Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Estimating Koopman Invariant Subspaces of Excited Systems Using Artificial Neural Networks

Marcel Bonnert, Ulrich Konigorski *

* *Control Systems and Mechatronics Laboratory, Technical University of Darmstadt, 64283 Darmstadt, Germany (e-mail: [mbonnert, ukonigorski]@iat.tu-darmstadt.de).*

Abstract: In recent years, the Koopman operator was the topic of many extensive investigations in the nonlinear system identification community. Especially, when dealing with nonlinear systems no straight forward method is available to identify systems of this class. In modern data science a standard method is using artificial neural networks to extract models from data. This method is mainly used when there is a certain function behind the measured data, but little other information is available. This paper combines the Koopman framework and artificial neural networks to achieve a linear model for nonlinear systems. The structure of the network is similar to an autoencoder. The input part is the encoder which itself consists of two different parts. The first part propagates the measurements directly to the middle part. The second encoder introduces the state-space lifting which characterizes the Koopman framework. The middle layer of the network represents an estimation of a linear state-space system that acts on a Koopman operator invariant subspace. After this layer, the extended state-space must be decoded so that the outputs of the Koopman linear system are functions of the true states. The method is evaluated with a single pendulum and a nonlinear yeast glycolysis model. Additionally, we show the advantage of considering inputs as true inputs rather than additional states.

Keywords: Koopman operator, System identification, Data Science, Neural networks, Optimization, Dynamical system

1. INTRODUCTION

To face problems like creating control strategies for technical systems a mathematical model is vital. This model should be able to recreate the behavior of the system with as few restrictions as possible. Mathematical models of dynamical systems can be achieved in two ways. Precisely, at first there is an analytical and at second there is a data-based way. The second method is often called system identification in that specific context. First research in this topic was done in the mid 1960s by e.g. Åström and Bohlin (1965).

Those mathematical models can be separated in a variety of ways. The most common one categorizes them in white-, gray- and black-box models. A white-box model is characterized by fully known dynamics. This can be achieved by using physical laws which are often based on differential equations and related parameters. In contrast to this, a black-box identification model is mostly data-based. It is used when almost no information about the internal structure of a system, but data is available. Everything in between is called a gray-box model. In fact, in practical applications often there is only a little information about the process available. This might be for example the systems order. According to Ljung (1999) it affects the identified model in a positive way the more information about the underlying system is fed into the identification process. In this paper we will examine data-based black-box identification.

At the beginning of an identification we need to set up the method. Mostly this also requires the selection of a certain system model. In the best case the model reproduces the system behavior globally, i.e. the mathematical model does not depend on a certain equilibrium point of the dynamical system. This is especially important in the context of nonlinear systems with many equilibria. In many practical applications the nonlinear behavior is reproduced by a manifold of linear systems which are built around certain equilibrium points. Depending on the region of validity of each individual linear model a huge number of linear substitutes might be necessary.

Since most of the identification algorithms are based on linear systems, mostly transfer functions or linear state-space equations, the identified system is not capable to reproduce the output of a nonlinear system in the entire state-space. Facing this problem, one of the first approaches was made by holding onto linear models but adding nonlinearities like saturation to the input and/or the output. Those are also known as Hammerstein, Wiener or Hammerstein-Wiener models.

A special problem, when identifying dynamical systems, is the input selection. This describes the process of selecting certain variables that might influence another considered variable which therefore is related to model selection. This occurs for example, when we estimate a polynomial as an approximation for the first equation of a nonlinear state-space system $x_{1,k+1} = f_1(x_{2,k})$. One can clearly see, that $x_{1,k}$ does not influence $x_{1,k+1}$. When computing an estimation for $x_{1,k+1}$ by considering $x_{1,k}$ and $x_{2,k}$ as inputs we hope that the parameters related to monomials in $x_{1,k}$ will be estimated to zero or at least small in comparison to the constants related to monomials in $x_{2,k}$. But due to noise, possibly neglected nonlinearities and numerical effects quite the opposite is the case. Depending on the nonlinearities and chosen monomials, the selected algorithm might find an even worse solution when only $x_{2,k}$ is considered.

A popular approach is choosing a certain number of models and selecting the best one according to a certain loss function. Even though the model selection results using Dynamic Mode Decomposition (DMD) in Mangan et al. (2017) were promising, it suffers from high order models with large polynomial order or a huge number of candidate functions for nonlinearities in general.

Another still not completely answered and structure estimation related problem concerns the identification of systems with nonlinear dynamics, i.e. $\dot{x} = -x^3$. A modern approach to this problem is the Koopman operator. This linear operator propagates so called *state observables* of a nonlinear system forward in time. In general, this operator is of infinite dimension. Thus, the nonlinearities of a finite dimensional system are traded for a linear system of infinite dimension.

Since an infinite-dimensional system cannot be handled, much research of the past few years addressed the problem of approximating a finite dimensional invariant subspace of the Koopman operator in the form of a state-space system and its matrices. Not only it is not guaranteed that such a subspace exists, as seen in Kutz et al. (2018), the solution is also highly dependent on the choice of the aforementioned observables. Furthermore, the estimation does not necessarily converge to trajectories of the base system with an increasing number of observables.

