN architectures (number of layers q, activation functions per layer m, type of activation functions) used to parametrize f, g and are system dependent. However, overfitting on the data caused by the choice of an over-parametrized architecture is suppressed by the used innovation noise model structure, regularization induced by the overlapping subsections, early stopping and batch optimization. Hence, a suggested baseline is to use 2 hidden layer networks with 64 nodes per layer, tanh activation and a linear bypass (similar to a residual component).
- IO normalization is essential to make the signals involved in the estimation to be zero-mean and have standard deviation of one. This is required as IO normalization is a key assumption in many parameter initialization methods (e.g. Xavier initialization Glorot & Bengio, 2010) and the "active" range of many activation functions is also close to a range around zero with a width of 1. After estimation, to remove normalization, a back scaling is added to the resulting model estimate.
- The batch size should be the smallest size which only marginally compromises the data throughput speed (i.e. training samples processed per second) and further reduced to increase regularization effects of batch-optimization. This guideline is according to the current consensus in the ML community. The baseline is 256, but this is data and architecture dependent.
- For all our experiments a fixed learning rate of 10^{-3} using the Adam optimizer has been sufficient. The model quality can be further improved by using early stopping and returning the model of the epoch which had the lowest validation error.

3.5. Comparison to the state-of-the-art

Contrary to other approaches that use an encoder function such as Masti and Bemporad (2021), which is based on a modified auto-encoder to learn the latent state and a 1-step ahead prediction loss to learn the system dynamics, we do not need to introduce any additional loss function elements to fit the encoder function. Intuitively, a more accurate estimate of the state automatically reduces the transient error and hence the mismatch between the measurements and the predicted model output. Thus, reducing the transient also reduces the truncated prediction loss, which makes it superfluous to introduce any additional cost function terms.

The proposed estimation method can be also related to multiple shooting methods (Bock, 1981). Multiple shooting also subdivides the time series into multiple sections and adds the initial state of each section to the parameter vector together with additional constraints (Decuyper et al., 2020). Compared to this method, our proposed method uses the subspace encoder to directly estimate the initial state from past inputs and outputs for each section. As a consequence, the computational complexity does not increase for an increasing number of sections. Furthermore, our formulation uses overlapping subsections whereas multiple shooting does not make use of overlap. In Section 4.3, we prove that section overlap increases data efficiency.

Truncated back-propagation through time (truncated BPTT) (Tallec & Ollivier, 2017) also sub-divides the time series, but by truncating the gradient calculation at a truncation length to stabilize the gradient. This still requires a full pass over the time series data which can be computationally expensive and still unstable (value explosion) for large data sets. Furthermore, it adds extra bias and/or variability to the gradient estimate, which is not the case with the proposed subspace encoder method.

Lastly, the subspace encoder function not only qualifies as a reconstructability map, but also as a state observer. Hence, the encoder can be used to kick-start simulations on possibly unseen data sets. In particular, *nonlinear model predictive control* (NMPC) relies on accurate few-step-ahead prediction models and state estimates (Allgöwer & Zheng, 2012), which makes the combined SUBNET structure with the encoder based observer readily applicable for MPC.

4. Theoretical analysis

In this section, we show key theoretical properties of the proposed encoder method in terms of consistency corresponding to statistical validity of the estimator, loss function smoothness that implies optimization efficiency, and data efficiency resulting from allowed overlaps in the subsections.

4.1. Consistency of the estimator

The notion of consistency, as defined in Ljung (1978), expresses that the resulting model estimates tend to an equivalent representation of the system that generated the data when the number of data points tends to infinity. In other words, the model estimate is asymptotically unbiased and converges asymptotically

to a true model of the system. Thus in this section, we will show consistency of the SUBNET approach relying on the results of Ljung (1978).

Data-generating system: To show consistency, we need to introduce some conditions on the data-generating system. As we discussed, the true system (1) can be reformulated in a 1-step-ahead predictor form given by (5). For $k \geq 1$, let $\mathcal{W}_{[1;k]}$ denote the -algebra generated by the random variables $(u_1^k; e_1^k)$ and let $_{\mathrm{w}} : \mathcal{W}_{[1;k]} \to [0; 1]$ denote the associated probability measure. Furthermore, as f and h are deterministic, define

$$\mathfrak{B} = \{(y_1^\infty; x_1^\infty; u_1^\infty; e_1^\infty) \in (\mathsf{R}^{n_\mathrm{w}})^{\mathsf{N}} \mid (u_1^\infty; e_1^\infty) \in \mathcal{W}_{[1;\infty]}; \text{ and } (y_k; x_k; u_k; e_k) \text{ satisfies (1) } \forall k \in \mathsf{N}\}; \tag{14}$$

with $n_\mathrm{w} = n_\mathrm{y} + n_\mathrm{x} + n_\mathrm{u} + n_\mathrm{y}$, being the sample path behavior, i.e., the set of all solution trajectories, of (1). Note that by defining the -algebra $\mathcal{B}$ over $\mathfrak{B}$ and an appropriate probability measure $_\mathrm{b}$, the stochastic behavior of (1) can be fully represented, see Willems (2013).

Let $\mathfrak{B}_{[k_\mathrm{o};k]}$ and $\mathcal{B}_{[k_\mathrm{o};k]}$ be the restriction of $\mathfrak{B}$ and $\mathcal{B}$ to the time interval $[k_\mathrm{o}; k] \subseteq \mathsf{N}$ with $k \geq k_\mathrm{o}$, respectively. Then, for a given sample path $\{(y_k; x_k; u_k; e_k)\}_{k=k_\mathrm{o}}^{\infty} \in \mathfrak{B}_{[k_\mathrm{o};\infty]}$ of (1) with $x(k_\mathrm{o}) = x_\mathrm{o}$, $\{(\tilde{y}_k; \tilde{x}_k; u_k; e_k)\}_{k=k_\mathrm{o}}^{\infty} \in \mathfrak{B}_{[k_\mathrm{o};\infty]}$ corresponds to the response of (1) for the perturbed state value $\tilde{x}(k_\mathrm{o}) = \tilde{x}_\mathrm{o}$ at time moment $k_\mathrm{o} \in \mathsf{N}$ subject to the same input and disturbance as the nominal state response. Based on these, the following stability condition is formulated:

Condition 1 (*Incremental Exp. Output Stability*)**.** *The data-generating system (4) is (globally) incrementally exponentially output stable, meaning that for any > 0, there exist a $0 \leq C( ) < \infty$ and a $0 \leq < 1$ such that*

$$\mathsf{E}_e[\|y_k - \tilde{y}_k\|_2^4] < C( )^{k-k_\mathrm{o}}; \quad \forall k \geq k_\mathrm{o} \tag{15}$$

under any $k_\mathrm{o} \geq 1$, $x_\mathrm{o}; \tilde{x}_\mathrm{o} \in \mathsf{R}^{n_\mathrm{x}}$ with $\|x_\mathrm{o} - \tilde{x}_\mathrm{o}\|_2 <$ and $(u_1^\infty; e_1^\infty) \in \mathcal{W}_{[1;\infty]}$, where the random variables y_k and $\tilde{y}_k$ belong to $\mathcal{B}_{[k_\mathrm{o};\infty]}$ with the same $(u_k; e_k)$, but with $x_{k_\mathrm{o}} = x_\mathrm{o}$ and $\tilde{x}_{k_\mathrm{o}} = \tilde{x}_\mathrm{o}$.

