Deep Identification of Nonlinear Systems in Koopman Form

Lucian Cristian Iacob, Gerben Izaak Beintema, Maarten Schoukens and Roland Tóth

Abstract— **The present paper treats the identification of nonlinear dynamical systems using Koopman-based deep state-space encoders. Through this method, the usual drawback of needing to choose a dictionary of lifting functions a priori is circumvented. The encoder represents the lifting function to the space where the dynamics are linearly propagated using the Koopman operator. An input-affine formulation is considered for the lifted model structure and we address both full and partial state availability. The approach is implemented using the the deepSI toolbox in Python. To lower the computational need of the simulation error-based training, the data is split into subsections where multi-step prediction errors are calculated independently. This formulation allows for efficient batch optimization of the network parameters and, at the same time, excellent long term prediction capabilities of the obtained models. The performance of the approach is illustrated by nonlinear benchmark examples.**

I. INTRODUCTION

Nonlinear system identification is a wide and intensely researched topic, aiming at the estimation of dynamical systems directly form data. Multiple methods have been developed, where the commonly used model structures are: NARX, nonlinear state space, block-oriented, see [1] for an overview. However, even if the resulting models have good simulation or prediction performance, the representation of the system remains confined in the nonlinear system class. While several nonlinear control methods have been developed (e.g. feedback linearization, backstepping, sliding mode control, to name a few [17]), they are often complicated to apply and there is no systematic approach for shaping the performance of the closed-loop system in contrast to the approaches of the linear time invariant (LTI) framework. While LTI control is well advanced, designs are limited to operate in the neighbourhood of given linearization points. Hence, pressing needs in engineering led to the idea of developing various linear embeddings of nonlinear systems to apply the powerful LTI control methods with global stability and performance guarantees.

One such embedding technique is based on the Koopman framework, where the concept is to lift the nonlinear state space to a (possibly) infinite dimensional space through so-called observable functions. The dynamics of the original system are preserved and governed by the linear Koopman operator [4], [15]. In practice, if a dictionary of a finite number of observables is chosen a priori and used to construct time shifted data matrices, the linear Koopman-based model can be calculated via least squares [11]. One such approach, called Dynamic Mode Decomposition (DMD) [12], is based on constructing the time shifted data matrices using the original states of the system. If the dictionary consists of nonlinear functions of the state, this technique is known as Extended DMD (EDMD) [13]. Besides issues that arise with the presence of noise and biasedness of the estimates, the main difficulty lays with choosing the lifting functions such that, on the lifted state-space, an LTI model exists that can well capture the dynamic behavior of the original nonlinear system.

Learning the lifting functions from data has been addressed recently using Artificial Neural Networks (ANNs). A common approach is to construct autoencoders, to represent the lifting function and its inverse, [6]-[8], and enforce the linearity condition of the lifted state transition. Alternatively, [9] proposes to specify the entire dictionary of observables as the outputs of an ANN, and perform an EDMD type of regression for model estimation. While [9] addresses partial state observations, availability of full state measurements is a common assumption in the Koopman identification literature. Moreover, only a few papers present examples where measurement noise is present (e.g. [20]) and often only the robustness of the methods is analysed instead of ensuring stochastic consistency of the estimators. Furthermore, in this research area, the treatment of inputs has only recently been addressed, either through a nonlinear lift [5] or by using state and input dependent observables, together with input increments [7].

To address these issues, we introduce a Koopman-based state-space encoder model and a corresponding estimation method, implemented based on the deepSI toolbox[1] in Python [2]. We summarize the main characteristics and contributions as follows:

- Constructability-based formulation of a lifted space encoder (nonlinear mapping) based on deep-ANN
- Formulation of an identification approach for estimating Koopman models with Output Error (OE) noise structure in state-space using T-step ahead prediction
- Incorporation of lifted-state dependent linear effect of the input for general representation of nonlinear systems
- Construction that allows for both full and partial state availability

This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement nr. 714663).

The authors are with the Control System Group, Eindhoven University of Tehcnology, The Netherlands {l.c.iacob, g.i.beintema, r.toth, m.schoukens}@tue.nl

Roland Tóth is also with the Systems and Control Laboratory, Institute for Computer Science and Control, Budapest, Hungary

[1]deepSI toolbox available at https://github.com/GerbenBeintema/deepSI

- Computationally scalable implementation of the estimation via batch-wise (multiple-shooting) optimization of a single and relatively simple loss function, compared to the more complex training criteria in [6], [7].

The paper is structured as follows. Section II details the general Koopman framework and we discuss the notions of observability and state constructability in the Koopman form together with the role of inputs. Section III describes the proposed Koopman-based encoder and the associated model structure together with the used optimization method. In Section IV, the approach is tested using a Van der Pol oscillator and the Silverbox benchmark [10], followed by a discussion of the results. The conclusions are presented in Section V.