Thus, choosing these observable functions influences the identification results vigorously. Therefore, much research of the past few years approached this problem by using artificial neural networks (ANNs) like Takeishi et al. (2017), Lusch et al. (2018) or Yeung et al. (2019) to mention a few. Most of them used an ANN structured similar to an autoencoder.

This autoencoder structure is used in several system identification approaches as an alternative to the intensively used Singular Value Decomposition to extract the most important parts of the system dynamics from data. Erichson et al. (2019) used it to extract the most significant features of a fluid flow. In a more related topic Li et al. (2017) investigated the use of neural networks which lift the states to a higher dimension and used this as a dictionary followed by an iterative process where the Koopman operator and the observables are trained in alternating order. Otto and Rowley (2017) extended this idea so it can be used with small sets of data with high dimensional systems. In contrast to the encoder approach Li et al. (2019) used a graph neural network to estimate Koopman observables.

A different approach to the above ones was conducted by Mardt et al. (2018). The authors proposed a way to estimate stochastic processes using the variational approach for Markov processes (VAMP) with deep learning. In this framework one can define scores of VAMP-r family, Wu and Noé (2020). The VAMP-2 score used in Mardt et al. (2018) can then be maximized to achieve a better suited model to given data.

Our black-box (dark-gray-box) identification approach proposed in this paper builds mainly on the work of Lusch et al. (2018) and Yeung et al. (2019). Furthermore, it differs from the above methods in two ways. On the one hand, it introduces the possibility to include a control term to develop linear controllers for the Koopman linear system achieved by our method. On the other hand, we introduce linear layers in parallel to the encoders. These layers can be used to ensure that the state vector occurs directly in the Koopman operator subspace. Moreover, the linear layers can be used to feed the algorithm predefined observables that might be beneficial to use.

Introducing inputs to these promising ANN based methods is of great interest since the input of a system is the interface for control engineers. Additionally, inputs complicate the entire framework, since excited systems may have an infinite number of equilibria in comparison to their unexcited counterparts.

For the ANN approach it is necessary to take a closer look to systems with inputs. The EDMD approach on the one hand with a previously chosen set of observables delivers an additional column in the solution matrix which is dedicated to this certain input. The ANN approach on the other hand cannot handle this in the same way because the observables delivered by the encoder network is then a function of the state and the input and cannot be simply separated into a system- and an input-matrix necessary for a linear controller design. One of the ultimate goals of the Koopman framework is controlling nonlinear systems with extended *linear* controllers as shown in Fig. 1. Please note that an input decoder $\underline{\mathrm{g}}_w^{-1}\left(\underline{\mathrm{w}}_k\right)$ is not estimated in this paper as this will be a topic for future research. This controller in its entirety is nonlinear but with known nonlinearities combined with a linear state-controller of a Koopman linear system. To use this structure, we need a Koopman linear system with a dedicated input which is again a function of the system's input.

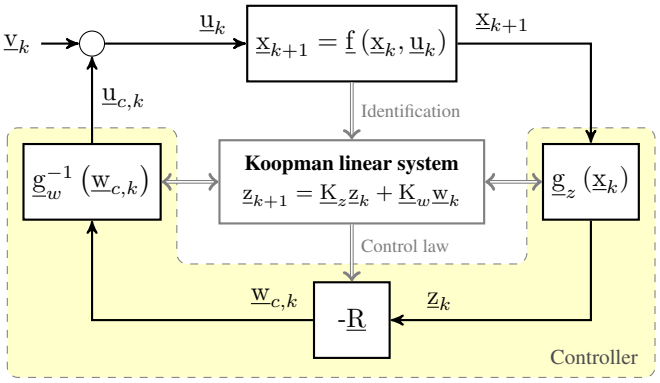

Fig. 1. Koopman control loop. The gray double lined arrows indicate a dependency but no information is transmitted.

The paper is separated into six sections. First, we shortly introduce the used nomenclature. Second, we describe the Koopman framework that is used throughout the paper. In section 4 we explain the developed method in detail followed by section 5, where the method is tested using a nonlinear yeast glycolysis system and a single pendulum. At the end, we summarize the achieved results and give a short outlook for future research topics.

2. NOTATION

In what follows vectors and matrices are illustrated with an underline and as lowercase and uppercase letters, respectively. Scalars will be shown as normal letters. A single measurement of the i-th state-variable at time step k is written as $x_{i,k}$.

To compute meaningful estimations for the system our first goal must be a good fit to training, validation and possibly test data. To evaluate the fit, we use the mean relative squared error (MRSE) of the output

$$\mathrm{fit}_{\mathrm{MRSE}}[\%] = 100\left(1 - \frac{1}{n}\sum_{i=1}^{n}\sqrt{\frac{\sum_{k=1}^{N}\left(x_{i,k} - \hat{x}_{i,k}\right)^2}{\sum_{k=1}^{N} x_{i,k}^2}}\right) \tag{1}$$

with the number of samples N and the number of outputs n.