Model Set: The considered SUBNET model (13b)–(13e) corresponds to a model structure M parametrized by a finite-dimensional parameter vector $= [\ ^\top \ \ ^\top]^\top$ that is restricted to vary in a compact set $\subset \mathsf{R}^n$. The resulting model set is $M = \{M \mid \in \}$. For each $\in$, the SUBNET model M with a given encoder lag $n \geq 1$, can be written in a 1-step-ahead predictor form

$$\hat{y}_{t+k|t} = \hat{}_k(\ ; y_{t-n}^{t+k-1}; u_{t-n}^{t+k-1}). \tag{16}$$

For M, two important conditions are considered.

Condition 2 (*Differentiability*)**.** *The 1-step-ahead predictor $\hat{}_k : \mathsf{R}^{n\ +n} \times \mathsf{R}^{(n_\mathrm{y}+n_\mathrm{u})(n+k)} \to \mathsf{R}^{n_\mathrm{y}}$ is differentiable with respect to for all $\in \breve{}$, where $\breve{}$ is an open neighborhood of .*

Next, we require the influence of delayed inputs and outputs on the predictor map $\hat{}_k$ to be exponentially decaying with a number of delays to assure the convergence of the predictor. This is formalized as follows;

Condition 3 (*Predictor Convergence*)**.** *There exist a $0 \breve{} \leq C < \infty$ and a $0 \leq < 1$ such that, for any $k \geq 0$ and $\in \breve{}$, where is an open neighborhood of , the deterministic predictor map $\hat{}_k$ under Condition 2 satisfies*

$$\|\hat{}_k(\ ; u_{-n}^{k-1}; y_{-n}^{k-1}) - \hat{}_k(\ ; \tilde{u}_{-n}^{k-1}; \tilde{y}_{-n}^{k-1})\|_2 \leq C \sum_{s=-n}^{k-s} \|u_s - \tilde{u}_s\|_2 + \|y_s - \tilde{y}_s\|_2 \ ; \tag{17}$$

for any $(u_{-n}^{k-1}; y_{-n}^{k-1}); (\tilde{u}_{-n}^{k-1}; \tilde{y}_{-n}^{k-1}) \in \mathsf{R}^{(n_\mathrm{y}+n_\mathrm{u})(n+k)}$ and

$$\|\hat{}_k(\ ; 0_{-n}^{k-1}; 0_{-n}^{k-1})\|_2 \leq C; \tag{18}$$

where $0_{t-n}^{t+k-1} = [\ 0 \ \cdots \ 0\ ]^\top$. Furthermore, (17) is also satisfied by $\frac{@}{@}\hat{}_k(\ ; y_{-n}^{k-1}; u_{-n}^{+k-1})$.

Convergence: Under the previous considerations, convergence of the SUBNET estimator can be shown, which is a required property to show consistency.

Theorem 4 (*Convergence*)**.** *Consider system (1) satisfying Condition 1 with a quasi-stationary u independent of the white noise process e. Let the set of models M defined by the model structure (13b)–(13e) for $\forall \in$ satisfy Conditions 2 and 3. Then*

$$\sup_{(\ ; \ )\in} V_{D_N}^{\mathrm{enc}}(\ ; \ ) - \mathsf{E}_e[V_{D_N}^{\mathrm{enc}}(\ ; \ )]_{\ 2} \to 0; \tag{19}$$

with probability 1 as $T; N \to \infty$ and the sequence of functions $\mathsf{E}_e[V_{D_N}^{\mathrm{enc}}(\ ; \ )]$ is equicontinuous in $\in$.

Proof. The mean squared prediction error identification criterion used in (13) satisfies Condition C1 in Ljung (1978), hence the proof of Ljung (1978, Lemma 3.1) applies for the considered case.

Consistency: In order to show consistency, we need to assume that the system is part of the model set. Consider the state-reconstructability map $_n$ in (10c) for the data-generating system (1) with $n \geq n_\mathrm{x}$. Note that

$$\begin{aligned} y_{t+k} &= {}_k(\ {}_n(u_{t-n}^{t-1}; y_{t-n}^{t}; e_{t-n}^{t}); u_t^{t+k-1}; y_t^{t+k-1}) + e_{t+k}; \\ &= \breve{}_k(u_{t-n}^{t+k-1}; y_{t-n}^{t+k-1}; e_{t-n}^{t}) + e_{t+n} \end{aligned} \tag{20}$$

for any $t; k \geq 0$, where $_k$ is according to (5), i.e., $_k = (h \circ_k \tilde{f})$. Then,

$$\begin{aligned} \bar{y}_{t+k|t} &= \mathsf{E}_e[\breve{}_k(u_{t-n}^{t+k-1}; y_{t-n}^{t+k-1}; e_{t-n}^{t})] \\ &= \bar{}_k(u_{t-n}^{t+k-1}; y_{t-n}^{t+k-1}); \end{aligned} \tag{21}$$

is the optimal one-step-ahead predictor associated with (1) under an n-lag based reconstructability map.

Definition 5 (*Equivalence Set*)**.** For a given model structure M with encoder lag $n \geq n_\mathrm{x}$, predictor $\hat{}_k(\ ; \ )$ (see (16)) and $\in \subset \mathsf{R}^n$, the set of equivalent models with the data-generating system (1) in the one-step-ahead prediction sense (21) is defined as

$$_* = \{ \in \mid \hat{}_k(\ ; \ ) = \bar{}_k( ); \ \forall k \geq 0 \ : \} \tag{22}$$

Note that if $_* \neq ?$, then there exists a $M_{\ *} \in M$ that is equivalent with (1). In this case, we call the considered model set to be sufficiently rich to contain an equivalent realization of the data-generating system. Next we need to ensure that under the given observed data from (1), we can distinguish non-equivalent models in M.

Condition 6 (*Persistence of Excitation*)**.** *Given the model set $M = \{M \mid \in \}$ with $n \in \mathsf{N}$ and the associated $V_{D_N}^{\mathrm{enc}}$ in terms of (13) with $0 \leq T \leq N$, we call the input sequence u_1^N in D_N generated by (1) to be weakly persistently exciting, if for all pairs of parameterizations given by $. \ _1; \ _1 / \in$ and $. \ _2; \ _2 / \in$ for which the function mapping is unequal, i.e., $V_{(\cdot)}^{\mathrm{enc}}(\ _1; \ _1) \neq V_{(\cdot)}^{\mathrm{enc}}(\ _2; \ _2)$, we have*

$$V_{D_N}^{\mathrm{enc}}(\ _1; \ _1) \neq V_{D_N}^{\mathrm{enc}}(\ _2; \ _2): \tag{23}$$

with probability 1.

To show consistency, we also require that any element of $_*$ has minimal cost.

Property 7 (*Minimal Cost*). *If M is sufficiently rich, then for any $= [\ ^\top\ \ ^\top]^\top \in$ and $_* = [\ ^\top_*\ \ ^\top_*]^\top \in\ _*$ the encoder loss in (13) has the following property:*

$$\lim_{T;N\to\infty} V^{\text{enc}}_{D_N}(\ _*;\ _*) \le \lim_{T;N\to\infty} V^{\text{enc}}_{D_N}(\ ;\ ) \tag{24}$$

with probability 1.

