Data-Driven Nonlinear System Identification of a Throttle Valve Using Koopman Representation

Nicolas Bongiovanni, Bojan Mavkov, Renato Martins, Guillaume Allibert

Data-Driven Nonlinear System Identification of a Throttle Valve Using Koopman Representation

Nicolas Bongiovanni[1] Bojan Mavkov[1] Renato Martins[2, 3] Guillaume Allibert[1]

***Abstract*—Electrical Throttle Bodies (ETBs) are massively used in the automotive industry and their modeling and control are challenging because of their high nonlinearity and stochasticity. In this paper, we present a data-driven method grounded on the Koopman operator for the identification of a real ETB valve. The model obtained is control-oriented and represented in a quasi-linear-parameter varying framework. Different experiments are performed to evaluate the performance of the proposed method, including the comparison with two classical nonlinear system identification methods.**

***Index Terms*—System identification, Koopman operator theory, neural networks, Electronic Throttle Body**

I. Introduction

Nonlinearity is a central challenge in all the fields related to dynamical systems. A plethora of methods have been developed to nonlinear system control [1], but there is not a unique systematic approach in contrast with the Linear Time-Invariant (LTI) case. One way to use LTI control theory for nonlinear systems is through linearization. However, this approach has many limitations: it requires staying in the vicinity of an equilibrium point and it can be too limited to capture higher-order non-linearities of the dynamics. More versatile and compact ways to capture the complexities of nonlinear systems is through bilinear [2], Linear Parameter-Varying (LPV) [3], [4], and quasi-LPV model representations [4], for which various tools have been developed.

Koopman operator theory [5] is a reformulation of classical dynamical system theory with the goal of finding approximately linear representations of dynamical systems, by lifting the state to an infinite dimensional space through so-called observable functions. This theory was first introduced in 1931 [5], and has recently been the subject of intense research since the introduction of data-driven methods. The introduction of Dynamic Mode Decomposition (DMD) [6] and Extended DMD [7] improved the calculation of the finite-dimensional approximation of the Koopman operator. Moreover, new methods based on Neural Networks (NNs) were introduced to facilitate the approximation of the lifting functions [8]–[10]. These representations were mostly used in an LTI framework, but also in a quasi-LPV framework [8], in order to address systems with continuous spectrum.

[1]Université Côte d'Azur, CNRS, I3S, France
Email: nicolas.bongiovanni@etu.univ-cotedazur.fr,
{bojan.mavkov, guillaume.allibert}@univ-cotedazur.fr
[2]Université de Bourgogne, ICB UMR 6303 CNRS, France
[3]Université de Lorraine, CNRS, Inria, LORIA, France
Email: renato.martins@u-bourgogne.fr

The extension of the Koopman approach to actuated systems has proven to be difficult because the actuation signals change the spectral properties of the Koopman operator [11]. In [12] an approximation of Koopman representation is introduced in a bilinear-control form for control-affine systems, named Koopman Canonical Transform (KCT). This work was further extended in [11], [13], [14]. In [15] LPV Koopman representation for nonlinear systems in introduced, and [16]–[18] introduce quasi-LPV Koopman approximations for control-affine system. The approach developed in this work is closely related to and inspired by the method presented in [17], however, we design distinct loss functions aiming to enhance the estimation of the studied system.

In this work, a learning-based identification method is applied to an Electrical Throttle Body (ETB). This type of valve is used in the industry for flow control, typically in the automotive, chemical, pharmaceutical, and food industries. Performing system identification and control on these devices is challenging because of static friction effects and non-linearities of the gearbox, and the return spring. This results in particularly complex dynamics usually modeled with an asymmetric hysteresis [19]. Furthermore, even if dynamical models have been developed, each valve has different parameters [20], [21]. Finally, ETBs are known for their stochasticity. This leads us to address the challenge of applying NNs-based Koopman theory to represent the nonlinear dynamics of an ETB. An ETB can be modeled as a control-affine system [22], and thus, the state-of-the-art methods for Koopman representation can be directly applied to it. The resulting quasi-LPV model is relevant because it can be adapted for a control design [23], [24]. To the best of our knowledge, this is the first time that Koopman theory has been used to perform system identification of this kind of experimental dynamical system.

The remaining of this paper is structured as follows. The identification problem and core concepts of Koopman theory for control systems are recalled in Section II. Section III presents the proposed Koopman-based identification algorithm. Finally, the experiments and results are presented in Section IV. Section V concludes the paper and discusses possible future directions.