3. KOOPMAN FRAMEWORK

The general idea of the Koopman operator dates to the year 1931 and was presented by Koopman (1931). He introduced an infinite-dimensional linear operator $\mathcal{K}$ acting on an infinite-dimensional Hilbert space $\mathcal{H}$ spanned by eigenfunctions $\varphi_j\left(\underline{\mathrm{x}}\right) : \mathbb{R}^n \to \mathbb{R}$ to evolve nonlinear dynamics forward in time linearly. Due to the linear character of the Koopman operator, observable functions $g_i\left(\underline{\mathrm{x}}\right) \in \mathcal{H}$ can be created by a weighted sum of the eigenfunctions. These observables evolve with the equation

$$\mathcal{K}g(\underline{\mathrm{x}}_k) = g \circ \underline{\mathrm{f}}$$

with $\mathcal{K} : \mathcal{H} \to \mathcal{H}$. This idea was rediscovered by Mezić (2005) in the context of dynamical systems a few years ago. As we will see this method is not restricted to a specific use-case, instead quite the opposite is the case. It is a general framework to expand the dimension or change the basis of the state-space to represent certain nonlinear behavior as linear modes in an infinite-dimensional Hilbert-space $\mathcal{H}$ using measurement coordinates or observable functions $g_i(\underline{\mathrm{x}})$. Those observables can also be understood as states z_i of a linear state-space system. In general, those observables include all imaginable combinations and modifications of the states. For example $z_l = x_3 \cdot \sin\left(x_1 x_2^2\right)$ may be one of them. Choosing meaningful states is a topic on its own and needs experience or strategies like the one presented in Brunton et al. (2016).

Since the Koopman operator is infinite-dimensional, what cannot be handled, we seek for a finite dimensional invariant subspace of the Koopman operator. By assuming observables as states z_i the linear state-space system

$$\underline{\mathrm{z}}_{k+1} = \underline{\mathrm{K}}\underline{\mathrm{z}}_k.$$

with $\underline{\mathrm{K}} : \mathbb{R}^{n_e} \to \mathbb{R}^{n_e}$ can be defined assuming $\underline{\mathrm{z}} \in \mathcal{H}$ holds. To be precise the derived state-space must not be of higher order, i.e. $n_e > n$. It can also be transformed to a different basis with a nonlinear transformation with $n_e = n$ and handled linear in those coordinates as seen in Lusch et al. (2018) or Kaiser et al. (2018).

As presented by Proctor et al. (2018) the idea of Koopman can be extended, so that an external excitation can be used. We will neglect mixed observables ($\underline{\mathrm{g}}(\underline{\mathrm{x}}, \underline{\mathrm{u}}) = \underline{0}$) due to the chosen structure displayed in Fig, 1. Thus, the observables we will use are only functions of $\underline{\mathrm{x}}$ or $\underline{\mathrm{u}}$ i.e. $\underline{\mathrm{g}}_z(\underline{\mathrm{x}}) : \mathbb{R}^n \to \mathbb{R}^{n_e}$ and $\underline{\mathrm{g}}_w(\underline{\mathrm{u}}) : \mathbb{R}^m \to \mathbb{R}^{m_e}$. The final Koopman linear state-space equation will then be

$$\underline{\mathrm{z}}_{k+1} = \underline{\mathrm{K}}_z\underline{\mathrm{z}}_k + \underline{\mathrm{K}}_w\underline{\mathrm{w}}_k \tag{2}$$

with the extended Koopman state-space $\underline{\mathrm{z}} \in \mathbb{R}^{n_e}$ and Koopman input space $\underline{\mathrm{w}} \in \mathbb{R}^{m_e}$.

4. DEEP KOOPMAN OPERATOR LEARNING WITH CONTROL

In the presented work we look at nonlinear state-space systems

$$\dot{\underline{\mathrm{x}}} = \underline{\mathrm{f}}\left(\underline{\mathrm{x}}, \underline{\mathrm{u}}\right)$$

or their discrete time equivalent

$$\underline{\mathrm{x}}_{k+1} = \underline{\mathrm{f}}\left(\underline{\mathrm{x}}_k, \underline{\mathrm{u}}_k\right). \tag{3}$$

In this section we propose the *Deep Koopman Operator Learning with Control* (DKLc) algorithm. It mainly builds on certain previous works of Lusch et al. (2018) and Yeung et al. (2019). They introduced ways to estimate a Koopman operator subspace from data using deep learning. Lusch et al. (2018) used an autoencoder structure with a linear layer after the encoder. This linear layer has a certain structure backed with an auxiliary network to achieve a parameterized linear system. With these modifications it was possible to estimate systems with a continuous spectrum such as a simple single pendulum or even complex fluid flows. Yeung et al. (2019) used a similar structure but they used a network to encode measured states to observable functions and used the transformed data to compute an estimation for the Koopman operator with Dynamic Mode Decomposition.

Our contribution is extending the general idea of those two methods to allow for inputs and the possibility to give the model restrictions. This is done by arranging multiple layers in parallel before the Koopman linear state-space layer. The network is shown in Fig. 2. The linear layers are used to feed the extended linear state-space model the states and inputs directly. Furthermore, previously chosen observable functions like x_1x_2 can be included with those linear layers. Therefore, we can give the estimation knowledge about the system and reduce the degrees of freedom in a heuristic way.