Proof. Since $\mathsf{E}_e[V^{\text{enc}}_{D_N}(\ _*;\ _*)]$ exists as shown in Theorem 4, it is sufficient to show that

$$\lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T-1} \|\hat{y}^*_{t+k|t} - y_{t+k}\|_2^2 \le \lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T-1} \|\hat{y}_{t+k|t} - y_{t+k}\|_2^2 \tag{25}$$

for all t where $\hat{y}^*_{t+k|t} = \hat{}_k(\ _*; y^{t+k-1}_{t-n}; u^{t+k-1}_{t-n})$. By the law of large numbers, as $T \to \infty$, the sample distribution of $\{e_{t+k}\}_{k=0}^{T-1}$ will converge to the original white noise distribution of e with finite variance $_e$ and with probably 1. Thus, it is sufficient to show that

$$\mathsf{E}_e[\|\hat{y}^*_{t+k|t} - y_{t+k}\|_2^2] \le \mathsf{E}_e[\|\hat{y}_{t+k|t} - y_{t+k}\|_2^2] \tag{26}$$

which can be expanded with $y_{t+k} = h(x_{t+k}) + e_{t+k}$ as

$$\begin{aligned}\mathsf{E}_e[\|\hat{y}^*_{t+k|t} - y_{t+k}\|_2^2] &= \mathsf{E}_e[\|\hat{y}^*_{t+k|t} - h(x_{t+k})\|_2^2] - \\ &\mathsf{E}_e[2(\hat{y}^*_{t+k|t} - h(x_{t+k}))\cdot e_{t+k}] + \mathsf{E}_e[\|e_{t+k}\|_2^2].\end{aligned} \tag{27}$$

The second term of this expansion is equal to zero since e_{t+k} is uncorrelated to $(\hat{y}^*_{t+k|t} - h(x_{t+k}))$ and e_{t+k} is zero-mean. Furthermore, the first term is also zero since in terms of Definition 5, $\hat{y}^*_{t+k|t} = \hat{}_k(\ _*; y^{t+k-1}_{t-n}; u^{t+k-1}_{t-n})$ is equal to $\bar{y}_{t+k|t}$ in (21). Hence,

$$\mathsf{E}_e[\|\hat{y}^*_{t+k|t} - y_{t+k}\|_2^2] = \|\ _e\|_2^2 \tag{28}$$

which is irreducible and thus minimal.

Theorem 8 (*Consistency*). *Under the conditions of Theorem 4, Condition 6 and Property 7,*

$$\lim_{T;N\to\infty} \hat{}_N \in\ _* \tag{29}$$

with probability 1, where

$$\hat{}_N = \arg\min_{\in} V^{\text{enc}}_{D_N}(\ ;\ ). \tag{30}$$

Proof. See Lemma 4.1 in Ljung (1978). Note that the squared loss function (13) fulfills Condition (4.4) in Ljung (1978).

4.2. Increased cost smoothness due to truncation

Next, we show that the considered estimation structure and the truncated prediction loss increase the smoothness of the cost function, which potentially makes the optimization process for model estimation more stable and less prone to get stuck in local minima (Ribeiro et al., 2020). For this purpose, we investigate the smoothness of the encoder loss function by the means of the Lipschitz-continuity analysis. The Lipschitz constant $L_{\text{enc};T} \ge 0$ for the considered loss function is defined as

$$\begin{aligned}&\|V^{\text{enc};T}_{D_N}(\ _1;\ _1) - V^{\text{enc};T}_{D_N}(\ _2;\ _2)\|_2^2 \\ &\quad\le L^2_{\text{enc};T}(\|\ _1 - \ _2\|_2^2 + \|\ _1 - \ _2\|_2^2)\end{aligned} \tag{31}$$

with $[\ ^\top\ \ ^\top]^\top \in\ =\ \times$, assuming that is not only compact, but it is also convex. Here, the T dependence of the loss function is added explicitly. Since $L_{\text{enc};T}$ bounds the slope of the function, it provides insight into the smoothness of the cost function as T changes. By the following theorem, we show that smoothness of $V^{\text{enc};T}_{D_N}$ can decrease exponentially with increasing T.

Theorem 9. *Assume that f, h and are Lipschitz continuous with Lipschitz constants L_f, L_h and L. Then, $L_{\text{enc};T}$ and $L'_{\text{enc};T}$, representing the Lipschitz constant of the derivative of $V^{\text{enc};T}_{D_N}$, scale as*

$$L_{\text{enc};T} = O(L^{2T}_{\tilde{f}});\quad L'_{\text{enc};T} = O(L^{3T}_{\tilde{f}}); \tag{32}$$

if $L_{\tilde{f}} = L_f\sqrt{1 + L_h^2} > 1$.

Proof. For Lipschitz-continuous functions $q(x)$ and $p(x)$, two known properties of the Lipschitz constant are: *(i)* the *sum* of two functions $q(x)+p(x)$ has the Lipschitz constant $L_q + L_p$ and *(ii)* the *multiplication* of two functions $q(x)p(x)$ has the Lipschitz constant $L_q m_p + L_p m_q$, where m_q is the maximum of $q(x)$ on the considered compact set $x \in\ _x$ and m_q is similarly defined.

The Lipschitz constants of f, h and are defined by the following relations:

$$\|h\ _1(x) - h\ _2(\tilde{x})\|_2^2 \le L_h^2(\|\ _1 - \ _2\|_2^2 + \|x - \tilde{x}\|_2^2); \tag{33a}$$

$$\begin{aligned}&\|f\ _1(x; u; y - h\ _1(x)) - f\ _2(\tilde{x}; u; y - h\ _2(\tilde{x}))\|_2^2 \\ &\quad\le L_f^2(\|\ _1 - \ _2\|_2^2 + \|x - \tilde{x}\|_2^2 + \|h\ _1(x) - h\ _2(\tilde{x})\|_2^2) \\ &\quad\le L_f^2(1 + L_h^2)(\|\ _1 - \ _2\|_2^2 + \|x - \tilde{x}\|_2^2);\end{aligned} \tag{33b}$$

and

$$\begin{aligned}&\|\ \ _1(u^{t-1}_{t-n}; y^t_{t-n}) - \ \ _2(u^{t-1}_{t-n}; y^t_{t-n})\|_2^2 \\ &\quad\le L^2\|\ _1 - \ _2\|_2^2.\end{aligned} \tag{33c}$$

Since $V^{\text{enc};T}(\ ;\ ) = 1/N_{\text{sec}} \sum_t\ _t$ with $N_{\text{sec}} = N - T - n + 1$ as defined in (13), by the sum property, we have that $L_{\text{enc};T} = L_{v_t}$. Using the relations (33a)–(33c), it is possible to derive L_{v_t} in terms of $L_h; L_f; L$ and T. A similar derivation has been done by Ribeiro et al. (2020) of the cost function $V_T(\ ) = \frac{1}{T}\sum_{t=1}^{T}\|y_t - \hat{y}_t\|_2^2$ where an OE noise model was considered and, instead of an encoder, different initial states x_0 were used. They showed that the following scaling law applies

$$L_{V_T} = O(L^{2T}_f);\quad L'_{V_T} = O(L^{3T}_f); \tag{34}$$

when $L_f > 1$ and where L'_{V_T} represents the Lipschitz constant of the derivative of V_T. Hence, to derive the scaling of L_{v_t} and thus $L_{\text{enc};T}$ we rely on this derivation by only showing that these differences leave the exponential scaling with T unaltered.

Adapting this result to our considered case is relatively simple to show since the encoder only changes the initial state difference $\|x_0 - \tilde{x}_0\|_2^2$ to $L^2\|\ _1 - \ _2\|_2^2$ which is independent of T and the change to innovation structure replaces L_f by $L_{\tilde{f}} = L_f\sqrt{1 + L_h^2}$.

