Deep Learning for Koopman Operator Optimal Control:
IIB Project Technical Milestone Report

Xiaoding Lu, Pembroke College xl402

Nonlinear dynamics are ubiquitous in complex systems; they remain computationally challenging as there exists no generalized mathematical framework for their analysis. The Koopman operator framework is becoming increasingly popular for obtaining linear representations of nonlinear systems from data, thereby enabling comprehensive techniques for linear system analysis and control. Much of the recent research directly seeks eigenfunctions of the Koopman operator [1], [2], as they provide coordinates which globally linearize the nonlinear dynamics. This process is data-driven and benefits from the power of deep learning (DL). Many works also address the subject of optimal control under the Koopman framework [3], [4], although few are DL based. Therefore this project aims to optimally input non-affine nonlinear systems, utilizing DL to discover the Koopman invariant subspace, bridging the gap between DL based Koopman eigenfunction discovery and optimal predictive control.

I. Introduction

A. The Koopman Operator

The classical geometric theory of dynamical systems considers a set of coupled differential equations which can take the form of a discrete map:

$$\mathbf{x}_{k+1} = F(\mathbf{x}_k), \quad k \in \mathbb{Z} \tag{1}$$

where $\mathbf{x}$ belongs to the state space $S \subset \mathbb{R}^n$, and $F : S \to S$ is the dynamic map. The dynamics of F are generally nonlinear and often unknown, i.e. only measurements of the dynamics are available.

In 1931, B.O. Koopman described dynamical systems in terms of the evolution of functions in the Hilbert space $g(\mathbf{x})$ [5], which are measurements of the states rather than the states themselves. The Koopman operator $\mathcal{K}$ advances these measurements one step forward in time:

$$\mathcal{K}g(\mathbf{x}_k) = g(\mathbf{x}_{k+1}) \tag{2}$$

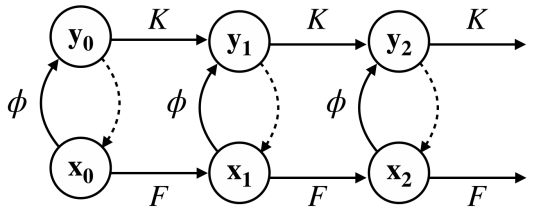


Fig. 1: Illustration of Koopman operator eigenfuntions, observable states $\mathbf{x}$ are lifted by the eigenfunctions ϕ into $\mathbf{y}$ where the dynamics are linear under the Koopman operator K

as $g(\mathbf{x}_{k+1}) = g(F(\mathbf{x}_k))$, equation 2 can be rewritten as:

$$\mathcal{K}g = g \circ F \tag{3}$$

where $\circ$ denotes function composition operator, therefore $\mathcal{K}$ is a linear operator, with which the analysis of nonlinear dynamics can be lifted to a linear regime.

In order to find a finite-dimensional representation of $\mathcal{K}$, one may seek the Koopman eigenfunctions $\{\phi_1, \phi_2, ...\}$ and eigenvalues $\{\lambda_1, \lambda_2, ...\}$ which span the Koopman invariant subspace $\mathcal{U}$. All measurements coming from $\mathcal{U}$ remain within the subspace under the Koopman operator such that:

$$g(\mathbf{x}) = \sum_{k=0}^{\infty} \alpha_k \phi_k(x) \tag{4}$$

$$\mathcal{K}g(\mathbf{x}) = \sum_{k=0}^{\infty} \beta_k \phi_k(x) \tag{5}$$

where $\beta_k = \alpha_k \lambda_k$. When a finite number m of such eigenfunctions is sought, the Koopman operator becomes a matrix $K \in \mathbb{C}^{m \times m}$. The key in applying Koopman analysis is to directly obtain a finite number of exact or approximate Koopman eigenfunctions, i.e. $\mathbf{y} = \phi(\mathbf{x})$, $\phi : \mathbb{R}^n \to \mathbb{C}^m$ and $\mathbf{y}_{k+1} = K\mathbf{y}_k$. This concept is illustrated in figure 1. If ϕ are invertable, then $\mathbf{x}_{k+1}$ can be reconstructed from $\mathbf{y}_{k+1}$.