Additionally, this specific structure avoids the occurrence of mixed observables $g_i\left(\underline{\mathrm{x}}, \underline{\mathrm{u}}\right)$ which reduces the complexity if it is not necessary. Furthermore, we assume that the output space of the Koopman operator can be restricted to be solely in the function-space of the state observables. A deeper insight is given in Appendix A.

Our artificial neural network model is pictured in Fig. 2. It consists of five parts:

[1] Linear encoders
[2] Nonlinear encoders
[3] Koopman operator estimation layer
[4] Linear state-decoder
[5] Nonlinear decoders

The linear encoders [1] have only one not trainable layer with the identity matrix $\underline{\mathrm{I}}$ of appropriate dimensions as weighting and a bias term equal to zero. The nonlinear encoders [2] deliver the observables $\tilde{z}_i$ and $\tilde{w}_j$ of the states and excitation, respectively. Both the output of the linear and the nonlinear encoders are then concatenated and fed to the Koopman operator estimation layer [3].

Since we want to extract a linear discrete state-space system from the network's layer weights, the Koopman operator estimation layer [3] has a linear activation function. The idea is that this layer propagates the extended state vector $\underline{\mathrm{z}}_k$ one time

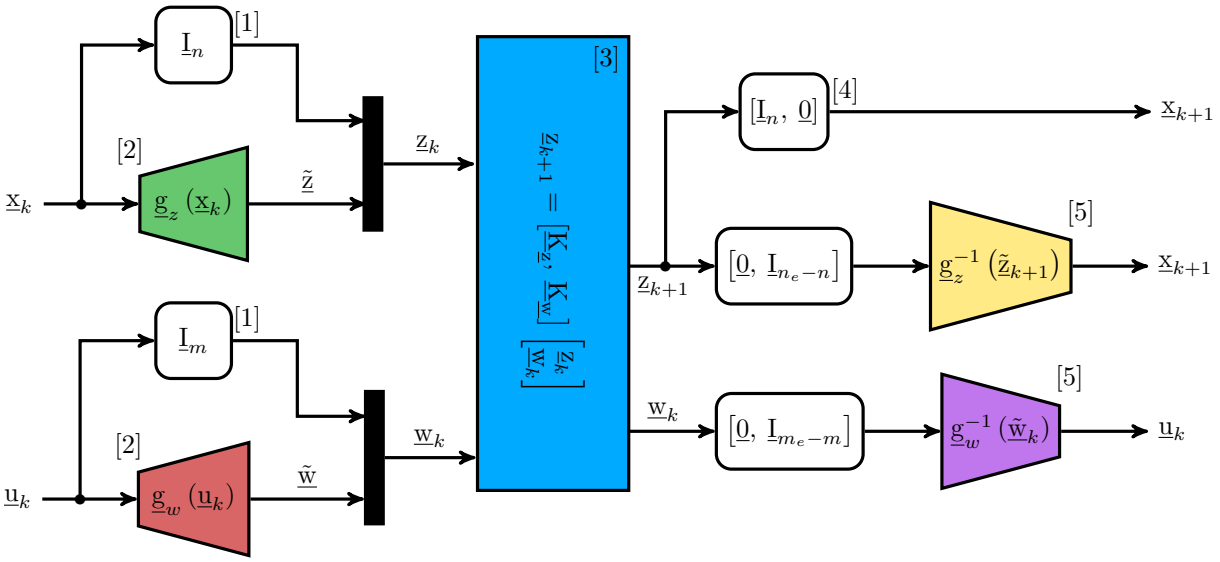

Fig. 2. Network model to identify a finite dimensional subspace approximately invariant to the infinite-dimensional Koopman operator of a nonlinear system. The linear layer in the middle holds the Koopman state-equation from equation (2).

step into the future. Thus, the output of the Koopman operator estimation layer is given by

$$\underline{\mathrm{z}}_{k+1} = \left[\underline{\mathrm{K}}_{\mathrm{z}},\ \underline{\mathrm{K}}_{\mathrm{w}}\right] \begin{bmatrix} \underline{\mathrm{z}}_k \\ \underline{\mathrm{w}}_k \end{bmatrix}.$$

To ensure that the first n outputs of layer [3] deliver the true states or the previously chosen observables we propagate them directly to one output of the network through the linear state-decoder [4]. If this would be the only output, it is not guaranteed that the remaining $n_e - n$ outputs of layer [3] are indeed functions of the state $\underline{\mathrm{x}}$. Thus, we need to ensure that the additional states passed through the Koopman operator estimation layer, are functions of the state $\underline{\mathrm{x}}$ again. To do so, we feed the remaining outputs of layer [3] to the nonlinear state-decoder $\underline{\mathrm{g}}_z^{-1}\left(\tilde{\underline{\mathrm{z}}}\right)$ which is the inverse function if the state-encoder $\underline{\mathrm{g}}_z\left(\underline{\mathrm{x}}\right)$. The nonlinear excitation-decoder delivers the inverse function $\underline{\mathrm{g}}_w^{-1}\left(\tilde{\underline{\mathrm{w}}}\right)$ of the excitation-encoder $\underline{\mathrm{g}}_w\left(\underline{\mathrm{u}}\right)$ that is used to compute the true excitation from the excitation-observables. This function can be used to compute the control signal passed to the nonlinear model as shown in Fig. 1.