II. Problem setting

Before presenting the proposed learning-based system representation and control method, we introduce core notions of nonlinear systems and of the Koopman operator. Our modeling

considers discrete-time nonlinear dynamical systems of the form:

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

$$\boldsymbol{y_k} = h(\boldsymbol{x_k}), \tag{1b}$$

where $\boldsymbol{x_k} \in \mathbb{R}^{n_x}$ denotes the state vector, $\boldsymbol{y_k} \in \mathbb{R}^{n_y}$ is the output vector of the system, $\boldsymbol{u_k} \in \mathbb{R}^{n_u}$ is the input vector, and k denotes the time step, $\boldsymbol{x_k} = \boldsymbol{x(t_k)} = \boldsymbol{x(k\Delta t)}$, where Δt is the sampling time. f and h are unknown nonlinear mapping functions, where f is assumed to be continuously differentiable. In order to represent the unknown dynamics of the system, a training dataset $\mathcal{D}$ is used, which consists of N measurement samples of the inputs $\{u(t_0), u(t_1), \ldots, u(t_N)\}$ and the outputs of the system $\{y(t_0), y(t_1), \ldots, y(t_N)\}$. The dataset is collected at time instants $T = \{t_0 = 0, t_1, \ldots, t_N\}$ from experiments performed on the system.

A. Background on Koopman Theory

The Koopman operator provides an alternative representation that describes the evolution of nonlinear dynamical systems through a globally linear representation. The goal of this work is to develop an algorithm that estimates (1) using the dataset $\mathcal{D}$ in Koopman form. We start introducing the Koopman modeling for autonomous dynamical systems. The nonlinear dynamical system in eq. (1a) becomes:

$$\boldsymbol{x_k} = f(\boldsymbol{x_k}), \tag{2}$$

where $f : \mathbb{R}^{n_x} \longrightarrow \mathbb{R}^{n_x}$. The Koopman operator $\mathcal{K} : \mathcal{F} \to \mathcal{F}$ associated with the nonlinear map f is then defined by:

$$(\mathcal{K}\psi)(\boldsymbol{x_k}) = \psi \circ f(\boldsymbol{x_k}) = \psi(\boldsymbol{x_{k+1}}),\ \forall \psi \in \mathcal{F}, \tag{3}$$

where $\psi : \mathbb{R}^n \to \mathbb{R}$ are called observables (functions) and $\mathcal{F}$ is a Banach function space of the observables functions. Although the linear Koopman operator is infinite-dimensional, in practice, finite approximations must be found. In this case, the existence of a finite-dimensional Koopman invariant subspace $\mathcal{F}_{n_\mathrm{f}} \subseteq \mathcal{F}$ is assumed (n_f being the dimension of $\mathcal{F}_{n_\mathrm{f}}$) as discussed in [25].

B. Extension to Actuated Systems

There are several ways of generalizing the Koopman operator to controlled systems [15], [17], [26]–[28]. Here, the approach of [28] is exposed. Consider the general representation of the nonlinear system, whose actuation effects are isolated in g:

$$\boldsymbol{x_{k+1}} = f\left(\boldsymbol{x_k}, \boldsymbol{u_k}\right) = f\left(\boldsymbol{x_k}, 0\right) + g\left(\boldsymbol{x_k}, \boldsymbol{u_k}\right), \tag{4}$$

with $\boldsymbol{u_k} \in \mathbb{U} \subseteq \mathbb{R}^{n_\mathrm{u}}$, $\boldsymbol{x_0} \in \mathbb{X}$ and $f : \mathbb{R}^{n_\mathrm{x}} \times \mathbb{U} \to \mathbb{R}^{n_\mathrm{x}}$. It is considered that $\mathbb{U}$ is given such that $\mathbb{X}$ is compact and forward invariant under f.

[28] demonstrates the existence of a finite-dimensional Koopman representation of (4) in the following form:

$$\Phi\left(\boldsymbol{x_{k+1}}\right) = A\Phi\left(\boldsymbol{x_k}\right) + B\left(\boldsymbol{x_k}, \boldsymbol{u_k}\right)\boldsymbol{u_k}, \tag{5a}$$

with $\boldsymbol{A} \in \mathbb{R}^{n_\mathrm{f} \times n_\mathrm{f}}$ and

$$B\left(\boldsymbol{x_k}, \boldsymbol{u_k}\right) = \int_0^1 \frac{\partial \mathcal{B}}{\partial u}\left(\boldsymbol{x_k}, \lambda \boldsymbol{u_k}\right) \mathrm{d}\lambda, \tag{5b}$$

where $\mathcal{B}\left(\boldsymbol{x_k}, \boldsymbol{u_k}\right) =$

$$\left(\int_0^1 \frac{\partial \Phi}{\partial x}\left(f\left(\boldsymbol{x_k}, 0\right) + \lambda g\left(\boldsymbol{x_k}, \boldsymbol{u_k}\right)\right) \mathrm{d}\lambda\right) g\left(\boldsymbol{x_k}, \boldsymbol{u_k}\right). \tag{5c}$$

The conditions identified by [28] are the following: Φ is of class $\mathcal{C}^1$ such that $\Phi(f(\cdot, 0)) \in \mathrm{span}\{\Phi\}$ with $\Phi : \mathbb{X} \to \mathbb{R}^{n_\mathrm{f}}$ and $\mathbb{X}$ convex. Finally, Eq. 5a can be rewritten in a quasi-LPV way, as follows:

$$\boldsymbol{z_k} = \Phi\left(\boldsymbol{x_k}\right), \tag{6a}$$

$$\boldsymbol{z_{k+1}} = \boldsymbol{A}\boldsymbol{z_k} + B_\mathrm{z}\left(p_k\right)\boldsymbol{u_k}, \tag{6b}$$

where the scheduling map $p_k = \mu\left(\boldsymbol{z_k}, \boldsymbol{u_k}\right)$ is introduced such that $B_\mathrm{z} \circ \mu = \boldsymbol{B}$. μ, z_k and u_k are assumed to be bounded.