B. Koopman Eigenfunctions Discovery

Recent advances in computational methods enabled the data-driven discovery of the Koopman eigenfunctions. One such class of methods is *dynamic mode decomposition* (DMD), where snapshots of state measurements are assembled:

$$\mathbf{X}_i = \begin{pmatrix} | & | & & | \\ g(\mathbf{x}_i) & g(\mathbf{x}_{i+1}) & \cdots & g(\mathbf{x}_{M+i}) \\ | & | & & | \end{pmatrix} \quad (6)$$

The algorithm estimates a linear relationship between two data matrices:

$$A = \mathbf{X}_0^{+}\mathbf{X}_1 \quad (7)$$

where $A \in \mathbb{R}^{n \times n}$ is the propagator matrix. When the measurement function $g(\mathbf{x})$ is the identity mapping, i.e. $g(\mathbf{x}) = \mathbf{x}$, for linear systems, the eigenvalues of A and the Koopman operator are identical. For nonlinear systems, one can work with an augmented state consisting of possibly nonlinear *dictionary* functions g of the state, this procedure is termed *extended dynamic mode decomposition* (EDMD) [6]. If Koopman eigenfunctions ϕ are within the span of g, then left eigenvectors of A correspond to ϕ. The performance of EDMD, therefore, depends strongly on the choice of the *dictionary* functions g, which usually requires prior knowledge of the system. Compact dictionaries with high capacities can be achieved by formulating EDMD as a kernel method termed KDMD. However the choice of kernels is empirical, and the algorithm may over-fit the data.

Deep learning based dictionary discovery requires no prior knowledge on the system, and dictionary sparsity is easily implemented through regularization. Many in literature [2], [7] utilize autoencoder-like networks shown in figure 2. The encoder lifts nonlinear dynamics into linear regime: $\mathbf{y} = g(\mathbf{x})$; a linear operator steps $\mathbf{y}$ forward in time: $\mathbf{y_{k+1}} = K\mathbf{y_k}$. The loss function usually takes the form of:

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \mathcal{L}_{\text{recon}} \quad (8)$$

$$= \|\mathbf{x} - \hat{\mathbf{x}}\|_2 + \|\mathbf{y} - \hat{\mathbf{y}}\|_2 \quad (9)$$

where $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ denote the predicted state and linear operator output respectively. The reconstruction cost $\mathcal{L}_{\text{recon}}$ ensures g is invertable so that non-trivial transformations are learnt. Koopman eigenfunctions approximations can be reconstructed similar to EDMD.

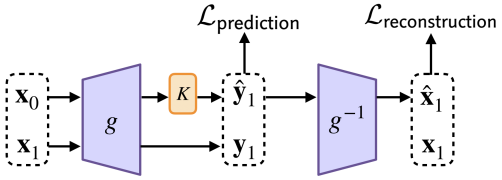


Fig. 2: Koopman autoencoder network architecture

C. Koopman Operator Optimal Control

For affine systems, optimal control is achieved if the observables $\mathbf{x}$ are within the span of the measurements $g(\mathbf{x})$ [8], a trivial case is when $\mathbf{x} \in g(\mathbf{x})$. When this is not the case, approximations of *Koopman operator optimal control* (KOOC) can be achieved when g is invertable [4].

For non-affine systems, by extending equation 1 to allow for control input $u \in \mathcal{U} \subset \mathbb{R}^l$:

$$\mathbf{x}_{k+1} = F(\mathbf{x}_k, \mathbf{u}_k) \quad (10)$$

Korda and Mezic [3] have shown that a set of linear predictors $\mathbf{y} = \psi(\mathbf{x})$, $\psi : \mathbb{R}^n \to \mathbb{R}^m$ can be constructed such that:

$$\mathbf{y}_{k+1} = A\mathbf{y}_k + B\mathbf{u}_k \quad (11)$$

$$\hat{\mathbf{x}}_{k+1} = C\mathbf{y}_{k+1} \quad (12)$$

Equation 11 indicates an optimal controller may be designed for $\mathbf{y}$, which is equivalent to KOOC in $\mathbf{x}$ in the limit that $\hat{\mathbf{x}}_{k+1} = \mathbf{x}_k$.