5. EXAMPLES

5.1 Single Pendulum

As a seemingly easy introductory example we will use the excited frictionless pendulum

$$\dot{\underline{\mathrm{x}}} = \begin{bmatrix} x_2 \\ -\sin\left(x_1\right) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{4}$$

with all parameters set to 1. We stimulate the system with $u = 0.5\sin\left(2t\right)$ and an initial state $\underline{\mathrm{x}}_0 = \left[\pi/2,\ 0\right]^\top$. The amplitude of the input signal is limited to avoid chaotic behavior.

The used ANN consists of three layers of $[10, 20, 10]$ nodes in the state-encoder. The state-decoder has the same structure. To approximate the Koopman operator subspace we added two

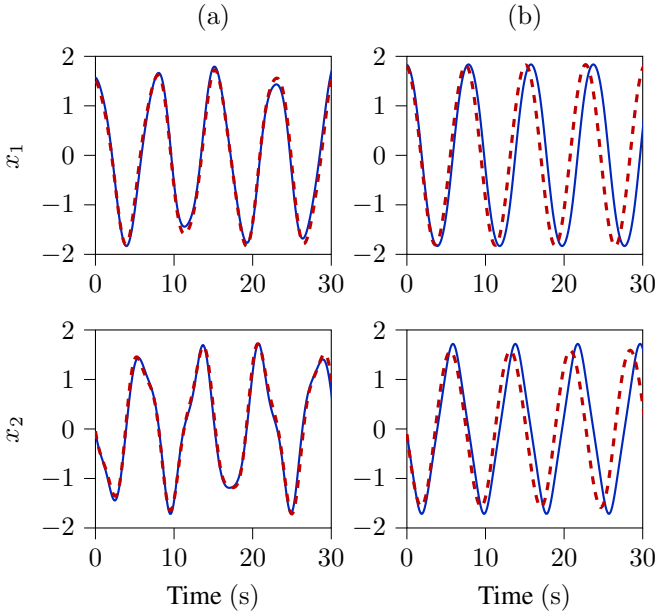

Fig. 3. Single pendulum measurement (–) versus prediction (– –)

observables to the two original states. We neither used the encoder nor the decoder for the excitation.

The two plots on the left in Fig. 3 ((a)) show that our model is able to reconstruct the systems behavior just as seen in Lusch et al. (2018) but with a present excitation. Furthermore, the two plots on the right show the autonomous behavior of our model. The model is able to reconstruct the dynamics quite accurately in magnitude but not in frequency. Thus, the influence of the input is not estimated exactly. This problem could be faced by the consideration of more than one sample to predict the future outputs in one estimation step. This is a topic of future research.

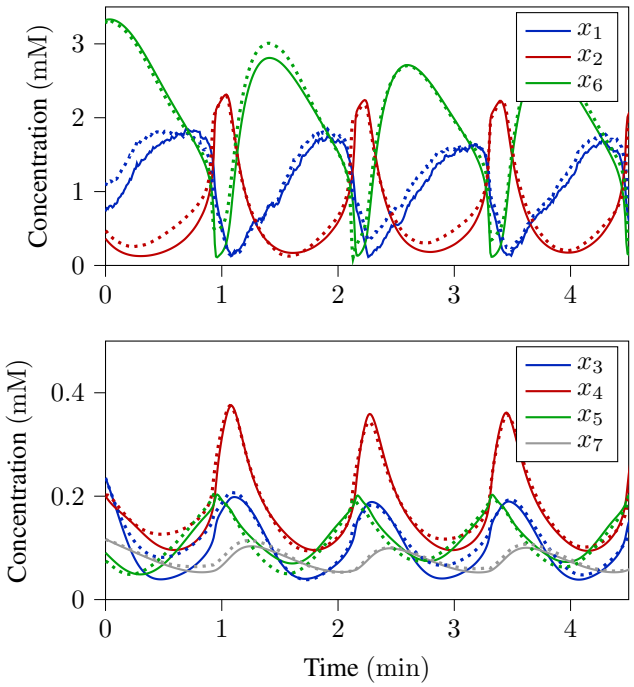

Fig. 4. Simulation results versus data from the nonlinear yeast glycolysis system from equation (5). Solid lines (−): nonlinear system; Dotted lines (· · ·): Koopman system estimation