4.3. Data-efficiency with overlapping subsections

To quantify the data-efficiency of overlapping subsections, consider a fixed T for the T-step truncated prediction loss and analyze the data efficiency using equidistantly placed sections in terms of the distance parameter d, i.e. $I = \{1 + dk \in \mathbb{I}_1^{N-T-1} \mid k \in \mathbb{N} \cup 0\}$. The parameter d regulates the distance between each sub-section where $d = 1$ recovers the encoder formulation and

$d = T$ recovers the conventional approach in multiple shooting with no overlap. To make the notation more compact, introduce a change of variables in the sum (9b) such that

$$V_{D_N}^d(\ ;\ ) = \frac{1}{m_d} \sum_{k=0}^{m_d-1} v_{1+dk} \tag{35}$$

where $N - T + 1 = dm_d + r_d$ with $m_d; r_d \in \mathbb{N}$ and $0 \leq r_d < d$.

To define data-efficiency, we assume stationary input and output signals

Assumption 10 (*Stationarity*)**.** Both the model output $\hat{y}_{t+k|t}$ and system output y_t are assumed to be strictly statistically stationary. In other words, the cumulative distribution function p_Y of the joint distribution of instances of y_t at times $t_1; \ldots; t_n$ has the property that

$$p_Y(y_{t_1+\ }; \ldots; y_{t_n+\ }) = p_Y(y_{t_1}; \ldots; y_{t_n}): \tag{36}$$

for all $t_1; \ldots; t_n;\ \in \mathbb{Z}$.

This assumption is reasonable since many fading memory system like bi-linear systems (Priestley, 1988), and Volterra series (Boyd & Chua, 1985) have the property that the system output y_t is quasi stationary if the input u_t is stationary. However, to our knowledge a proof of this property for stable systems defined via an NL-SS representation is not present in the literature. Firstly, to show enhanced data efficiency we need that the cost functions converge to the same cost function in the limit of infinite data.

Theorem 11 (*Asymp. insensitivity For T*)**.** *Under Assumption 10, both $V_{D_N}^1(\ ;\ )$ and $V_{D_N}^T(\ ;\ )$ converge to the same loss function with probability 1 when $N \to \infty$.*

Proof. Assumption 10 implies that v_t is also strictly stationary since any signal which is dependent only on strictly stationary variables is also strictly stationary. Furthermore, by the law of large numbers, the infinite mean sum of v_t becomes equal to $\mathbb{E}[v_k]$ with probability 1. Hence, the limit cost can be expressed as

$$\lim_{N\to\infty} V_{D_N}^d(\ ;\ ) = \lim_{N\to\infty} \frac{1}{m_d} \sum_{k=0}^{m_d-1} v_{1+dk} = \mathbb{E}[v_k]$$

which is independent of d.

Next, we show that allowing for overlap, e.g. by taking $d = 1$, reduces the variance of the loss function compared to disallowing overlap by $d = T$.

Theorem 12 (*Overlap Effect*)**.** *With Assumption 10, there exists an $N_* \in \mathbb{N}$, such that, for all $N > N_*$, the following relation holds*

$$\mathrm{Var}(V_{D_N}^1(\ _*;\ _*)) \leq \mathrm{Var}(V_{D_N}^T(\ _*;\ _*))$$

for all $(\ _*;\ _*) \in\ _*$ given by Eq. (22).

Proof. Proving this statement is equivalent to showing that the function

$$G(d)\ ,\ \mathrm{Var}\ V_{D_N}^d(\ _*;\ _*)\ = \mathrm{Var}\ \left(\frac{1}{m_d} \sum_{k=0}^{m_d-1} v_{1+dk}\right) \tag{37}$$

has the property of $G(1) \leq G(T)$ for all $N > N_*$. By expanding (37) using conventional variance and covariance relations, we get

$$G(d) = \frac{1}{m_d^2} \sum_{k=0}^{m_d-1} \sum_{l=0}^{m_d-1} \mathrm{Cov}(v_{1+kd}; v_{1+ld}): \tag{38}$$

As we have shown in Theorem 11, v_t is strictly stationary under Assumption 10, hence we can replace the covariance by $C(d|k-l|)\ ,\ \mathrm{Cov}(v_{kd}; v_{ld})$ and use the auto-correlation function $R(t)\ ,\ C(t)=C(0)$ to simplify the expression to

$$G(d) = \frac{1}{m_d^2}\left(m_d + 2\sum_{t=1}^{m_d-1} (m_d - t)R(td)\right) : \tag{39}$$

The only part which remains to complete the proof is to determine if $R(td)$ under the given assumptions implies $G(1) \leq G(T)$ using this expression.

The value of $R(td)$ can be derived from

$$v_t = \frac{1}{T} \sum_{k=0}^{T-1} \|y_{t+k} - \hat{y}_{t+k|t}\|_2^2 = \frac{1}{T} \sum_{k=0}^{T-1} \|e_{t+k}\|_2^2 \tag{40}$$

since we evaluate the cost in $(\ _*,\ _*)$. Hence,

$$\mathrm{Cov}(v_t; v_{t+\ }) \sim \sum_{i=0}^{T-1} \sum_{j=0}^{T-1} \mathrm{Cov}(\|e_{t+i}\|_2^2; \|e_{t+\ +j}\|_2^2) \tag{41}$$

where $\mathrm{Cov}(\|e_{t+i}\|_2^2; \|e_{t+\ +j}\|_2^2)$ is nonzero if and only if $i =\ + j$ and it is the same value for any t since e_t is white. Hence, the auto-correlation function is $R(t) = \max(0; 1 - t/T)$, which simply counts the number of terms which have the same index in the sum. After substitution of $R(t) = \max(0; 1 - t/T)$ in (39), it directly follows that $G(1) \leq G(T)$.

Under consistency of the estimator, the parameter estimates will converge to $(\ _*,\ _*)$, hence, this result hold in general. This shows that allowing for overlap in the subsections results in a more efficient estimator.

5. Simulation study

In this section, we demonstrate the effectiveness of the proposed SUBNET architecture based identification approach in an extensive simulation study. As the method has a number of hyperparameters that can substantially alter its behavior and performance, hence we investigate the effects of these hyperparameters and also motivate the previously provided guidelines for choosing them. An evaluation of the subspace encoder method on experimental data is provided in Section 6.

5.1. Data-generating system

The following system is considered:

$$x_{k+1}^{(1)} = \frac{x_k^{(1)}}{1{:}2 + \left(x_k^{(2)}\right)^2} + 0{:}4 \cdot x_k^{(2)}; \tag{42a}$$

$$x_{k+1}^{(2)} = \frac{x_k^{(2)}}{1{:}2 + \left(x_k^{(1)}\right)^2} + 0{:}4 \cdot x_k^{(1)} + u_k; \tag{42b}$$

$$y_k = x_k^{(1)} + e_k; \tag{42c}$$

where $x_k = [x_k^{(1)}\ \ x_k^{(2)}]^\top$ denotes the elements of the state vector, $x_0 = 0$ and e is generated by an i.i.d. white Gaussian noise process resulting in an output SNR of 20 dB. This noise signal is only present in the training and validation data sets and it is omitted in the test set to accurately measure the performance of the obtained models. The system input u is a white, random, uniformly distributed signal $u_k \sim U(-2; 2)$. $N = 10^4$ training, $3 \cdot 10^3$ validation and 10^4 test samples are generated with inde-

pendent realizations of e and u. Note that this system for zero input has two stable equilibria at $x = \pm[0.68\ 0.68]^\top$ and one unstable equilibrium point at $x = [0\ 0]^\top$ (the largest eigenvalue of $\nabla_x f(x; 0)|_{x=[0\ 0]^\top}$ is 1.23). Hence, (42) is not strictly contractive, but it is stable in the sense of Condition 1 which has been checked numerically.