The problem setup corresponds to a Koopman operator $\mathcal{A} = [A, B]$ operating on the extended state-space $\mathcal{X} = [\mathbf{x}, \mathbf{u}]^T$ such that:

$$\mathcal{A}\phi(\mathcal{X}_k) = \phi(\mathcal{F}(\mathcal{X}_k)) \quad (13)$$

where $\mathcal{F}(\mathcal{X}_k) = [F(\mathbf{x}_k, \mathbf{u}_k), \mathbf{u}_{k+1}]^T$, equation 11 is achieved by constraining ϕ such that:

$$\phi(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \psi(\mathbf{x}) \\ \mathbf{u} \end{bmatrix} \quad (14)$$

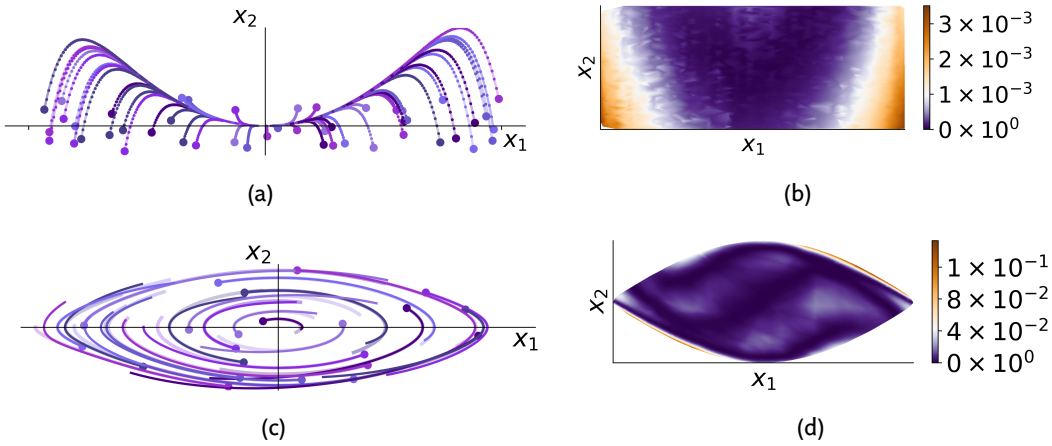


Fig. 3: LREN results. Top: discrete-spectrum example, bottom: continuous-spectrum (simple pendulum) example. Left: validation set predicted trajectories, large dots symbolize initial states, with subsequent dots/lines being consecutive discrete-time predictions; ground truth is shown using transparent lines. Right: validation set mean squared error over long time horizon (5 seconds for both systems) under different state initializations.

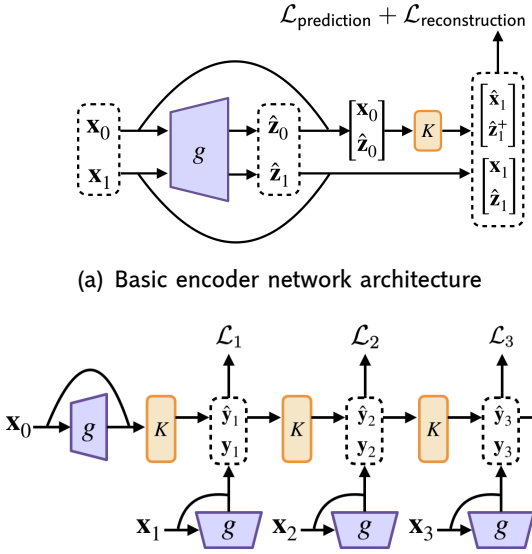
II. Results

A. Linearly Recurrent Encoder Network

The drawback of the autoencoder-like architectures (shown in figure 2) is that system control is not trivial [4], as optimal control over the embedded space $\mathbf{y}$ does not necessarily translate to optimal control over $\mathbf{x}$. The dimension of the K matrix represents the network's representation capacity. When the system exhibits *continuous Koopman spectrum*, eigenfunctions become state-dependent and require an infinite-dimensional K. This issue may be circumvented using an auxiliary network to parameterize K so that $K = \Phi(\mathbf{x})$ [2]; however, one is left with the same problem regarding optimal control.

This work considers a *linearly recurrent encoder network* (LREN) shown in figure 4, which modifies the architecture proposed by Lusch et al. [2]. The network architecture closely relates to the EDMD algorithm, with learnt *dictionary* functions g. Skip connections are added so that $\mathbf{y} = [\mathbf{x}, g(\mathbf{x})]^T$ (shown in figure 4(a)); this ensures that observables $\mathbf{x}$ are within the measurement functions and are advanced linearly by K. The loss function considers the state-reconstruction error $\mathcal{L}_{\text{recon}}$ and linearity of the embedded space (prediction error) $\mathcal{L}_{\text{pred}}$ separately, i.e.

$$\mathcal{L} = \alpha_0 \mathcal{L}_{\text{recon}} + \alpha_1 \mathcal{L}_{\text{pred}} + \alpha_2 \mathcal{L}_{\text{regu}} + \alpha_3 \mathcal{L}_{\text{inf}} \quad (15)$$