II. Behavior of Koopman embeddings

This section details the Koopman framework focusing on a finite dimensional lifted form. Next, we briefly discuss observability and constructability notions in the original and lifted forms. We show that, while the system behavior can be represented by a linear form, a nonlinear constraint remains on the initial conditions to ensure one-to-one connection between the solution sets.

A. Preliminaries

Consider a discrete-time nonlinear autonomous system:

$$x_{k+1} = f(x_k) \tag{1}$$

with $x : \mathbb{Z} \to \mathbb{R}^{n_\mathrm{x}}$ the state variable, $f : \mathbb{R}^{n_\mathrm{x}} \to \mathbb{R}^{n_\mathrm{x}}$ is a bounded nonlinear function and $k \in \mathbb{Z}$ is the discrete time step. The Koopman framework proposes an alternative representation of system (1) by introducing so-called observable functions $\phi \in \mathcal{F}$, with $\mathcal{F}$ a Banach function space and $\phi : \mathbb{R}^{n_\mathrm{x}} \to \mathbb{R}$. As described in [4], the Koopman operator $\mathcal{K} : \mathcal{F} \to \mathcal{F}$ associated with (1) and $\mathcal{F}$ is defined through:

$$\mathcal{K}\phi = \phi \circ f, \quad \forall \phi \in \mathcal{F} \tag{2}$$

where $\circ$ denotes function composition and (2) is equal to:

$$\mathcal{K}\phi(x_k) = \phi(x_{k+1}). \tag{3}$$

An important property of the Koopman operator is that it is linear when $\mathcal{F}$ is a vector space of functions [15]. Considering two observables $\phi_1, \phi_2 \in \mathcal{F}$ and scalars $\alpha_1, \alpha_2 \in \mathbb{R}$, if (2) holds, then it implies:

$$\begin{aligned} \mathcal{K}(\alpha_1\phi_1 + \alpha_2\phi_2) &= (\alpha_1\phi_1 + \alpha_2\phi_2) \circ f \\ &= \alpha_1\phi_1 \circ f + \alpha_2\phi_2 \circ f \\ &= \alpha_1\mathcal{K}\phi_1 + \alpha_2\mathcal{K}\phi_2, \end{aligned} \tag{4}$$

proving the linearity property. While often the existence of the Koopman operator requires $\mathcal{F}$ to be spanned by an infinite number of basis functions, for practical purposes, an n_f-dimensional linear subspace $\mathcal{F}_{n_\mathrm{f}} \subset \mathcal{F}$ is considered, with $\mathcal{F}_{n_\mathrm{f}} = \operatorname{span}\{\phi_j\}_{j=1}^{n_\mathrm{f}}$. As detailed in [4], with a projection operator $\Pi : \mathcal{F} \to \mathcal{F}_{n_\mathrm{f}}$, the finite-dimensional approximation of the Koopman operator $\mathcal{K}$ is given by:

$$\mathcal{K}_{n_\mathrm{f}} = \Pi\mathcal{K}|_{\mathcal{F}_{n_\mathrm{f}}} : \mathcal{F}_{n_\mathrm{f}} \to \mathcal{F}_{n_\mathrm{f}}. \tag{5}$$

In practice, the Koopman matrix representation $A \in \mathbb{R}^{n_\mathrm{f} \times n_\mathrm{f}}$ is used, such that the element-wise relation is satisfied:

$$\mathcal{K}_{n_\mathrm{f}}\phi_j = \sum_{i=1}^{n_\mathrm{f}} A_{ji}\phi_i. \tag{6}$$

For a more detailed analysis, one can consult [4]. Next, we introduce the lifted state $z_k = \Phi(x_k)$, where $\Phi(x_k) = \begin{bmatrix}\phi_1(x_k) & \dots & \phi_{n_\mathrm{f}}(x_k)\end{bmatrix}^\top$. The lifted finite dimensional linear representation of (1) is then given by:

$$z_{k+1} = Az_k. \tag{7}$$

However, the main difficulty of the Koopman framework is the choice of the lifting functions such that the resulting observables generate a Koopman invariant subspace [14]. Furthermore, it is often not clearly stated in the literature on the subject that a linear system whose dynamics are driven by the Koopman matrix A is only equivalent in terms of behavior (collections of all solution trajectories) with the original nonlinear system (1) if explicit nonlinear constraints are defined for the initial condition of the lifted state. We explore this next using a simple example.