5.2. Model estimation

At this point, the system under study is only disturbed by measurement noise, hence, the SUBNET structure is simplified to an output error noise structure (i.e. f does not depend on $\hat{e}_k$). During the analysis, we have varied one hyper-parameter while keeping all the others equal to the following base values: $T = 40$, $n = 10$, $n_x = 4$. All the functions h, f and are parametrized using 2 hidden layer feedforward neural networks with 64 nodes per layer, linear bypass and IO normalization. To compute the model estimate, the Adam optimizer (Kingma & Ba, 2015) with a batch size of 256 has been used with default learning rate of 10^{-3} and early stopping using a validation set.

We chose a base model order $n_x = 4$ as it will be shown that *immersion effects* are observed under the considered ANN complexity for h, f and . Immersion is a well-known phenomenon (Lee & Marcus, 1988; Ohtsuka, 2005), corresponding to the fact that at the price of introducing extra state variables, the original system can be equivalently represented by less complex state-transition and output functions, till reaching the class of linear, but often infinite dimensional representations, coined in the literature as Koopman forms (Williams, Hemati, Dawson, Kevrekidis, & Rowley, 2016). As will be shown, this trade-off between n_x and the layer-depth of the involved ANNs makes the optimization problem more stable and ensures relatively fast convergence.

The performance of the estimated models is characterized by the *normalized root mean square* (NRMS) simulation error:

$$\text{NRMS} = \frac{\text{RMS}}{\sigma_y} = \frac{\sqrt{\frac{1}{N}\sum_{k=1}^{N} \|y_k - \hat{y}_k\|_2^2}}{\sigma_y}; \tag{43}$$

where σ_y is the *sample standard deviation* of y and NRMS% $=$ NRMS $\times$ 100. In this case, an independent noiseless data set has been used to calculate the NRMS simulation error to clearly characterize the accuracy of the estimated model. The estimated encoder is used to initialize the state for every simulation. Hence, the first n outputs are not used to compute (43) as they have been used to feed the encoder.

5.3. Computational cost and performance

The computational cost and performance of the proposed subspace encoder based method is compared to existing methods in Fig. 2 and Table 1. For the comparison, multiple ANN-based state–space identification methods are considered: (i) the classical SS-ANN simulation error minimization method which simulates the entire training range (T = training data length) starting from an initial state which is estimated together with the system parameters (Parameter init OE) (Schoukens & Ljung, 2019), (ii) the unconstrained multiple-shooting method for SS-ANNs which adds the initial state of each considered simulation section to the parameter vector without (Parameter init no-overlap) (Bock, 1981) and with (Parameter init overlap) overlapping simulation sections, and finally (iii) the considered subspace encoder method without (Encoder init no-overlap) and with (Encoder init overlap) overlapping simulation sections.

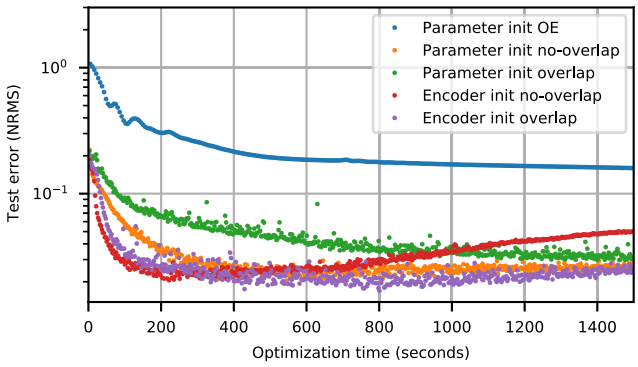


Fig. 2. Evolution of the NRMS simulation error of the estimated models by the considered approaches w.r.t. the test data. The keywords "encoder init" and "parameter init" indicate if either encoder-based prediction or parametric estimation is used to estimate the initial states, "overlap" and "no-overlap" indicate if the subsections can overlap, while "OE" stands for simulation based cost over the entire data sequence with no subsections.

Table 1
Performance of the compared approaches (see Section 5.3) given the same training budget of 25 min (and same hardware).

Combination	NRMS test
Parameter init OE	15.9%
Parameter init no-overlap	2.0%
Parameter init overlap	3.0%
Encoder init no-overlap	2.1%
Encoder init overlap	1.7%

Fig. 2 shows that the "Parameter init OE" approach is indeed the slowest as it needs to simulate the entire training data set to perform one optimization step. It also shows that the encoder provides improved performance compared to parametrization of the initial condition and co-estimating it with the model. Furthermore, the overlap variants of the methods suffer less from overfitting than standard multiple-shooting. This is in line with the variance reduction obtained from the overlapping subsections as shown in Section 4.2.

5.4. Truncation length (T)

A key parameter of the encoder method is the truncation length T. Fig. 3 illustrates that overfitting is a significant issue when T is smaller than the dominant timescale of the system. However, a bigger T increases the computational cost. To choose T in an informed manner, we employ *expected normalized k-step-error* plots in Fig. 4 in terms of

$$\text{NRMS}_{\text{k-step}} = \frac{1}{\sigma_y}\sqrt{\frac{1}{N-k}\sum_{t=1}^{N-k} \|\hat{y}_{t+k|t} - y_{t+k}\|_2^2}; \tag{44}$$

where $\hat{y}_{t+k|t}$ corresponds to the k-step-ahead prediction by the estimated model relative to sample t. These show that, for the current system under test, the transient error has a length of 30 time steps, which approximately corresponds to the characteristic timescale of the system. Furthermore, Fig. 4 shows that the error in the output is larger for low values of k compared to large values of k. Hence, the estimated state at $k = 0$ is less accurate than

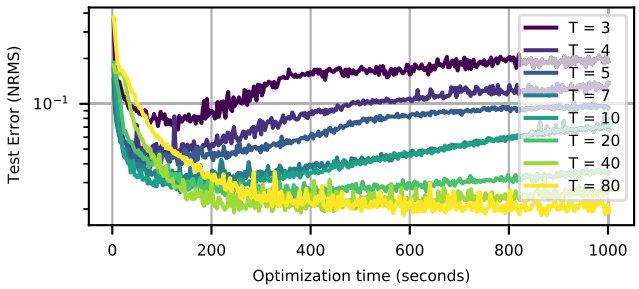


Fig. 3. Influence of the truncation length T of the loss function on the test error during training.

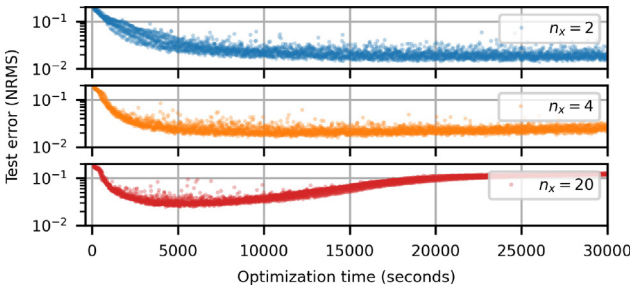


Fig. 5. The influence of the model order n_x on the resulting test error of the estimates during training. The displayed results are based on 7 models trained using different random initial parameter values.