C. Control-affine Systems with Unavailable State

An important class of systems can be represented in a control-affine form. In this case, one can remove the dependency of the scheduling map to $\boldsymbol{u_k}$. In the present work, the state $\boldsymbol{x_k}$ is not directly available, and thus, $\boldsymbol{y_k}$ and $\boldsymbol{u_k}$ are used to estimate the lifted state. As a result, the following formulation can be adopted:

$$\boldsymbol{z_k} = \phi_e(\boldsymbol{y_k}, \boldsymbol{u_k}), \tag{7a}$$

$$\boldsymbol{z_{k+1}} = \boldsymbol{A}\boldsymbol{z_k} + B_f(\boldsymbol{z_k})\boldsymbol{u_k}, \tag{7b}$$

where $k \in \mathbb{N}$ and $n_z = n_f$. This representation can represent an ETB valve, because it can be modeled as a control-affine system [22].

III. Koopman Representation Learning

In this section, we present the data-driven algorithm developed to identify and control the system (7) from data using Koopman representation. We highlight that the structure of ϕ_e in eq. (7) is unknown, and it should be approximated directly from data. Our approach leverages a data-driven strategy to perform this estimation, specifically utilizing NNs to model these functions through autoencoders. In this context, $\boldsymbol{A}$ is an unknown matrix to be identified and $B_f(\boldsymbol{z_k})$ is similarly modeled with a NN. Finding $\hat{\boldsymbol{y}}_\boldsymbol{k}$ from $\boldsymbol{z_k}$ is crucial in the context of control. In this work, we evaluate two state reconstruction options. The first one is through a nonlinear decoder modeling ϕ_d:

$$\hat{\boldsymbol{y}}_\boldsymbol{k}^{\boldsymbol{nlin}} = \phi_d(\boldsymbol{z_k}). \tag{8}$$

The second alternative representation is a linear approximation of the output defined as

$$\hat{\boldsymbol{y}}_\boldsymbol{k}^{\boldsymbol{lin}} = \boldsymbol{C}\boldsymbol{z_k}, \tag{9}$$

where $\boldsymbol{C} \in \mathbb{R}^{n_\mathrm{y} \times n_\mathrm{z}}$ represents the linear mapping. Although more generic, the former nonlinear decoder representation is less practical to be used with most classical control methods [24], [29]. Once these mappings ϕ_e, ϕ_d, $\boldsymbol{C}$, $\boldsymbol{A}$, B_f are learned, the resulting quasi-LPV model (7) with (8) or (9) is then suitable for control as shown in our experiments.

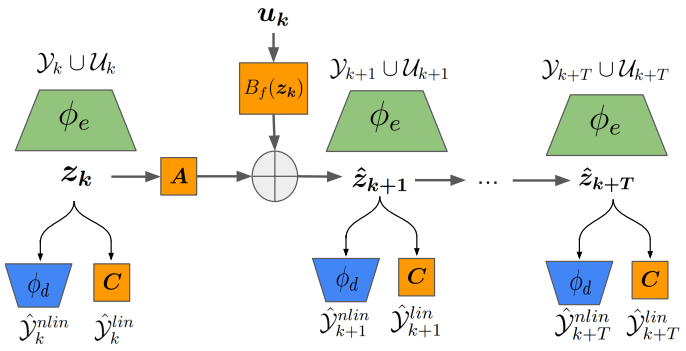


Fig. 1: Schematic illustrating the Koopman operator for nonlinear systems by using learning techniques.

A. Koopman State Reconstruction Losses

Our approach leverages autoencoders to approximate jointly ϕ_e, ϕ_d, and the dynamics in the latent space. The designed autoencoder has nonlinear activation functions to approximate ϕ_e. The inputs are a function of the last n_b inputs and n_a sensor values ($\mathcal{Y}_k \cup \mathcal{U}_k$), in order to identify the lifted states. Similarly, an MLP is used to model ϕ_d in order to reconstruct the outputs $\hat{\boldsymbol{y}}_k$. When computing the linear reconstruction of $\hat{\boldsymbol{y}}_k$, a matrix $\boldsymbol{C}$ is used instead of ϕ_d.