B. Linear representations subject to nonlinear constraints

To illustrate the concept, consider a nonlinear system represented by (1) with a polynomial f, similar to the one described in [14]. Denote $x_{k,i}$ to be the i^{th} element of x_k. In this notation, the system dynamics are described as follows:

$$\begin{bmatrix} x_{k+1,1} \\ x_{k+1,2} \end{bmatrix} = \begin{bmatrix} ax_{k,1} \\ bx_{k,2} - cx_{k,1}^2 \end{bmatrix} \tag{8}$$

with constant parameters $a, b, c \in \mathbb{R}$. By considering solutions of (8) only on $[0, \infty)$ with initial condition $x_0 \in \mathbb{R}^{n_2}$, the feasible trajectories are given by:

$$\mathcal{B} = \left\{x : \mathbb{Z}_0^+ \to \mathbb{R}^2 \mid \text{s.t. (8) is satisfied}\right\}. \tag{9}$$

To represent the system in the Koopman form, the following observables are chosen: $\phi_1(x_k) = x_{k,1}$, $\phi_2(x_k) = x_{k,2}$ and $\phi_3(x_k) = x_{k,1}^2$. Then, the dynamics are represented by:

$$\Phi(x_{k+1}) = \underbrace{\begin{bmatrix} a & 0 & 0 \\ 0 & b & -c \\ 0 & 0 & a^2 \end{bmatrix}}_{A} \Phi(x_k). \tag{10}$$

Based on (10), consider the system $z_{k+1} = Az_k$ of dimension $n_\mathrm{z} = 3$, with $z_0 \in \mathbb{R}^3$; the solution set is described as:

$$\mathcal{B}_\mathcal{K} = \left\{z : \mathbb{Z}_0^+ \to \mathbb{R}^3 \mid \text{s.t. } z_{k+1} = Az_k\right\}. \tag{11}$$

Note that (11) represents an unrestricted LTI behavior. It is easy to show that $\Phi(\mathcal{B}) \subseteq \mathcal{B}_\mathcal{K}$, as any $z_k \in \mathcal{B}_\mathcal{K}$ with $z_0 \in \mathbb{R}^3$ for which $z_{0,3} \neq z_{0,1}^2$ will not correspond to a solution of (8), i.e. $\Phi^{-1}(z_k) = x_k \notin \mathcal{B}$. By introducing the constraint $\Psi : \mathbb{R}^3 \to \mathbb{R}, \Psi(z_k) = z_{k,1}^2 - z_{k,3}$, the solution set (11) with constraint Ψ is:

$$\hat{\mathcal{B}}_\mathcal{K} = \left\{z : \mathbb{Z}_0^+ \to \mathbb{R}^3 \mid \text{s.t. } z_{k+1} = Az_k,\ \Psi(z_0) = 0\right\}. \tag{12}$$

Then, it is possible to show that $\Phi(\mathcal{B}) = \hat{\mathcal{B}}_\mathcal{K}$. Our example shows that, to have a bijective relation between the solu-

tion sets, additional constraints need to be imposed on the Koopman form, or, as we call it now, embedding of (1).

C. Observability, constructability and extension to inputs

1) Autonomous case

Consider the system (1) having the output defined as:

$$y_k = h(x_k), \tag{13}$$

with the nonlinear output map $h : \mathbb{R}^{n_\mathrm{x}} \to \mathbb{R}^{n_\mathrm{y}}$. Given $x_0 \in \mathbb{R}^{n_\mathrm{x}}$, the observability map for the nonlinear system represented by (1) and (13) is:

$$\mathcal{O}_\mathrm{x}(x_0) = \begin{bmatrix} h(x_0) \\ h^{(1)}(x_0) \\ \dots \\ h^{(n_\mathrm{x}-1)}(x_0) \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \dots \\ y_{n_\mathrm{x}-1} \end{bmatrix} \tag{14}$$

with $h^{(i)}(x_0) = h(f^{(i)}(x_0))$ and $f^{(i)}$ is the composition of f with itself i times. As described in [18], the system satisfies the observability rank condition at x_0 if the rank of the analytical map[2] $\mathcal{O}_\mathrm{x} : \mathbb{R}^{n_\mathrm{x}} \to \mathbb{R}^{n_\mathrm{x} n_\mathrm{y}}$ is equal to n_x. If this condition is met, the system is strongly locally observable at $\bar{x} \in X$, where X is a neighbourhood of x_0 and there exists an analytic function $\mathcal{O}_\mathrm{x}^{-1} : \mathbb{R}^{n_\mathrm{x} n_\mathrm{y}} \to X$, such that $\mathcal{O}_\mathrm{x}^{-1}\left(\begin{bmatrix} y_0^\top & \dots & y_{n_\mathrm{x}-1}^\top \end{bmatrix}^\top\right) = x_0$.

In the Koopman form, assuming that the output function is in the span of the lifted states, i.e., $y_k = C z_k$, with $C \in \mathbb{R}^{n_\mathrm{y} \times n_\mathrm{z}}$, the observability map can be defined as follows:

$$\begin{bmatrix} y_0 \\ y_1 \\ \dots \\ y_{n_\mathrm{z}-1} \\ 0 \end{bmatrix} = \begin{bmatrix} \begin{pmatrix} C \\ CA \\ \dots \\ CA^{n_\mathrm{z}-1} \end{pmatrix} z_0 \\ \Psi(z_0) \end{bmatrix} = \mathcal{O}_\mathrm{z}(z_0) \tag{15}$$