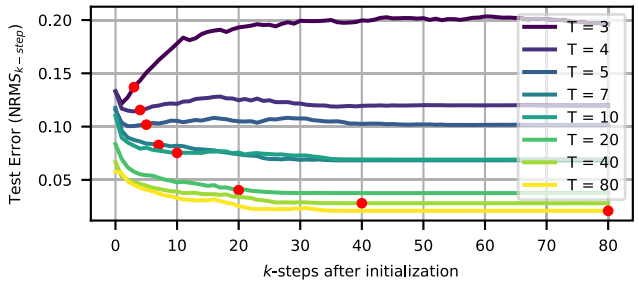


Fig. 4. The k-step NRMS error of estimated models under different truncation lengths, computed on the test data. The red dots indicate the truncation length T.

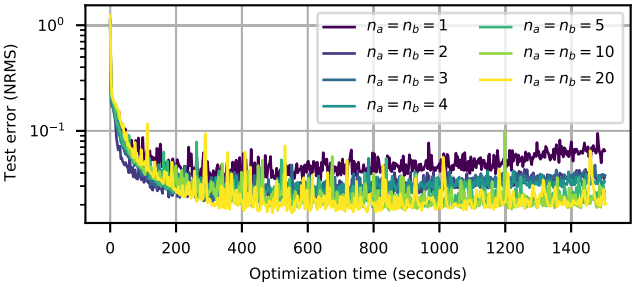


Fig. 6. The influence of the lag window n_a and n_b, i.e. the horizon of past data used by the encoder, on the test error during training.

the state obtained after a number of steps. This suggests that the encoder is unable to accurately recover/estimate the initial state, i.e. the encoder estimate has a high variance, indicating that the encoder parametrization is not sufficiently rich ($n = 10$ is insufficient or the used number of layers/neurons is too low).

5.5. Model order (n_x)

Changing the state order of the model has also significant influence on the behavior of the estimation as shown in Fig. 5, where the obtained results are averaged over 7 identification runs. n_x values that are much larger than the order of the true data-generating system quickly result in overfitted model estimates. Using the true state dimension $n_x = 2$ results in a model with the lowest obtained NRMS, but the variability of the obtained models over the 7 identification runs is quite high, and the optimization time is significantly larger than for $n_x = 4$. As discussed earlier, we suspect that this can be attributed to an effect similar to immersion where providing additional degrees of freedom makes the functions f, h and less complex and hence simpler to estimate up to the point where the variance of the sheer number of extra parameters overtakes this advantage.

The 7 different runs, each initialized with different random parameters, converge to the same test error level in NRMS. This suggests that the encoder method gives reproducible results for different initializations which highlight insensitivity of the method for initialization.

5.6. Encoder window length (n)

Varying the lag window n (or n_a and n_b considered separately for u and y) of the encoder in terms of the considered past IO samples shows in Fig. 6 that the minimal required $n = n_a = n_b = n_x - 1 = 1$, with n_x the order of the system for state reconstructability, on which the encoder is based on, is not the optimal value in this case. As seen from the figure, $20 \geq n = n_a = n_b > 4$ perform much better. As we argued earlier, this can be contributed to a variance reduction in the state estimate (i.e. encoder as a minimum variance observer) which reduces the average transient error. However, for values that are much larger than $n_x - 1 = 1$ (around 20), the computational cost significantly increases without too much gain in performance. Lastly, the choice of lag windows has a less strong impact on performance than the choice of T and n_x.

5.7. Neural network depth and width

The influence of varying the depth and width of the neural networks used to parametrize , f and h is illustrated in Fig. 7. A network structure that is too complex results in a growing computational cost and variance of the model estimates. Whereas a network structure that is too simple results in under-fitting as the model is unable to capture the dynamics of the true underlying system. Overfitting is suppressed in our approach by the regularization introduced with early stopping, the stochastic gradient descent algorithm, and the increased data efficiency allowing for overlap in the subsections. Hence, neural network structure selection is more sensitive to underparametrization than overparametrization for the proposed method.

5.8. Estimation under process noise

The subspace-encoder based estimation approach has been introduced to provide reliable model estimates under general

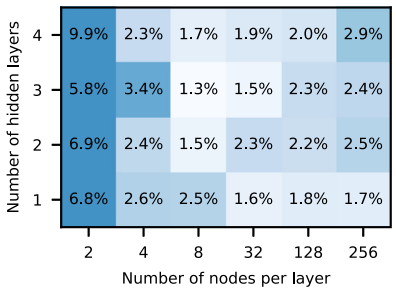


Fig. 7. The influence of the neural network architecture on the model estimates in terms of the achieved NRMS simulation error on test data.

noise conditions that can be described in an innovation form. To demonstrate this property, we extend (42) with a process noise term. First we will consider the case when the noise enters the state equation linearly, then we will consider the case when the noise enters nonlinearly:

$$x_{k+1}^{(1)} = \frac{x_k^{(1)}}{1{:}2 + x_k^{(2)\,2}} + 0{:}4x_k^{(2)} + g_1(x_k^{(1)}; e_k); \tag{45a}$$

$$x_{k+1}^{(2)} = \frac{x_k^{(2)}}{1{:}2 + x_k^{(1)\,2}} + 0{:}4x_k^{(1)} + u_k + g_2(x_k^{(2)}; e_k); \tag{45b}$$

$$y_k = x_k^{(1)} + e_k; \tag{45c}$$

where $g_i(x_k^{(i)}; e_k) = K^{(i)}e_k$ for the linear and $g_i(x_k^{(i)}; e_k) = K^{(i)}x_k^{(i)}e_k$ for the nonlinear case respectively, K , $_K K_0 = \|K_0\|_2$ with K_0 , $[1{:}0 \quad -0{:}9]^\top$, and $_K \geq 0$ is an adjustable parameter which regulates how strong the process noise is affecting the state. The noise e_k is generated by a white Gaussian noise process with a standard deviation $_e = 0{:}082$ (resulting in 20 dB SNR) for the training data and $_e = 0$ for the test data.

We compare estimation with the subspace encoder method under three different noise models: *OE noise:* there is no process noise considered in the model, *linear innovation:* $\hat{e}_{t+k|t}$ appears linearly in f , i.e. in the state Eq. (7a) of the model and *general innovation:* $\hat{e}_{t+k|t}$ is passed through the neural network f . The resulting NRMS of the simulation error over the test data for the three considered models can be viewed in Tables 2 and 3 under linear and nonlinear innovation noise in the data-generating system, respectively. To indicate the significance of the results these tables also contain the standard deviation of the mean which is the sample standard deviation divided by $\sqrt{4-1}$ since there are 4 independent samples considered. The linear case shows that for $_K > 0{:}5$, both the linear and the nonlinear parametrization of the process noise model structure significantly reduces the test error in comparison with the OE model. For the nonlinear case, the nonlinear parametrization outperforms the linear innovation noise model structure for $_K > 0{:}5$, as expected.

6. Benchmark results

The Wiener–Hammerstein benchmark (Schoukens & Ljung, 2009) is an electronic circuit with a diode-resistor nonlinearity and has been used in benchmarking a wide variety of nonlinear system identification methods. As in Beintema et al. (2021b), we split the data set into 80,000 training, 20,000 validation and 78,000 test samples. Similarly as before, we utilize the same network structure and OE noise structure for all three networks in the model, but now with a single hidden layer with 15 hidden nodes and a linear bypass. The hyperparameters used are $n_x = 6$, $T = 80$, $n = n_a = n_b = 50$, a batch size of 1024, early stopping, input/output normalization, and the Adam optimizer with default learning rate of $= 10^{-3}$.