The learning of the network models is done with a carefully designed multi-objective loss function with several terms to weigh the current state reconstruction and future states of the dynamic system. For instance, the loss L_{lin} is the prediction error in the observable space, and L_{pred} is the prediction error in the state space. These two losses are designed to enforce the linearity of the estimated system in the observable space. We also consider three regularization losses to prevent overfitting, where L_{reg1} and L_{reg3} are the l_2 norm of the weight of the two MLPs, and L_{reg2} is the l_2 norm of $\hat{\boldsymbol{A}}$. The full minimization problem is thus formulated as:

$$\min_{\mathcal{W}_\phi, \mathcal{W}_B, \hat{\boldsymbol{A}}} L(\mathcal{W}_\phi, \mathcal{W}_B, \hat{\boldsymbol{A}}), \tag{10a}$$

where:

$$\begin{aligned} L = \alpha_1 \overbrace{\frac{1}{b_s(n_s-1)} \sum_{k=1}^{b_s} \sum_{k=2}^{n_s} \|\hat{\boldsymbol{z}}_{\boldsymbol{k}} - \phi_e(\boldsymbol{y_k}, \boldsymbol{u_k})\|_2}^{L_{lin}} \\ + \alpha_2 \overbrace{\frac{1}{b_s(n_s-1)} \sum_{k=1}^{b_s} \sum_{k=2}^{n_s} \|\hat{\boldsymbol{y}}_{\boldsymbol{k}} - \boldsymbol{y_k}\|_2}^{L_{pred}} \\ + \alpha_3 \overbrace{\|\mathcal{W}_\phi\|_2^2}^{L_{reg_1}} + \alpha_4 \overbrace{\|\hat{\boldsymbol{A}}\|_2^2}^{L_{reg_2}} + \alpha_5 \overbrace{\|\mathcal{W}_B\|_2^2}^{L_{reg_3}} \end{aligned} \tag{10b}$$

with $\mathcal{W}_\phi$ the weights and biases of the Multi-Layer Perception (MLP) networks that approximate ϕ_e and ϕ_d respectively while $\mathcal{W}_B$ represents the weights of the MLP that approximates $\hat{B}_f$. n_e is the number of epochs, n_s is the number of steps in each trajectory, b_s is the batch size. To deal with the multi-objective optimization problem (10), a linear scalarization is used where weights α_1 to α_5 are used to deal with the different scales of the individual loss functions.

Algorithm 1

Inputs: training dataset $\mathcal{D}$, weights (α_1-α_5), number of epochs (n_e), batch size (bs), dimension of the observable space (n_z), learning rate (η)

Initialize $\hat{\boldsymbol{A}}$ as the identity and the parameters $\mathcal{W}_B$ and $\mathcal{W}_\phi$
for epoch in 1 to n_e **do**
  for batch in $\mathcal{D}$ **do**
    $L_{pred} = 0$, $L_{lin} = 0$
    for $(\boldsymbol{y_1}, \boldsymbol{u_1}, ..., \boldsymbol{y_{n_s}}, \boldsymbol{u_{n_s}})$ in batch **do**
      $\hat{\boldsymbol{z}}_{\boldsymbol{1}} = \phi_e(\boldsymbol{y_k}, ..., \boldsymbol{y_{k-n_a}}, \boldsymbol{u_k}, ..., \boldsymbol{u_{k-n_b}})$
      $\hat{\boldsymbol{y}}_{\boldsymbol{1}} = \phi_d(\hat{\boldsymbol{z}}_{\boldsymbol{1}})$
      for k in 1 to $n_s - 1$ **do**
        $\hat{\boldsymbol{z}}_{\boldsymbol{k+1}} = \hat{\boldsymbol{A}}\hat{\boldsymbol{z}}_{\boldsymbol{k}} + \hat{B}_f(\hat{z}_k)\boldsymbol{u_k}$
        $\hat{\boldsymbol{y}}_{\boldsymbol{k+1}} = \phi_d(\hat{\boldsymbol{z}}_{\boldsymbol{k+1}})$
        $L_{pred} \mathrel{+}= \sum_{k=2}^{n_s} \|\hat{\boldsymbol{y}}_{\boldsymbol{k}} - \boldsymbol{y_k}\|_2$
        $L_{lin} \mathrel{+}= \sum_{k=2}^{n_s} \|\hat{\boldsymbol{z}}_{\boldsymbol{k}} - \phi_e(\boldsymbol{y_k}, \boldsymbol{u_k})\|_2$
      end for
    end for
    $L_{pred} = \frac{1}{n_s - 1} L_{pred}$
    $L_{lin} = \frac{1}{n_s - 1} L_{lin}$
    $L = \alpha_1 L_{pred} + \alpha_2 L_{lin} + \alpha_3 \|\mathcal{W}_\phi\|_2^2 + \alpha_4 \|\hat{\boldsymbol{A}}\|_2^2 + \alpha_5 \|\mathcal{W}_B\|_2^2$
    update $\hat{\boldsymbol{A}}, \mathcal{W}_B, \mathcal{W}_\phi$ by backpropagation
  end for
end for