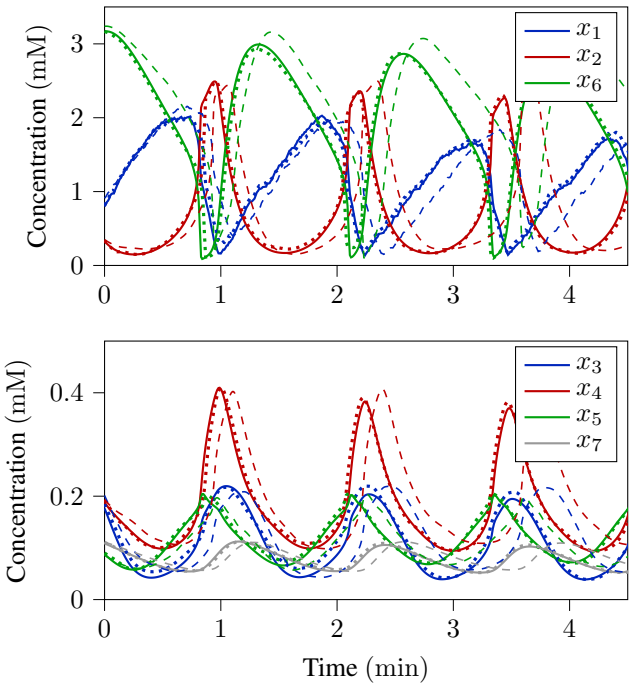

Fig. 5. Simulation results versus data from the nonlinear yeast glycolysis system from equation (5). Solid lines (−): nonlinear system; Dotted lines (· · ·): Saturated Koopman system estimation. Dashed lines (−−): Koopman system estimation without considered saturation

5.2 *Yeast Glycolysis System*

To go one step further we look at a yeast glycolysis system. The standard model from Daniels and Nemenman (2015) with Michaelis-Menten dynamics will be used. It is given by

$$\dot{\underline{x}} = \begin{bmatrix} -v_1 \\ 2v_1 - v_2 - v_3 \\ v_2 - v_4 \\ v_4 - v_5 - v_6 \\ v_2 - v_5 - v_3 \\ -2v_1 + 2v_4 - k_5 x_6 \\ \mu v_6 - k x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u \tag{5}$$

with

$$\begin{aligned} v_1 &= \frac{k_1 x_1 x_6}{1 + \left(\frac{x_6}{K_1}\right)^q} \\ v_2 &= k_2 x_2 (N - x_5) \\ v_3 &= k_6 x_2 x_5 \\ v_4 &= k_3 x_3 (A - x_6) \\ v_5 &= k_4 x_4 x_5 \\ v_6 &= \kappa (x_4 - x_7). \end{aligned}$$

The parameters used in this paper are given in table B.1. It should be mentioned that the v_i's are not Koopman states but variables to simplify the above equation. The excitation u is the input flux of glycolysis through the cell membrane. For identification purposes we excite the system with a normally distributed signal with $2.5\,\mathrm{mM/min}$ mean and a variance of $4\,\mathrm{mM^2/min^2}$. The excitation-signal u is shown in Fig. 6. We start the simulation with an initial state of $\underline{x}_0 = [\,1.5,\ 1.5,\ 0.15,\ 0.2,\ 0.2,\ 1.5,\ 0.075\,]\ \mathrm{mM}$ which is motivated by the data given in Daniels and Nemenman (2015) and also used in Yeung et al. (2019) for comparability.

The used ANN consists of two layers of 14 and 21 nodes in the encoder. Even though this special network is not deep we would use deep networks rather than wide networks, due to Eldan and Shamir (2017). The decoder has the same structure but mirrored, meaning the first decoder layer has 21 nodes and so forth. Furthermore, we add 7 observables to the 7 original states. A system with 7 additional states (14 in total) resulted in the best fit in a range from 1 to 10 additional states. No excitation-encoder is used.

The system is sampled for $100\,\mathrm{min}$ at a sampling rate of $0.01\,\mathrm{min}$ which results in a data set of 10,000 samples. The training of the ANN is done with the Adam algorithm at a learning rate of 0.1 over 2000 epochs but is interrupted if the loss function (MSE of the output) does not improve by 10^{-4} over 100 training steps. Since an increased number of batches resulted in a vast increase in the loss only one batch with the whole data set is used to train the network. Even though this made the whole algorithm sensitive to the initial weighting matrices, the results are much better after a certain amount of estimated systems. As activation functions the hyperbolic tangent is used due to previous tests that resulted in a better fit compared to other activation functions.

The one step predictions of the best estimation are shown in table 1. Different runs deliver a good ANN fit as well but no good Koopman layer fits. This seems to be the case due to a poor choice of the initial weights. One can see that all the data sets can be reconstructed almost perfectly from both the whole ANN and the linear layer. But this is also the case for a DKLc model with the input signal as an additional state. The

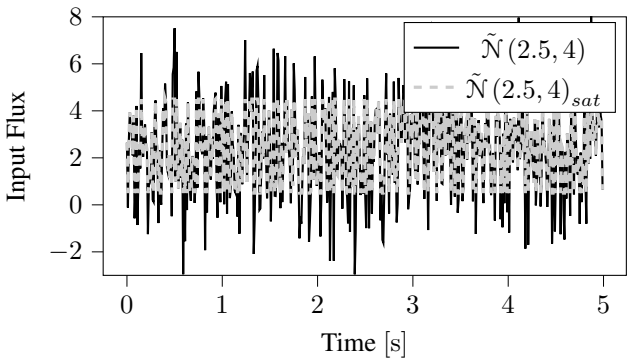

Fig. 6. One realization of an excitation-signal for system (5) with and without saturation between 0.5 and 4.5

prediction of the input signal is obviously neglected in this consideration.