(a) Basic encoder network architecture

(b) Linear Recurrency. For simplicity, $\mathbf{y} = [\mathbf{x}, \mathbf{z}]^T$, the encoder g and the propagator K are identical across all parts of the figure

Fig. 4: Linearly recurrent encoder network (LREN) architecture

Where $\mathcal{L}_{\text{regu}} = \|K\|_1$ is the regularization loss, promoting eigenfunction sparsity, and $\mathcal{L}_{\text{inf}}$ penalizes the worst prediction outcome in a mini-batch. Hyperparameters $\{\alpha_i\}_{i=0,\cdots,3}$ denote loss importance. To ensure prediction accuracy over a long prediction time-horizon, the LREN network takes advantage of longer snapshot sequences during training as shown

	Architecture	Parameters	Time Horizon	α_0	α_1	α_2	α_3	Learning Rate
Discrete	$2 \times 5 \times 10 \times 10$	235	5s, $f = 0.05s^{-1}$	0.1	10.0	$1e-10$	$1e-5$	$1e-4$
Pendulum	$2 \times 20 \times 20 \times 100 \times 100$	20,434	10s, $f = 0.1s^{-1}$	1.0	1.0	$1e-5$	$1e-3$	$1e-3$

TABLE I: Network architectures and hyperparameters from discrete and continuous spectra example dynamics. *Architecture* indicates encoder hidden layer dimensions. *Time horizon* indicates total time-length of the data generated, together with the sampling frequency f. $\alpha_0, \cdots, \alpha_3$ denote loss importance for parts of the loss function shown in equation 15.

in figure 4(b). The total loss is therefore expressed as the mean over several timestep predictions, i.e. $\mathcal{L} = \sum_{i=1}^{N} \mathcal{L}_i / N$.
This work considers the discrete (section II-B) and continuous (section II-C) spectra dynamical systems used by Lusch et al. [2]. In contrast to their architecture where optimal control is not easily achieved, our network LREN enables KOOC on both these systems, utilizing a linear quadratic regulator on the embedded state **y**. Input non-affine systems' network architecture is proposed in section III.

B. Discrete Koopman Spectrum Example

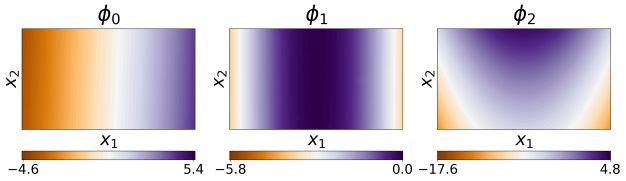


Fig. 5: Discrete Spectrum eigenfunctions composed from LREN

Before analysing systems with continuous Koopman spectra and high-dimensionality, consider a system with a quadratic slow manifold that has a single fixed point (at $\mathbf{x} = 0$) and a discrete eigenvalue spectrum:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{k+1} = \begin{bmatrix} \mu x_1 \\ \lambda (x_2 - x_1^2) \end{bmatrix}_k \tag{16}$$

This dynamic is well studied in literature [8]. A set of intrinsic coordinates that linearize system dynamics are shown to be $[y_1, y_2, y_3]^T = [x_1, x_2, x_1^2]$. In this observable function coordinate system, the Koopman eigenfunctions are:

$$\varphi_0(\mathbf{x}) = x_1, \quad \varphi_1(\mathbf{x}) = x_1^2, \quad \varphi_2(\mathbf{x}) = x_2 - x_1^2 \tag{17}$$

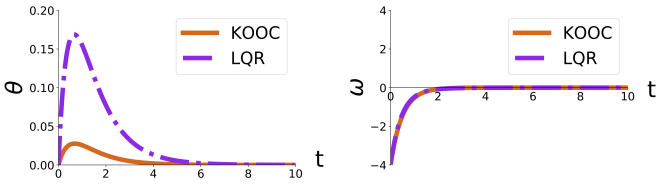


Fig. 6: Discrete spectrum under control: KOOC VS. LQR

which diagnoalize the system and define the intrinsic coordinates.
LREN is able to discover these Koopman eigenfunctions, the encoder architecture as shown in table I. Initial states are sampled at random from $[(-5, 5), (-5, 5)]^T$, figure 3(a) shows 25 actual and predicted trajectories from the validation dataset. The network accurately predicts the system dynamics over a long time-horizon, with figure 3(b) showing the mean squared prediction error for different initial positions. Figure 5 shows the Koopman eigenfunctions extracted from the network, which confirms that the network is able to learn the correct sets of eigenfunctions from equation 17.