Table 2
Mean and mean standard deviation of the NRMS% of the simulation error on the test data over 4 independent runs, when **linear** innovation noise is present in the data-generating system.

$_K$	OE noise	Linear innovation noise model	Nonlinear innovation noise model
0.0	1.8 ± 0.1	**1.7** ± 0.1	1.8 ± 0.1
0.25	2.1 ± 0.2	**1.9** ± 0.1	2.1 ± 0.1
0.5	2.3 ± 0.1	**1.9** ± 0.1	2.1 ± 0.1
1.0	3.2 ± 0.1	2.2 ± 0.1	**2.0** ± 0.2
2.0	4.8 ± 0.1	3.0 ± 0.1	**2.5** ± 0.0
4.0	8.4 ± 0.3	5.7 ± 0.4	**4.4** ± 0.1

Table 3
Same as Table 2 but for **nonlinear** innovation noise.

$_k$	OE noise	Linear innovation noise model	Nonlinear innovation noise model
0.0	1.8 ± 0.1	**1.7** ± 0.0	1.8 ± 0.1
0.25	2.2 ± 0.1	2.2 ± 0.1	**1.8** ± 0.1
0.5	2.8 ± 0.1	2.4 ± 0.1	**1.8** ± 0.1
1.0	4.1 ± 0.1	3.2 ± 0.1	**2.3** ± 0.2
2.0	8.2 ± 0.3	6.5 ± 0.4	**4.3** ± 0.3
4.0	14.2 ± 0.8	12.6 ± 0.6	**10.5** ± 0.3

The obtained results are displayed in Table 4 together with results achieved by other approaches on this benchmark. Many of these methods have been developed for Wiener–Hammerstein identification problems and use varying level of structural knowledge about the system. Remarkably, the proposed approach is able to get the lowest test error results using no structural knowledge or guided initialization as some of the methods that are reported on this benchmark. Note however that, while good performance is obtained quite rapidly, a long optimization (over 200 h) is required to fine tune the estimate and obtain the final result. This aspect can be further improved by using learning rate schedulers and higher-order optimization methods.

7. Conclusion

By carefully combining elements from machine learning (batch optimization), multiple-shooting (multi-step-ahead loss), and subspace identification (encoder), a subspace encoder-based ANN method for nonlinear state–space identification has been introduced. The method is proven to be locally consistent, to have a relatively smooth cost function by using a multiple shooting strategy, and to be more data efficient than other ANN-based strategies by considering overlapping subsections. The method has shown state-of-the-art results in our simulation studies and on a widely used benchmark identification problem. Remarkably, it does not need specialized initialization of the model parameters to achieve this performance or structural knowledge of the system and besides of some rules of thumb, it is also relatively insensitive to the choices of its hyper parameters.

Table 4
Results of the subspace encoder on the Wiener–Hammerstein benchmark compared to results reported in the literature.

Identification method	Test RMS simulation (mV)	Test NRMS simulation
Subspace Encoder	**0.241**	**0.0987%**
QBLA (Schoukens, Pintelon, & Rolain, 2014)	0.279	0.113%
Pole-zero splitting (Sjöberg, Lauwers, & Schoukens, 2012)	0.30	0.123%
NL-LFR (Schoukens & Tóth, 2020)	0.30	0.123%
PNLSS (Paduart, Lauwers, Pintelon, & Schoukens, 2012)	0.42	0.172%
Generalized WH (Wills & Ninness, 2009)	0.49	0.200%
LS-SVM (Falck, Pelckmans, Suykens, & De Moor, 2009)	4.07	1.663%
Bio-social evolution (Naitali & Giri, 2016)	8.55	3.494%
SS auto-encoder (Masti & Bemporad, 2021)	12.01	4.907%
Genetic Programming (Khandelwal, 2022)	23.50	9.605%
SVM (Marconato & Schoukens, 2009)	47.40	19.373%
BLA (Lauwers, Pintelon, & Schoukens, 2009)	56.20	22.969%

References

Allgöwer, F., & Zheng, A. (2012). *Nonlinear model predictive control, Vol. 26*. Birkhäuser.

Beintema, G. I., Tóth, R., & Schoukens, M. (2021a). Non-linear state-space model identification from video data using deep encoders. *IFAC-PapersOnLine*, *54*(7), 697–701.

Beintema, G. I., Tóth, R., & Schoukens, M. (2021b). Nonlinear state-space identification using deep encoder networks. In *The proc. of machine learning research (3rd annual learning for dynamics & control conference), Vol. 144* (pp. 241–250). Zurich.

Billings, S. A. (2013). *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. Wiley.

Birpoutsoukis, G., Marconato, A., Lataire, J., & Schoukens, J. (2017). Regularized nonparametric Volterra kernel estimation. *Automatica*, *82*, 324–327.

Bock, H. G. (1981). Numerical treatment of inverse problems in chemical reaction kinetics. In *The proc. of modelling of chemical reaction systems* (pp. 102–125). Springer.

Boyd, S., & Chua, L. (1985). Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Transactions on Circuits and Systems*, *32*(11), 1150–1161.

Darouach, M., & Zasadzinski, M. (1997). Unbiased minimum variance estimation for systems with unknown exogenous inputs. *Automatica*, *33*(4), 717–719.

Decuyper, J., Dreesen, P., Schoukens, J., Runacres, M. C., & Tiels, K. (2019). Decoupling multivariate polynomials for nonlinear state-space models. *IEEE Control Systems Letters*, *3*(3), 745–750.

Decuyper, J., Runacres, M., Schoukens, J., & Tiels, K. (2020). Tuning nonlinear state-space models using unconstrained multiple shooting. *IFAC-PapersOnLine*, *53*(2), 334–340.

Falck, T., Pelckmans, K., Suykens, J. A., & De Moor, B. (2009). Identification of Wiener–Hammerstein systems using LS-SVMs. *IFAC Proceedings Volumes*, *42*(10), 820–825.

Forgione, M., Mejari, M., & Piga, D. (2022). Learning neural state-space models: do we need a state estimator? arXiv preprint arXiv:2206.12928.

Forgione, M., & Piga, D. (2021). Continuous-time system identification with neural networks: Model structures and fitting criteria. *European Journal of Control*, *59*, 69–81.

Gedon, D., Wahlström, N., Schön, T. B., & Ljung, L. (2021). Deep state space models for nonlinear system identification. *IFAC-PapersOnLine*, *54*(7), 481–486.

Giri, F., & Bai, E.-W. (2010). *Block-oriented nonlinear system identification, Vol. 1*. Springer.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *The proc. of the thirteenth international conference on artificial intelligence and statistics* (pp. 249–256). JMLR Workshop and Conference Proceedings.

Isidori, A. (1985). *Nonlinear control systems: An introduction*. Springer.

Isidori, A., Sontag, E., & Thoma, M. (1995). *Nonlinear control systems, Vol. 3*. Springer.

Jansson, M. (2003). Subspace identification and ARX modeling. *IFAC Proceedings Volumes*, *36*(16), 1585–1590.

Katayama, T., et al. (2005). *Subspace methods for system identification, Vol. 1*. Springer.

Khandelwal, D. (2022). *Springer theses, Automating data-driven modelling of dynamical systems: An evolutionary computation approach*. Springer.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *International conference on learning representations*.

Lauwers, L., Pintelon, R., & Schoukens, J. (2009). Modelling of Wiener–Hammerstein systems via the best linear approximation. *IFAC Proceedings Volumes*, *42*(10), 1098–1103.