where, as observed in section II-B, it is also necessary to consider the nonlinear constraints $\Psi : \mathbb{R}^{n_\mathrm{f}} \to \mathbb{R}^{n_\mathrm{c}}$. The map $\mathcal{O}_\mathrm{z}$ should be locally invertible to uniquely determine z_0, i.e. there exists $\mathcal{O}_\mathrm{z}^{-1} : \mathbb{R}^{n_\mathrm{z} n_\mathrm{y}} \to \mathbb{R}^{n_\mathrm{f}}$, such that $\mathcal{O}_z^{-1}\left(\begin{bmatrix} y_0^\top & \dots & y_{n_\mathrm{x}-1}^\top \end{bmatrix}^\top\right) = z_0$. This is a different point of view than in the work [22], where the observability notions are discussed based on an explicit definition of the lifting map. Similar to (15), the notion of constructability refers to uniquely determining z_0 using current and past measurements, that is, $\mathcal{R}_\mathrm{z}^{-1}\left(\begin{bmatrix} y_{-n_\mathrm{z}+1}^\top & \dots & y_0^\top \end{bmatrix}^\top\right) = z_0$ with $\mathcal{R}_\mathrm{z}^{-1} : \mathbb{R}^{n_\mathrm{z} n_\mathrm{y}} \to \mathbb{R}^{n_\mathrm{f}}$. $\mathcal{R}_\mathrm{z}$ is the constructability map and $\mathcal{R}_\mathrm{z}^{-1}$ denotes its inverse. In the proposed ANN implementation, we formulate the encoder as a nonlinear function in order to reconstruct a state that can be associated with the Koopman form of the nonlinear system.

2) Systems with input

To extend the considerations to the non-autonomous case, we consider the class of nonlinear control-affine systems:

$$x_{k+1} = f(x_k) + g(x_k) u_k, \tag{16}$$

with a potential nonlinear function $g : \mathbb{R}^{n_\mathrm{x}} \to \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{u}}$ and input $u : \mathbb{Z} \to \mathbb{R}^{n_\mathrm{u}}$. The treatment of the inputs in the lifted form is a topic of debate with many different approaches present in the literature. In general, an LTI form is assumed due to its ease of use with existing linear control methods [16]. However, this form may be insufficient to capture the nonlinear behavior of (16). Based on the results of [22] for continuous time, we consider an input-affine Koopman form:

$$\Phi(x_{k+1}) = A\Phi(x_k) + B(\Phi(x_k))u_k. \tag{17}$$

We argue that, although (17) is more complex than an LTI model, it provides a better approximation capability of the dynamics of (16).

Let $z_k = \Phi(x_k)$ and $y_k = C z_k$. The observability map $\Theta_z : \mathbb{R}^{n_\mathrm{z} n_\mathrm{u} n_\mathrm{z}} \to \mathbb{R}^{n_\mathrm{z} n_\mathrm{y} + n_\mathrm{c}}$ is described as:

$$\begin{aligned} (\Theta_z(z_0, \mathbf{u}))^\top &= \begin{bmatrix} y_0^\top & y_1^\top & y_2^\top & \dots & y_{n_\mathrm{z}-1}^\top & 0^\top \end{bmatrix}^\top \\ &= \mathcal{O}_\mathrm{z}(z_0) + \begin{bmatrix} 0 \\ CB(z_0)u_0 \\ CAB(z_0)u_0 + CB(\zeta^0(z_0, u_0))u_1 \\ \dots \\ \zeta^{n_\mathrm{z}-1}(z_0, \mathbf{u}) \\ 0 \end{bmatrix} \end{aligned}$$

where, for ease of readability, $\zeta^0(z_0, u_0) = Az_0 + B(z_0)u_0$, $\zeta^{n_\mathrm{z}-1}(z_0, \mathbf{u})$ represents a nonlinear function containing all the remaining terms in the expansion of $y_{n_\mathrm{z}-1}$ and $\mathbf{u} = [u_0, \dots, u_{n_\mathrm{z}-1}]$. As can be seen, for the lifted form (17), to determine z_0 for future input and output data requires the inversion of an even more complex nonlinear map. The same holds for constructability, where the aim is to determine z_0 based on past input-output data. Similar to the autonomous case, the encoder is formulated as a nonlinear function that estimates the inverse of the constructability map to determine the lifted state using past measurements.

III. Identification framework

Based on the the constructability map and the state-dependent affine mapping for the input discussed in Section II, we have now all the ingredients to develop an identification method for a data-driven Koopman embedding of a nonlinear system without prior selection of the observables. Due to the shown nonlinearity of the constructability map, a deep-ANN-based function estimator is needed to determine the state basis z.

A. Data generating system

Similar to (16), the data generating system is considered to be a nonlinear control affine system:

$$x_{k+1} = f(x_k) + g(x_k) u_k \tag{18a}$$

$$y_k = h(x_k) + v_k \tag{18b}$$

with $v_k \in \mathbb{R}^{n_\mathrm{y}}$ being an additive zero-mean (possibly coloured) noise. The stochastic system in (18) corresponds to an OE noise setting where our objective is to estimate a model of the deterministic (process) part. Next, we define the chosen model structure for the lifted Koopman form.