Outputs : $\hat{\boldsymbol{A}}, \mathcal{W}_B, \mathcal{W}_\phi$

The complete implementation is described in Algorithm 1, and a scheme of the training network architecture is shown in Fig. 1, where: $\mathcal{U}_k = [\boldsymbol{u_k}, \boldsymbol{u_{k-1}}, ..., \boldsymbol{u_{k-n_b}}]$, $\mathcal{Y}_k = [\boldsymbol{y_k}, \boldsymbol{y_{k-1}}, ..., \boldsymbol{y_{k-n_a}}]$, $\hat{\mathcal{Y}}_k^{nlin} = [\hat{\boldsymbol{y}}_{\boldsymbol{k}}^{nlin}, \hat{\boldsymbol{y}}_{\boldsymbol{k-1}}^{nlin}, ..., \hat{\boldsymbol{y}}_{\boldsymbol{k-n_a}}^{nlin}]$, $\hat{\mathcal{Y}}_k^{lin} = [\hat{\boldsymbol{y}}_{\boldsymbol{k}}^{lin}, \hat{\boldsymbol{y}}_{\boldsymbol{k-1}}^{lin}, ..., \hat{\boldsymbol{y}}_{\boldsymbol{k-n_a}}^{lin}]$. Here n_a and n_b are respectively the numbers of the last outputs and inputs taken into account. The loss L is computed on sections of trajectories of a number of steps n_s to focus on short-term prediction and to allow for simpler parallelization with modern batch optimization algorithms [30].

IV. Experimental Results

A. Data Acquisition

The data for the system representation learning were collected using multiple open-loop experiments of a real valve platform. The identification data was collected using inputs with different forms. The first batch of inputs was formed by 45% of Pseudo-random binary sequence (PRBS) [31]. PRBS are characterized by their randomness and binary nature, which enable an efficient and thorough exploration of a system's response across its operational ranges [31]. In the present work, PRBS was generated based on the algorithms presented

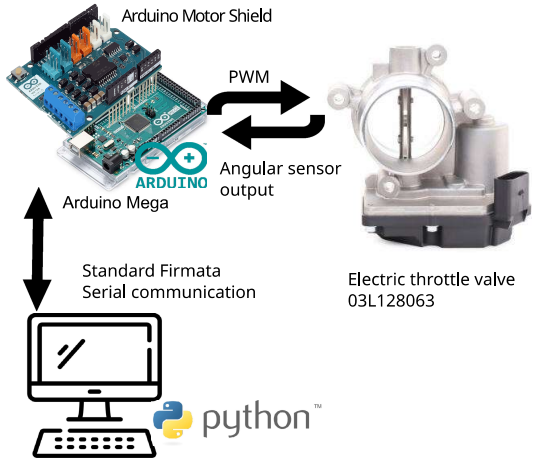
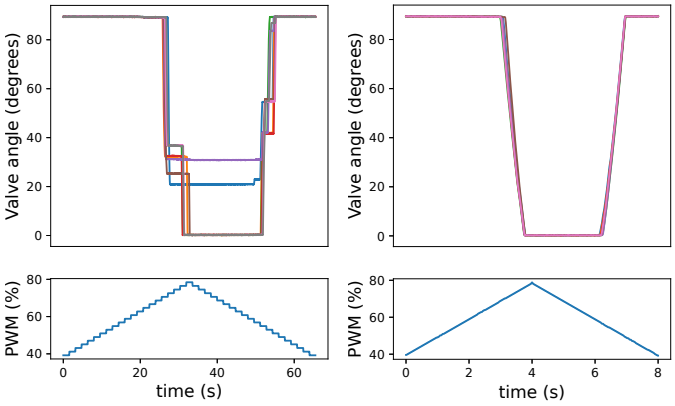


Fig. 2: Test bench and considered ETB valve system for the data acquisition and hardware configuration of the ETB control system.



(a) (b)

Fig. 3: Visualizations of control inputs for a series of openings and closings of the valve. In (a), inputs are generated using a staircase signal, and (b) shows the response of the system to a ramp input signal.

in [20], [32], [33]. The other 45% are composed of square signals with random amplitudes and lengths (see Fig. 7). Square signals with random varying amplitudes were used to excite the operation mode where stochasticity is present. Finally, 10% of sinusoidal signals with random varying amplitude were added to the inputs.

B. Considered Nonlinear Valve System

We have selected a real Electrical Throttle Body (ETB) system to perform the experiments. The ETB is a type of valve used in the industry for flow control in the automotive, chemical, pharmaceutical, and food industries. The present work focuses on a model that regulates air inflow into the motor's combustion system of cars. It is crucial for the superimposed engine speed control system [34]. In response to a driver's pedal motion, it controls the throttle plate position which in turn controls the car speed [19]. To optimize the fuel economy and emissions, fast throttle positioning response and accurate reference tracking are required. The exact model studied here is "Throttle body 158T0024", produced by Ridex®. It has equipped some Audi®, Škoda®, Seat® and VW® models. Fig. 2 displays the test bench that was used in this work. In order to control the opening of the valve and to recover the data, an Arduino Mega 2560® in combination with a Rev3® Arduino motor shield was used in serial communication with a Dell Precision 3650® computer.

The input was generated by manipulating the duty ratio of the Pulse-Width Modulation (PWM) signal that is sent to the ETB by the Arduino board. The output was provided by the position sensor that measures the opening of the valve. Data was collected by exciting an ETB sampled at 20Hz for a total duration of thirty minutes. In order to present the nonlinear behaviors and the stochasticity, a series of experiments with identical inputs were performed. The results are presented in Fig. 3. In both figures, the asymmetric hysteresis is clearly visible. Stochasticity is observed when the input is slowly variated by using a staircase signal (Fig. 3 (b)) when different outputs are produced by repeating the identical inputs. Furthermore, the figure illustrates that only inputs falling within the PRBS range of $[40, 80](\%)$ trigger valve motion, and thus the input space was restricted in this range in further experiments.