In the next step to validate the model's true prediction performance we give the model only one initial state and look at the first 500 time steps of the prediction. As shown in Fig. 4 the prediction with the whole network does reproduce the systems behavior quite accurately with a fit of 89% with a test data set that is not used to train the ANN. The first 50 time steps are cut off due to settling behavior that is not estimated.

Table 1. One step prediction fits of the estimated system (5)

Model	Training	Validation	Test
Network	$> 99\%$	$> 99\%$	98%
Koopman Layer	95%	94%	95%

A model with the input as an additional state could not reproduce the systems dynamics over only a few time steps. Moreover, such a system is not usable prospectively due to the concepts mentioned in the introduction and shown in Fig. 1.

If we do the prediction with the Koopman layer only it is not as accurate anymore neither with training nor with validation or test data. This might be the case due to neglected Koopman prediction loss in this paper. Editing the loss function by adding explicit Koopman layer prediction capability holds huge potential as seen in Lusch et al. (2018).

To this moment we only analyzed systems with an affine input term. Since we defined our base model (3) with a nonlinear excitation term, we look at a more complex example. In this example all the conditions are equal to the above examined system, but the input is saturated between 0.5 and 4.5. One realization of a saturated input signal is shown in Fig. 6. Fig. 5 shows the network's prediction with considered saturation by using an additional input observable approximated by an input encoder with two layers and two nodes each. This results in a 92% fit. An estimation with equal initial weightings without an input signal encoder cannot reproduce the systems behavior as accurate as seen in the same plot as thin dashed lines. A clear drift can be seen in the data set which worsens with time. The same effect was observed in Yeung et al. (2019) with an autonomous version of the system.

Additionally, one can see, that the fit of the saturated system is even better than the system with an affine input term without a saturated input-signal. This might be due to the more complex model. Furthermore, with an input encoder the network might be more suitable to fit to a Koopman operator invariant subspace, since this subspace may also consist of observables that are functions in $\underline{u}$ as well, Proctor et al. (2018). Hence, it is no necessary condition that there are no additional observables in u even if the underlying nonlinear system is excited by an affine input term. Therefore, an ANN with an additional excitation-encoder may approximate a better fitting subspace rather than the true nonlinearity at the input. In any of both cases, this example shows, that the presented algorithm is capable of estimating systems with a nonlinearity at the input.

6. CONCLUSION

In this paper we showed a possible way to estimate an invariant subspace of the Koopman operator of a nonlinear system using artificial neural networks. With our method it is not only possible to hold onto the true states without a need to estimate them by a neural network. One may also choose certain observables previously and add additional states with the linear encoder.

To show the performance of the method we demonstrated the algorithm using a single pendulum and the yeast glycolysis pathway system which is a standard model for estimating metabolism in biology. It is possible to find a partly linear state-space system that represents the behavior of the system for almost the entire state vector within the chosen region of the state-space. Yet it was not tested, if the model can hold its validity if we examine an excitation that is different in its amplitude. This might result in a different point of operation of the state-space and thus to a loss of validity of the identified model.

During the work with this methodology several questions arose that must be addressed in the future. First, we mentioned an excitation-decoder that is necessary to finally compute controllers based in the Koopman operator subspace. Furthermore, we observed that many models estimated resulted in a poor multi step prediction using the estimated state-space whereas the one step fit is always almost 100%. This, on the one hand, might be due to initial weightings or on the other hand an inappropriately chosen loss. Thus, the method needs to be extended so that multiple time step are predicted during the training in both the Koopman layer and the whole ANN. For example, the loss used by Lusch et al. (2018) seems promising. Alternatively, one might introduce an iterative process to the method as seen in Li et al. (2017). Moreover, a fully linear prediction could not be achieved, meaning that the state vector must be predicted with the full network.

We have to point out, that the continuous spectrum of the pendulum can only be represented in its entirety by a really high order linear system or a parameterized system as seen in Lusch et al. (2018). At this stage, our method has no option to adapt to continuous spectra except to increase the model's order. It is a topic for further investigations to adopt the possibility of a parameterized Koopman operator for systems with a control input.

REFERENCES

Åström, K.J. and Bohlin, T. (1965). Numerical identification of linear dynamic systems from normal operating records. *IFAC Proceedings Volumes*, 2(2), 96–111. doi:10.1016/S1474-6670(17)69024-4.

Brunton, S.L., Brunton, B.W., Proctor, J.L., and Kutz, J.N. (2016). Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PloS one*, 11(2), e0150171. doi:10.1371/journal.pone.0150171.

Daniels, B.C. and Nemenman, I. (2015). Efficient inference of parsimonious phenomenological models of cellular dynamics using s-systems and alternating regression. *PloS one*, 10(3), e0119821. doi:10.1371/journal.pone.0119821.

Eldan, R. and Shamir, O. (2017). The power of depth for feedforward neural networks. URL http://arxiv.org/pdf/1512.03965v4.

Erichson, N.B., Muehlebach, M., and Mahoney, M.W. (2019). Physics-informed autoencoders for lyapunov-stable fluid flow prediction. URL https://arxiv.org/pdf/1905.10866.