B. Model structure

We apply the estimation concept from [2], [3] and develop an implementation of the resulting method using deepSI. To represent the input contribution in the lifted model,

[2]The rank of the Jacobian matrix of $\mathcal{O}_\mathrm{x}$ at x_0.

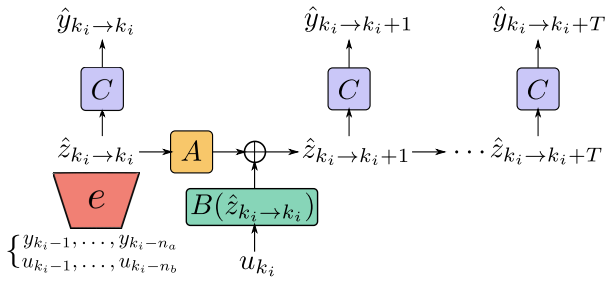

Fig. 1: Network architecture. The lifted state at moment k_i, $\hat{z}_{k_i \to k_i}$, is estimated using the encoder function e based on previously measured input and output data.

we consider an input-affine formulation. The chosen model structure is:

$$\hat{z}_{k+1} = A_\theta \hat{z}_k + B_\theta(\hat{z}_k) u_k \tag{19a}$$

$$\hat{y}_k = C_\theta \hat{z}_k \tag{19b}$$

where $\hat{z}_k \in \mathbb{R}^{n_z}$ is the lifted state, $u_k \in \mathbb{R}^{n_u}$ is the input, $\hat{y}_k \in \mathbb{R}^{n_y}$ is the model output and y_k is the measured system output. In the proposed model (19), A_θ is the Koopman matrix and $B_\theta(z)$ is a nonlinear function of the lifted state. We construct the model using a linear output map C_θ such that the outputs of the original model are spanned by the lifting functions. If state measurements are available, we can also enforce that the states can be recovered by a linear mapping. The neural network is constructed using the encoder function e_θ and the nonlinear map B_θ (both implemented using feedforward neural nets), and the linear maps A_θ and C_θ. The subscript θ represents the parameters (weights and biases) of the neural network. The main advantage of using the Koopman model structure (19a) is that it can be viewed as a Linear Parameter Varying (LPV) system, for which numerous control techniques have been developed (see [21]).

The orders n_a and n_b and their selection corresponds to the classical problem of model structure selection in system identification and hence it is out of the scope of the current paper. Furthermore, the proposed network architecture can also handle full state measurements, in this case the output function h being an identity function.

C. Cost function and estimation procedure

The computation of the simulated response and the corresponding gradient for ANN optimization has a heavy computational cost, which can become intractable when large data sets are used. To deal with this, a trade-off proposed in [2], [3] is to construct a cost function using subsections of the data set and, starting from an index k_i, perform a T-step ahead prediction. The cost function is formulated as follows:

$$V_{\text{enc}}(\theta) = \frac{1}{2N(T+1)} \sum_{i=1}^{N} \sum_{p=0}^{T} \|\hat{y}_{k_i \to k_i+p} - y_{k_i+p}\|_2^2 \tag{20a}$$

$$\hat{y}_{k_i \to k_i+p} := C_\theta \hat{z}_{k_i \to k_i+p} \tag{20b}$$

$$\hat{z}_{k_i \to k_i+p+1} := A_\theta \hat{z}_{k_i \to k_i+p} + B_\theta(\hat{z}_{k_i \to k_i+p}) u_{k_i+p} \tag{20c}$$

where $\hat{z}_{k_i \to k_i+p}$ is computed through p recursive iterations of (19a) starting from $\hat{z}_{k_i \to k_i}$. The initial lifting to $\hat{z}_{k_i \to k_i}$ is performed through the encoder function e_θ as follows:

$$\hat{z}_{k_i \to k_i} := e_\theta\left(x_{k_i-1}, u_{k_i-1}\right) \tag{21a}$$

$$\hat{z}_{k_i \to k_i} := e_\theta(y_{k_i-n_a:k_i-1}, u_{k_i-n_b:k_i-1}), \tag{21b}$$

where e_θ estimates the inverse of the constructability map, using past input-output data to determine the lifted state $\hat{z}_{k_i \to k_i}$. The notation $y_{k_i-n_a:k_i-1}, u_{k_i-n_b:k_i-1}$ represents the sets of past outputs and inputs. In the case of full state availability, for numerical reasons, the initial lifted state $\hat{z}_{k_i \to k_i}$ is computed via the encoder function e_θ based on the previous time step of the original state x_{k_i-1} and input u_{k_i-1} (21a).