C. Implementation, Metrics and Training Details

Using the data generated from the ETB, the model was trained using the proposed algorithm presented in Section III. In order to compare the different obtained models, the Normalised Root Mean Square Error (NRMSE) is used between the true and predicted system outputs:

$$\mathrm{NRMSE} = \frac{\mathrm{RMSE}}{\sigma_y} = \frac{\sqrt{\sum_{k=k_0}^{n_s} \|\hat{\boldsymbol{y}}_k - \boldsymbol{y_k}\|_2^2}}{\sqrt{\sum_{k=k_0}^{n_s} \|\hat{\boldsymbol{y}}_k - \bar{\boldsymbol{y}}\|_2^2}} \tag{11}$$

where σ_y and $\bar{y}$ are respectively the standard deviation and the mean of the measured outputs.

The networks and algorithm shown in algorithm 1 were implemented with PyTorch. The training hyperparameters are given in Table I. Optimizing the dimension of $\boldsymbol{z_k}$ is significant and, as shown in Fig. 4, the tuning outcomes indicate that selecting $n_z = 24$ gives the most favorable balance between model complexity and the quality of the fit. This tuning was performed using our model with linear output reconstruction because this strategy is more suitable for control. α_1 and α_2 were chosen by observing the evolution of the losses while training with a higher priority on L_{lin} during the first few epochs in order to find first a proper observable space. The model was trained using a GPU GeForce RTX 3060® and took three hours.

D. Results and Discussion

We present some qualitative results in Fig. 5. Here an efficient estimation is obtained by applying PRBS inputs in both cases, where nonlinear decoder and linear estimation were

TABLE I: Architecture details and training hyper-parameters.

Param.	Value	Param.	Value	Param.	Value
Scheduler	Exp LR	# layers	4	α_1	100
γ (scheduler)	0.995	layer width	60	α_2	1
Batch size	32	η	1e-3	α_3	1e-9
Act function	ReLu	n_z	24	α_4	0
Optimizer	Adam	n_a	3	α_5	0
# epochs	1000	n_b	3	n_s	130

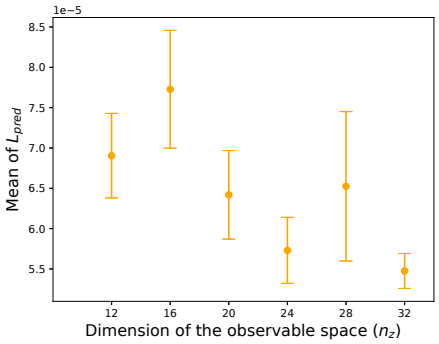


Fig. 4: For several dimensions n_z of $\boldsymbol{z_k}$, five pieces of training were performed with each a different weights initialization (determined by a manual seed). The points represent the mean of the prediction loss L_{pred} for each configuration, the bars represent their standard deviation.

performed for $n_z = 24$. The NRMSE scores of each method are indicated at the top of the figure. By zooming on Fig. 5, it is observed that the nonlinear decoder captures better the nonlinear behavior of the system in some regions where the system was brought to a steady state. In all the trainings, a $\boldsymbol{A}$ matrix with stable eigenvalues is obtained, which is coherent with the stability of the original data-generating system.

Subsequently, the performance of the proposed model is compared with some classical nonlinear system identification methods. Fig. 6 and 7 display estimations made by our model and by models identified using Matlab® R2022b Nonlinear System Identification Toolbox with a discrete Nonlinear ARX (NARX) and a discrete Neural State Space (NSS) models. Both were outperformed, although NSS was shown to be highly dependent on the tuning parameters. Concerning the hyper-parameters of the NSS, 4 layers of width 128 are used, the bias initializer is set to 0, the weights initializer is Glorot, and the optimizer is Adam. Concerning the NARX model, the number of delayed inputs is set to 4, the minimum input delay to 0, and the input-output delay to 1.

The results shown in Fig. 6 indicate that, for sequences in which the system is excited by PRBS inputs, our results are similar to the ones obtained with the NARX model, however, we recall that our model is more suitable for control. The obtained results are also better than the ones obtained with the NSS model, the latter faces difficulties in estimating the state in the data sequences where the mechanism of the ETB was at an angle of 90°. Finally, Fig. 7 contains sequences in which the behavior of our ETB is stochastic, in the sense that it is not always possible to predict if a particular input value will be high enough to close the mechanism in case it is opened