Lee, H.-G., & Marcus, S. (1988). Immersion and immersion by nonsingular feedback of a discrete-time nonlinear system into a linear system. *IEEE Transactions on Automatic Control*, *33*(5), 479–483.

Lee, L. H., & Poolla, K. (1999). Identification of linear parameter-varying systems using nonlinear programming. *Journal of Dynamic Systems, Measurement, and Control*, *121*(1), 71–78.

Ljung, L. (1978). Convergence analysis of parametric identification methods. *IEEE Transactions on Automatic Control*, *23*(5), 770–783.

Marconato, A., & Schoukens, J. (2009). Identification of Wiener–Hammerstein benchmark data by means of support vector machines. *IFAC Proceedings Volumes*, *42*(10), 816–819.

Masti, D., & Bemporad, A. (2021). Learning nonlinear state–space models using autoencoders. *Automatica*, *129*, Article 109666.

Naitali, A., & Giri, F. (2016). Wiener–Hammerstein system identification – an evolutionary approach. *International Journal of Systems Science*, *47*(1), 45–61.

Ohtsuka, T. (2005). Model structure simplification of Nonlinear Systems via immersion. *IEEE Transactions on Automatic Control*, *50*(5), 607–618.

Paduart, J., Lauwers, L., Pintelon, R., & Schoukens, J. (2012). Identification of a Wiener–Hammerstein system using the polynomial nonlinear state space approach. *Control Engineering Practice*, *20*(11), 1133–1139.

Paduart, J., Lauwers, L., Swevers, J., Smolders, K., Schoukens, J., & Pintelon, R. (2010). Identification of nonlinear systems using polynomial nonlinear state space models. *Automatica*, *46*(4), 647–656.

Priestley, M. B. (1988). *Non-linear and non-stationary time series analysis* (pp. 59–63). London: Academic Press.

Ribeiro, A. H., Tiels, K., Umenberger, J., Schön, T. B., & Aguirre, L. A. (2020). On the smoothness of nonlinear system identification. *Automatica*, *121*, Article 109158.

Schön, T. B., Wills, A., & Ninness, B. (2011). System identification of nonlinear state-space models. *Automatica*, *47*(1), 39–49.

Schoukens, M. (2021). Improved initialization of state-space artificial neural networks. In *The proc. of the European control conference* (pp. 1913–1918).

Schoukens, J., & Ljung, L. (2009). Wiener–Hammerstein benchmark. In *LiTH-ISY-R*. Linköping University Electronic Press.

Schoukens, J., & Ljung, L. (2019). Nonlinear system identification: A user-oriented road map. *IEEE Control Systems Magazine*, *39*(6), 28–99.

Schoukens, M., Pintelon, R., & Rolain, Y. (2014). Identification of Wiener–Hammerstein systems by a nonparametric separation of the best linear approximation. *Automatica*, *50*(2), 628–634.

Schoukens, M., & Tiels, K. (2017). Identification of block-oriented nonlinear systems starting from linear approximations: A survey. *Automatica*, *85*, 272–292.

Schoukens, M., & Tóth, R. (2020). On the initialization of nonlinear LFR model identification with the best linear approximation. *IFAC-PapersOnLine*, *53*(2), 310–315.

Sjöberg, J., Lauwers, L., & Schoukens, J. (2012). Identification of Wiener–Hammerstein models: Two algorithms based on the best split of a linear model applied to the SYSID'09 benchmark problem. *Control Engineering Practice*, *20*(11), 1119–1125.

Sliwiński, P., Marconato, A., Wachel, P., & Birpoutsoukis, G. (2017). Non-linear system modelling based on constrained Volterra series estimates. *IET Control Theory & Applications*, *11*(15), 2623–2629.

Suykens, J. A. K., Moor, B. L. R. D., & Vandewalle, J. (1995). Nonlinear system identification using neural state space models, applicable to robust control design. *International Journal of Control*, *62*(1), 129–152.

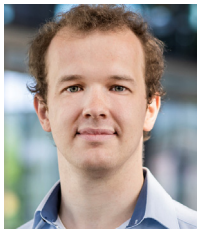
Tallec, C., & Ollivier, Y. (2017). Unbiasing truncated backpropagation through time. arXiv preprint arXiv:1705.08209.

Tóth, R. (2010). *Modeling and identification of linear parameter-varying systems, Vol. 403*. Springer.

Willems, J. C. (2013). Open stochastic systems. *IEEE Transactions on Automatic Control*, *58*(2), 406–421.

Williams, M. O., Hemati, M. S., Dawson, S. T., Kevrekidis, I. G., & Rowley, C. W. (2016). Extending data-driven Koopman analysis to actuated systems. *IFAC-PapersOnLine*, *49*(18), 704–709.

Wills, A., & Ninness, B. (2009). Estimation of generalised Hammerstein–Wiener systems. *IFAC Proceedings Volumes*, *42*(10), 1104–1109.



Gerben I. Beintema is a Doctoral Candidate at the Control Systems (CS) Group at the Department of Electrical Engineering. His current research is on the intersection of nonlinear system identification and deep learning, under the supervision of Associate Professor Roland Tóth and assistant Professor Maarten Schoukens. His main research interest is improving and understanding deep nonlinear state-space identification to obtain interpretable, robust, and generally applicable methods. Gerben I. Beintema obtained his B.Sc. degree in Applied Physics from the Eindhoven University of Technology (TU/e), cum laude, in 2019.

Maarten Schoukens is an Assistant Professor in the Control Systems group of the Department of Electrical Engineering at the Eindhoven University of Technology. He received a master's degree in electrical engineering and a Ph.D. degree in engineering from the Vrije Universiteit Brussel (VUB), Brussels, Belgium, in 2010 and 2015 respectively. From 2015 to 2017, he has been a Post-Doctoral Researcher with the ELEC Department, VUB. In October 2017 he joined the Control Systems research group, TU/e, Eindhoven, The Netherlands as a Post-Doctoral Researcher, in 2018 he became an Assistant Professor in the same group. Maarten was awarded an FWO Ph.D. Fellowship in 2011, a Marie Skłodowska-Curie Individual Fellowship in 2018, and an ERC Starting Grant in 2022. His main research interests include the measurement and data-driven modeling and control of nonlinear dynamical systems using system identification and machine learning techniques.

Roland Tóth received his Ph.D. degree with Cum Laude distinction at the Delft Center for Systems and Control (DCSC), Delft University of Technology (TUDelft) in 2008. He was a Post-Doctoral Research Fellow at TUDelft in 2009 and at the Berkeley Center for Control and Identification, University of California, Berkeley in 2010. He held a position at DCSC, TUDelft in 2011-12, then he joined to the Control Systems (CS) Group at the Eindhoven University of Technology (TU/e). Currently, he is an Associate Professor at the CS Group, TU/e and a senior researcher at the Systems and Control Laboratory, Institute for Computer Science and Control (SZTAKI) in Budapest, Hungary. His research interests are in identification and control of linear parameter-varying (LPV) and nonlinear systems, developing machine learning methods with performance and stability guarantees for modeling and control, model predictive control and behavioral system theory. On the application side, his research focuses on advancing reliability and performance of precision mechatronics and autonomous robots/vehicles with LPV and learning-based motion control. Dr. Tóth received the TUDelft Young Researcher Fellowship Award in 2010, the VENI award of The Netherlands Organisation for Scientific Research in 2011 and the Starting Grant of the European Research Council in 2016.