D. Batch optimization

As detailed in [2], eq. (20a) allows for the parallelization of the computations, allowing for the cost of each section to be computed individually. As such, the computation time is greatly decreased and, moreover, a batch cost function can be utilized, summing over only a subset of sections:

$$V_{\text{batch}}(\theta) = \frac{1}{2N_{\text{batch}}(T+1)} \sum_{i \in B} \sum_{p=0}^{T} \|\hat{y}_{k_i \to k_i+p} - y_{k_i+p}\|_2^2 \tag{22}$$

with $B \subset \{1, 2, \ldots, N\}$. This reformulation offers the possibility of using powerful optimization algorithms such as Adam [19].

IV. Experiments and results

We demonstrate the performance of the proposed method on the simulation example of an autonomous Van der Pol oscillator with full state measurements and the Silverbox benchmark system, which is a real-world setup with only input-output data available.

A. Van der Pol

We consider the dynamics of an unforced Van der Pol oscillator [17]:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= \mu(1 - x_1^2(t))x_2(t) - x_1(t) \end{aligned} \tag{23}$$

with $\mu = 1$. The continuous time system is discretized using the Runge-Kutta 4 numerical formula with a sampling frequency of 20 Hz. Training, validation and test trajectories are generated starting from initial conditions that are uniformly distributed $x_0 \sim \mathcal{U}(-2, 2)$, each trajectory having a length of 501 data points. White Gaussian noise v is added to the simulated state trajectories such that a Signal-to-Noise Ratio (SNR) of 20 dB is achieved per each individual channel (note that the test data is noiseless). For training, 80 sets of trajectories, for validation, 20 sets and, for testing, 10 sets are generated.

The lifting function e_θ given by (21a) (without the input term) is implemented as a feedforward neural network, with 1 hidden layer, 100 nodes and tanh activation. The parameters are initialized by sampling from a uniform distribution $\mathcal{U}(-\sqrt{m}, \sqrt{m})$, with $m = 1/\sqrt{n_{\text{in}}}$, and n_{in} represents the number of inputs to the layer. We consider a lifting dimension $n_z = 100$, a prediction horizon value $T = 149$ and a

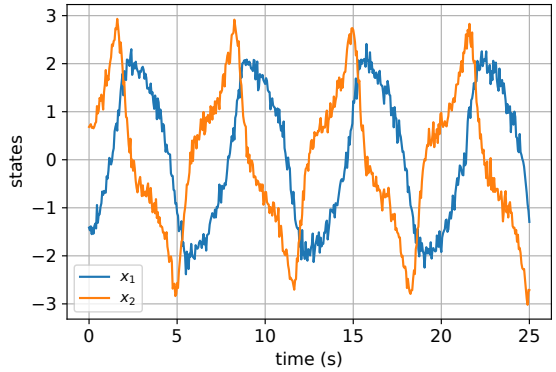

Fig. 2: State trajectories of the Van der Pol oscillator with added noise used for training.

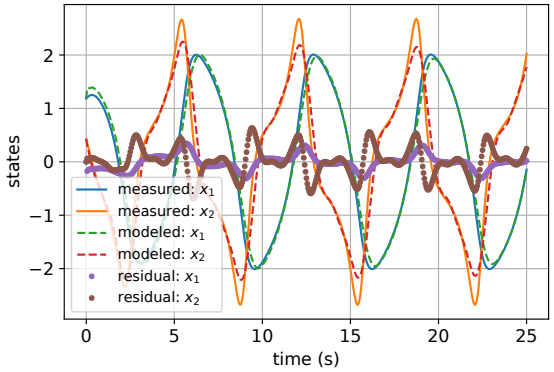

Fig. 3: Comparison of the noiseless test data from the Van der Pol oscillator and the the simulation response of the estimated Koopman model.

batch size of 256. For training, Adam batch optimization [19] is used, with a learning rate of $\alpha = 10^{-4}$ and the exponential decay rates set to: $\beta_1 = 0.7$ and $\beta_2 = 0.9$. We utilize early stopping, as described in [2], by computing the simulation error on the validation data set after each epoch. We then select the parameters for which the validation cost is minimal. After the training phase, the model along the epochs with the lowest simulation error is used for analysis.

Fig. 2 shows a set of noisy time-domain trajectories used as training data. Fig. 3 depicts one realization of the noiseless test set and the simulated state trajectories of the identified Koopman model, alongside the residuals. We observe that the proposed Koopman-based encoder is able to capture the oscillating dynamics of the test system with acceptable simulation error. This behavior can also be observed in the phase portrait depicted in Fig. 4. The quality of the model is assessed in terms of the Normalized Root Mean Square (NRMS) and RMS errors:

$$\text{NRMS} = \frac{\text{RMS}}{\sigma_y} = \frac{\text{mean}\left(\sqrt{\frac{1}{M-k_0+1}\sum_{k=k_0}^{M}\|\hat{y}_k - y_k\|_2^2}\right)}{\sigma_y} \tag{24}$$

where the total RMS error is computed as the mean of the RMS error per section of data and σ_y is the standard deviation of all test outputs. M is the total length of a section of test data and k_0 is the starting point ($k_0 = \max(n_a, n_b)+1$) for the input-output case and $k_0 = 1$ for the full state availability case). In terms of this error measure, the following results are obtained on the test data by the estimated Koopman model:

$$\text{NRMS} = 0.12 \qquad \text{RMS} = 0.18,$$

where the given error measures are the mean NRMS and RMS over the two state trajectories. These errors can be mostly attributed to the mismatch at the peaks of the sharp rising slopes, as seen in Fig. 2. However, the identified linear system based on the full state Koopman encoder is able to represent the nonlinear dynamics of the Van der Pol oscillator, successfully recovering the limit cycle. The results are quite satisfactory given that noisy training and validation data sets with 10% noise (in terms of power) are used.