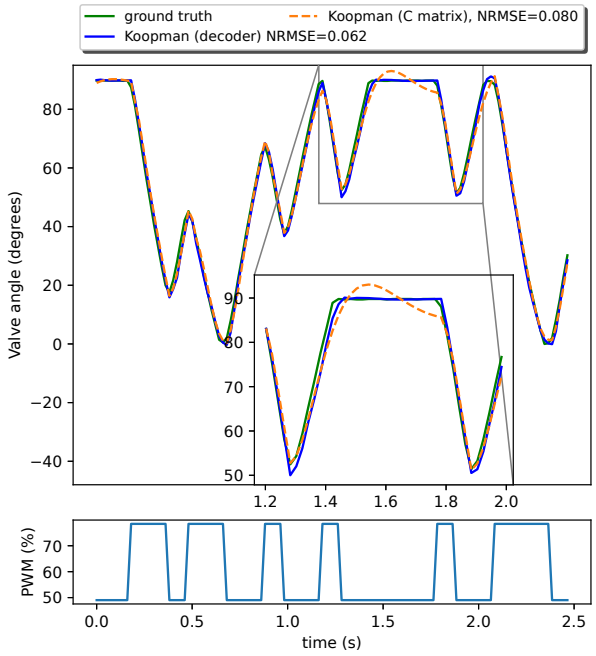


Fig. 5: Predictions of our model with PRBS inputs

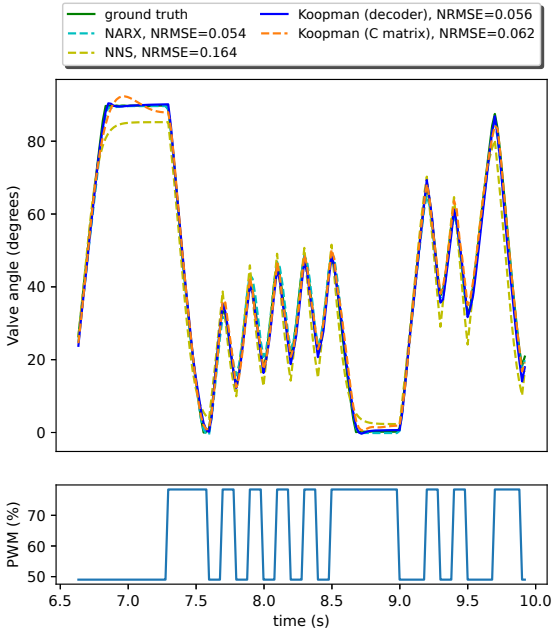


Fig. 6: Comparison of the two obtained models (and the two baseline competitors) with the state prediction of a trajectory excited by a PRBS signal.

(or if a particular input value will be low enough to open the mechanism in the case it is closed). With our particular model, this stochasticity is observed when the inputs are composed of sequences of values between 53% and 67% with changes at a frequency lower than 1Hz. In these cases, all the methods face significant difficulties in prediction, but our method still

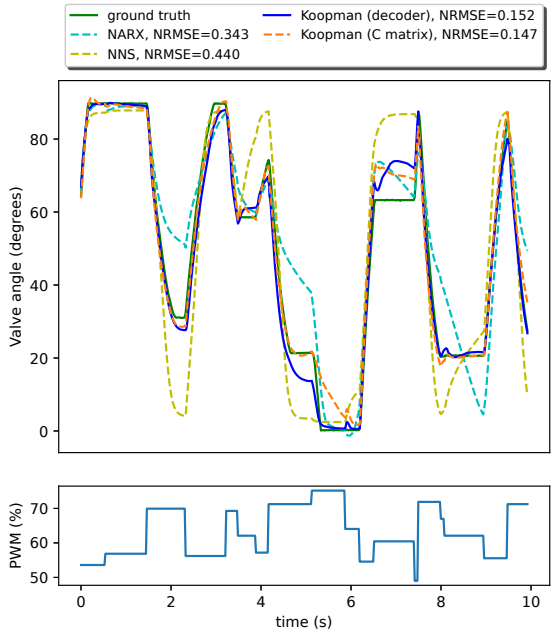


Fig. 7: Comparison of the two models (and two baseline competitors) obtained with the state prediction of a trajectory excited by a signal with random piece-wise constant inputs.

outperforms the two classical ones.

V. Conclusions

In this work, a system identification algorithm based on Koopman theory with NNs is presented. This algorithm is tailored to solve the challenge of identifying a model for the complex nonlinear system of an ETB valve. The two obtained Koopman models (a linear one and a non-linear reconstruction decoder) presented good performance, notably when compared with the other two existing nonlinear identification techniques. In future work, we plan to go further and evaluate the method to design and perform feedback control of the ETB valve.

Acknowledgments

This work was supported by the French National Research Agency (ANR) under contract ANR-23-CE23-0003-01.