Kaiser, E., Kutz, J.N., and Brunton, S.L. (2018). Data-driven discovery of koopman eigenfunctions for control. URL http://arxiv.org/pdf/1707.01146v2.

Koopman, B.O. (1931). Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, (5), 315–318.

Kutz, J.N., Proctor, J.L., and Brunton, S.L. (2018). Applied koopman theory for partial differential equations and data-driven modeling of spatio-temporal systems. *Complexity*, 2018, 1–16. doi:10.1155/2018/6010634.

Li, Q., Dietrich, F., Bollt, E.M., and Kevrekidis, I.G. (2017). Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the koopman operator. *Chaos (Woodbury, N.Y.)*, 27(10), 103111. doi:10.1063/1.4993854.

Li, Y., He, H., Wu, J., Katabi, D., and Torralba, A. (2019). Learning compositional koopman operators for model-based control. URL http://arxiv.org/pdf/1910.08264v1.

Ljung, L. (1999). *System identification: Theory for the user*. Prentice-Hall, Upper Saddle River, NJ.

Lusch, B., Kutz, J.N., and Brunton, S.L. (2018). Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1), 4950. doi:10.1038/s41467-018-07210-0.

Mangan, N.M., Kutz, J.N., Brunton, S.L., and Proctor, J.L. (2017). Model selection for dynamical systems via sparse regression and information criteria. *Proceedings. Mathematical, physical, and engineering sciences*, 473(2204), 20170009. doi:10.1098/rspa.2017.0009.

Mardt, A., Pasquali, L., Wu, H., and Noé, F. (2018). Vampnets for deep learning of molecular kinetics. *Nature communications*, 9(1), 5. doi:10.1038/s41467-017-02388-1.

Mezić, I. (2005). Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1-3), 309–325. doi:10.1007/s11071-005-2824-x.

Otto, S.E. and Rowley, C.W. (2017). Linearly-recurrent autoencoder networks for learning dynamics. URL http://arxiv.org/pdf/1712.01378v2.

Proctor, J.L., Brunton, S.L., and Kutz, J.N. (2018). Generalizing koopman theory to allow for inputs and control. *SIAM Journal on Applied Dynamical Systems*, (1), 909–930.

Takeishi, N., Kawahara, Y., and Yairi, T. (2017). Learning koopman invariant subspaces for dynamic mode decomposition. URL http://arxiv.org/pdf/1710.04340v2.

Wu, H. and Noé, F. (2020). Variational approach for learning markov processes from time series data. *Journal of Nonlinear Science*, 30(1), 23–66. doi:10.1007/s00332-019-09567-y.

Yeung, E., Kundu, S., and Hodas, N. (2019). Learning deep neural network representation for koopman operators of nonlinear dynamical systems. *2019 American Control Conference (ACC)*, 4832–4839.

Appendix A. SEPARATION OF INPUT AND OUTPUT SPACES

As given in Proctor et al. (2018) the linear state-equation in the Koopman space can be written as

$$\underline{\mathrm{Kg}}\left(\underline{\mathrm{x}},\underline{\mathrm{u}}\right) \approx \sum_{j=1}^{\infty} \sigma_j \psi_j\left(\underline{\mathrm{x}}\right) \underline{\mathrm{q}}_j$$

assuming that there are eigenfunctions that are solely function of the state $\underline{\mathrm{x}}$. Thus, the excitation-signal $\underline{\mathrm{u}}$ is not evolving dynamically with the Koopman operator. Furthermore, the eigenfunctions $\psi_j\left(\underline{\mathrm{x}}\right) : \mathbb{R}^n \to \mathbb{R}$ span the infinite-dimensional subspace $\mathcal{H}_x$ of another infinite-dimensional Hilbert-Space $\mathcal{H}$ where the Koopman operator is acting on. In our paper we assume that there are only observables in $\underline{\mathrm{x}}$ and $\underline{\mathrm{u}}$ but no mixed ones ($\mathcal{H}_{xu} = \emptyset$). With the above equation we set the first n_e observables (state observables) to

$$\underline{\mathrm{g}}\left(\underline{\mathrm{x}}\right) = \sum_{j=1}^{\infty} \underline{\tilde{\mathrm{g}}}_j \psi_j\left(\underline{\mathrm{x}}\right).$$

The remaining excitation-observables $\underline{\mathrm{g}}\left(\underline{\mathrm{u}}\right) \in \mathcal{H}_u$ are arbitrary functions that are estimated with the artificial neural network shown in Fig. 2. The constant vectors $\underline{\tilde{\mathrm{g}}}_j \in \mathbb{R}^{n_e}$ allocate the different eigenfunctions to certain observables.

Appendix B. SYSTEM PARAMETERS

The used parameters in equation (5) are given in the table below and are taken from Daniels and Nemenman (2015).

Table B.1. Parameters for system (5)

Parameter	Value	Parameter	Value	Parameter	Value
k_1	100	k_2	6	k_3	16
k_4	100	k_5	1.28	k_6	12
k	1.8	κ	13	K_1	0.52
μ	0.1	q	4	N	1
A	4				