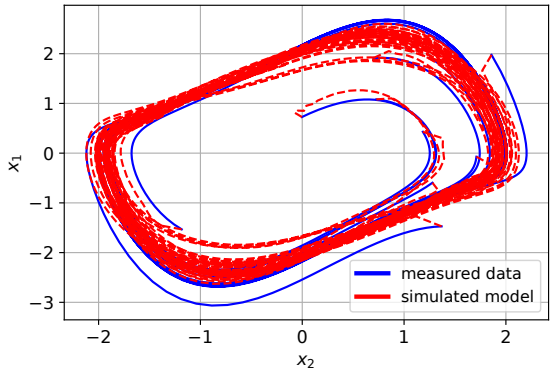

Fig. 4: Phase portrait of the noiseless state response of the Van der Pol oscillator in the test data and simulated state trajectories of the identified Koopman model.

B. *Siverbox*

To illustrate the capabilities of the proposed Koopman encoder structure when no state measurements are available, we use measured data from the Silverbox benchmark [10], which is a real-world electrical implementation of a mass-spring-damper system with a cubic spring, similar to the forced Duffing oscillator. The first part of the input signal is a filtered Gaussian noise sequence with linearly increasing amplitude, and the rest is generated as a multisine signal. Fig. 5 shows the separation of data, with the note that both the multisine and filtered Gaussian (arrowhead part) are used for testing and assessing the quality of the identified model.

For this experiment, both encoder e_θ (21b) and nonlinear input function $B(z)$ are implemented through feedforward neural networks, the former with 2 hidden layers and the latter with 1, each having 40 neurons per layer. The encoder settings are: $n_z = 20$, $T = 49$, $n_a = n_b = 10$ and a batch size of 256 is used. The initial parameters are sampled from a uniform distribution as detailed in the Van der Pol example and the same Adam batch optimization algorithm is used. The learning rate is set to $\alpha = 10^{-3}$ and the exponential decay rates are chosen as $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

As illustrated in Fig. 6, the identified model can accurately

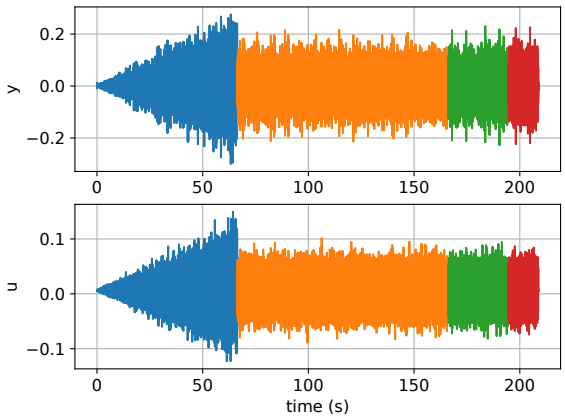

Fig. 5: Separation of the Silverbox data: arrowhead test , train , validation , test .

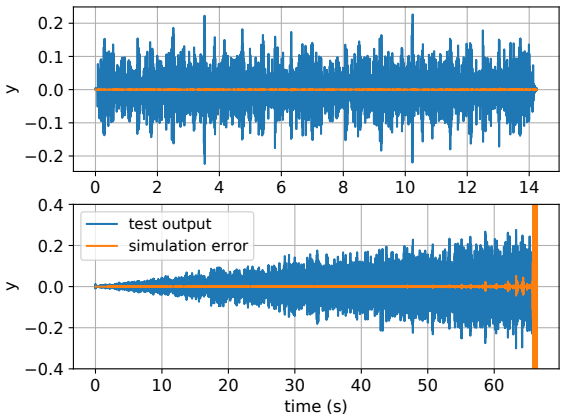

Fig. 6: Silverbox multisine test (top) and arrowhead test (bottom).

represent the dynamics of the original system, when a multisine test signal is applied. However, when the arrowhead test input data is applied, the error highly increases towards the end of the simulation, as Fig. 6 depicts. The problem in the extrapolation region is due to a mismatch between the representation of the nonlinearity in the model and the true polynomial nonlinearity. Methods which explicitly use polynomial basis may perform better in this regard [2]. Table I presents the NRMS and RMS error values. If we discard the extrapolation region of the arrowhead test signal, the obtained errors are comparable to the state of the art [2].