References

[1] H. K. Khalil, *Nonlinear systems; 3rd ed.* Prentice-Hall, 2002.
[2] D. L. Elliott, *Bilinear control systems: matrices in action*. Springer, 2009, vol. 169.
[3] J. Mohammadpour and C. W. Scherer, *Control of linear parameter varying systems with applications*. Springer, 2012.
[4] C. Briat, "Linear parameter-varying and time-delay systems," *Analysis, observation, filtering & control*, vol. 3, pp. 5–7, 2014.
[5] B. O. Koopman, "Hamiltonian systems and transformation in Hilbert space," *Proceedings of the National Academy of Sciences*, vol. 17, 1931.
[6] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of fluid mechanics*, vol. 656, pp. 5–28, 2010.
[7] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data–driven approximation of the koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, 2015.
[8] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature communications*, vol. 9, no. 1, p. 4950, 2018.
[9] N. Takeishi, Y. Kawahara, and T. Yairi, "Learning koopman invariant subspaces for dynamic mode decomposition," *Advances in neural information processing systems*, vol. 30, 2017.
[10] S. E. Otto and C. W. Rowley, "Linearly recurrent autoencoder networks for learning dynamics," *SIAM Journal on Applied Dynamical Systems*, vol. 18, no. 1, pp. 558–593, 2019.
[11] D. Goswami and D. A. Paley, "Global bilinearization and controllability of control-affine nonlinear systems: A koopman spectral approach," in *IEEE Conference on Decision and Control*, 2017.
[12] A. Surana, "Koopman operator based observer synthesis for control-affine nonlinear systems," in *Conference on Decision and Control*. IEEE, 2016, pp. 6492–6499.
[13] C. Folkestad, S. X. Wei, and J. W. Burdick, "Koopnet: Joint learning of koopman bilinear models and function dictionaries with application to quadrotor trajectory tracking," in *International Conference on Robotics and Automation*. IEEE, 2022, pp. 1344–1350.
[14] D. Bruder, X. Fu, and R. Vasudevan, "Advantages of bilinear koopman realizations for the modeling and control of systems with unknown dynamics," *Robotics and Automation Letters*, vol. 6, no. 3, 2021.
[15] M. O. Williams, M. S. Hemati, S. T. Dawson, I. G. Kevrekidis, and C. W. Rowley, "Extending data-driven koopman analysis to actuated systems," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 704–709, 2016.
[16] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of koopman eigenfunctions for control," *Machine Learning: Science and Technology*, vol. 2, no. 3, p. 035023, 2021.
[17] L. C. Iacob, G. I. Beintema, M. Schoukens, and R. Tóth, "Deep identification of nonlinear systems in koopman form," in *Conference on Decision and Control*. IEEE, 2021, pp. 2288–2293.
[18] P. S. Cisneros, A. Datar, P. Göttsch, and H. Werner, "Data-driven quasi-LPV model predictive control using koopman operator techniques," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6062–6068, 2020.
[19] R. Loh, T. Pornthanomwong, J. Pyko, A. Lee, and M. N. Karsiti, "Modeling, parameters identification, and control of an electronic throttle control (ETC) system," in *International Conference on Intelligent and Advanced Systems*. IEEE, 2007, pp. 1029–1035.
[20] E. Witrant, I. D. Landau, and M.-P. Vaillant, "A data-driven control methodology applied to throttle valves," *Control Engineering Practice*, vol. 139, p. 105634, 2023.
[21] P. Daoudi, B. Mavkov, B. Robu, C. Prieur, E. Witrant, M. Barlier, and L. D. Santos, "Improving a proportional integral controller with reinforcement learning on a throttle valve benchmark," *arXiv preprint arXiv:2402.13654*, 2024.
[22] R. N. Loh, W. Thanom, J. S. Pyko, A. Lee *et al.*, "Electronic throttle control system: modeling, identification and model-based control designs," *Engineering*, vol. 5, no. 07, p. 587, 2013.
[23] M. M. Morato, J. E. Normey-Rico, and O. Sename, "Model predictive control design for linear parameter varying systems: A survey," *Annual Reviews in Control*, vol. 49, pp. 64–80, 2020.
[24] P. S. Cisneros, S. Voss, and H. Werner, "Efficient nonlinear model predictive control via quasi-LPV representation," in *IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 3216–3221.
[25] A. Mauroy, Y. Susuki, and I. Mezić, *Koopman operator in systems and control*. Springer, 2020.
[26] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Generalizing koopman theory to allow for inputs and control," *SIAM Journal on Applied Dynamical Systems*, vol. 17, no. 1, pp. 909–930, 2018.
[27] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018.
[28] L. C. Iacob, R. Tóth, and M. Schoukens, "Koopman form of nonlinear systems with inputs," *Automatica*, vol. 162, p. 111525, 2024.
[29] D. Rotondo, F. Nejjari, and V. Puig, "Quasi-LPV modeling, identification and control of a twin rotor mimo system," *Control Engineering Practice*, vol. 21, no. 6, pp. 829–846, 2013.
[30] G. Beintema, R. Toth, and M. Schoukens, "Nonlinear state-space identification using deep encoder networks," in *Learning for dynamics and control*. PMLR, 2021, pp. 241–250.
[31] L. Ljung, "System identification," in *Signal analysis and prediction*. Springer, 1998, pp. 163–173.
[32] C. Adaptech, *Guide d'intégration SBPA*. Adaptech, 2002.
[33] I. D. Landau and G. Zito, *Digital control systems: design, identification and implementation*. Springer, 2006, vol. 130.
[34] M. Vasak, I. Petrovic, and N. Peric, "State estimation of an electronic throttle body," in *International Conference on Industrial Technology*, vol. 1. IEEE, 2003, pp. 472–477.