TABLE I: ERROR MEASURES

	NRMS	RMS (V)
Test	0.00552	0.00029
Arrowhead	229.411	12.2502
Arrowhead - no extrapol.	0.00811	0.00033

V. CONCLUSION

The present paper formulates a Koopman identification method as a nonlinear identification problem, using a neural network estimator consistent with the inverse of the constructability map. Furthermore, the effect of the input is accounted for in the Koopman model through an input-affine description. We have shown that this approach can successfully capture the dynamics of the underlying nonlinear system through motivating examples, for both full and partial state availability.

REFERENCES

[1] Schoukens, J. and Ljung, L., "Nonlinear System Identification: A User-Oriented Roadmap," *IEEE Control Systems Magazine*, Volume 39, no. 6, pp. 28-99, 2019

[2] Beintema, G., Tóth, R. and Schoukens., M., "Nonlinear state-space identification using deep encoder networks," *Proceedings of Learning for Dynamics and Control (L4DC)*, 2021

[3] Beintema, G., Tóth, R. and Schoukens., M., "Non-linear State-space Model Identification from Video Data using Deep Encoders," *19th IFAC Symposium on System Identification (SYSID)*, 2021

[4] Mauroy, A., Mezić, I. and Susuki, Y., *The Koopman Operator in Systems and Control*, Springer International Publishing, 2020

[5] Bonnert, M. and Konigorski, U., "Estimating Koopman Invariant Subspaces of Excited Systems Using Artificial Neural Networks," *IFAC-PapersOnLine*, Volume 53, Issue 2, pp. 1156-1162, 2020

[6] Lusch, B., Kutz, J.N. and Brunton, S.L., "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature Communications,* Volume 9, Article no. 4950, 2018

[7] Heijden, B.V.D., Ferranti, L., Kober, J. and Babuška, R., "DeepKoCo: Efficient latent planning with an invariant Koopman representation," *arXiv preprint*, arXiv:2011.12690, 2020

[8] Otto, S.E. and Rowley, C.W., "Linearly Recurrent Autoencoder Networks for Learning Dynamics", *SIAM Journal on Applied Dynamical Systems*, Volume 18, Issue 1, pp. 558-593, 2019

[9] Yeung, E., Kundu, S. and Hodas, N.O., "Learning Deep Neural Network Representations for Koopman Operators of Nonlinear Dynamical Systems," *American Control Conference (ACC)*, pp. 2832-4839, 2019

[10] Wigren, T. and Schoukens, J., "Three free data sets for development and benchmarking in nonlinear system identification," *European Control Conference (ECC)*, pp. 2933-2938, 2013

[11] Mauroy, A. and Goncalves, J., "Koopman-Based Lifting Techniques for Nonlinear Systems Identification," *IEEE Transactions on Automatic Control*, Volume 65, no. 6, pp. 2550-2565, 2020

[12] Rowley, C., Mezić, I., Bagheri, S., Schlatter, P. and Henningson, D., "Spectral analysis of nonlinear flows," *Journal of Fluid Mechanics*, Volume 641, pp. 115-127, 2009

[13] Williams, M.O., Kevrekidis, I.G. and Rowley, C.W., "A Data–Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition," *Journal of Nonlinear Science*, Volume 25, Issue 6, pp. 1307-1346, 2015

[14] Brunton, S.L., Brunton, B.W., Proctor, J.L. and Kutz, J.N., "Koopman Invariant Subspaces and Finite Linear Representations of Nonlinear Dynamical Systems for Control," *The Public Library of Science ONE*, Volume 11, Issue 2, 2016

[15] Brunton, S., Budisic, M., Kaiser, E. and Kutz, J., "Modern Koopman Theory for Dynamical Systems," *arXiv preprint*, arXiv:2102.12086, 2021

[16] Korda, M. and Mezic, I., "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, Volume 93, pp. 149-160, 2018

[17] Khalil, H.K., *Nonlinear Systems*, Third Edition, Prentice Hall, 2002

[18] Nijmeijer, H., "Observability of autonomous discrete time non-linear systems: a geometric approach," *International Journal of Control*, Volume 36, no. 5, pp. 867-874, 1982

[19] Kingma, D.P. and Ba, J., "Adam: A Method for Stochastic Optimization," *International Conference on Learning Representations (ICLR)*, 2015

[20] Takeishi, N., Kawahara, Y. and Yairi, T., "Learning Koopman Invariant Subspaces for Dynamic Mode Decomposition," *International Conference on Neural Information Processing Systems (NIPS)*, 2017

[21] Mohammadpour, J. Scherer and C.W., *Control of Linear Parameter Varying Systems with Applications*, Springer-Verlag New York, 2012

[22] Surana, A., "Koopman Operator Based Observer Synthesis for Control-Affine Nonlinear Systems," *IEEE 55th Conference on Decision and Control (CDC)*, pp.6492-6499, 2016