We demonstrate KOOC by applying LQR on the embedded space. Figure 6 compares KOOC under the above framework, and naive LQR about the fixed point $\mathbf{x} = 0$, where under the KOOC framework, system settles to the fixed point faster than naive LQR.

C. Continuous Koopman Spectrum Example

As a second example, we consider a nonlinear pendulum with continuous eigenvalue spectrum:

$$\ddot{x} = -\sin(x) \Rightarrow \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\sin(x_1) \end{cases} \tag{18}$$

The LREN architecture can only approximate the continuous spectrum using a finite-dimensional propagator matrix K. Figure 3(c)

shows 25 prediction outputs from the validation set. The network is able to approximate system trajectories, although the mean squared error (figure 3(d)) is a magnitude higher than the discrete-spectrum example. The top 3 eigenfunctions with the highest eigenvalues are visualized in figure 7 using magnitude and phase plots.

The simple pendulum is also a control-affine

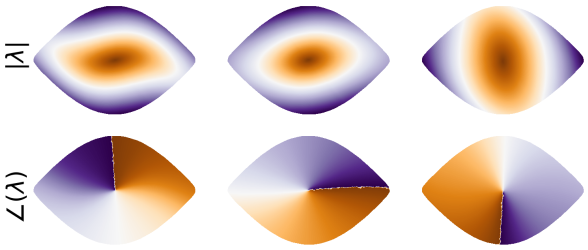


Fig. 7: Top 3 eigenfunctions for nonlinear pendulum example, top: magnitude plot, bottom: phase plot ranging from $-\pi$ to π.

system, KOOC is therefore readily applied. Figures 8 show the angular position and velocity under KOOC and iterative LQR (iLQR), the advantage of KOOC is not apparent in this example. Better control techniques such as *model predictive control* (MPC) are discussed in section III.

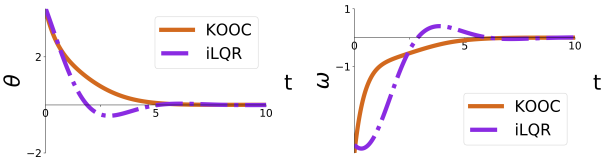


Fig. 8: Nonlinear pendulum under control: KOOC VS. Iterative LQR

III. Future Work

The primary future focuses are: a) improving model performance for systems with continuous Koopman spectra, b) extending the method to cover input non-affine systems. Other potential areas include: analysing model robustness under noisy input and developing theoretical grounds for the best-performing method.

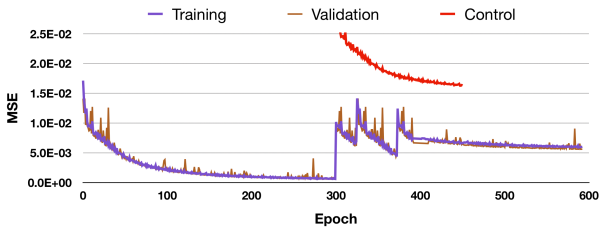
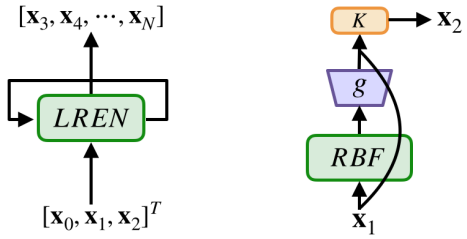


Fig. 9: Custom training strategy for loss minimization over long time-horizon, control: no incremental-training, network tasked to directly predict 100 time-steps



(a) (b)

Fig. 10: Proposed extensions upon LREN architecture. Left: *Delayed LREN* (dLREN), right: LREN with kernel (radial basis) transformation (kLREN)

A. Prediction Accuracy

Prediction accuracy directly affects the quality of KOOC, therefore accurate prediction over long time-horizon is desired. Current network training strategy utilizes an incremental-training approach; i.e. the first 300 epoch are trained by letting the LREN stepping through 30 time-steps, over the next 50 epoch, the prediction time-horizon increases to 50 time-steps. This is repeated until 100 time-steps time-horizon is reached. It is observed that the network settles at a lower mean squared prediction error under the incremental training strategy (figure 9).

Other methods that can potentially improve prediction accuracy are depicted in figure 10. The *Delayed-LREN* (dLREN) shown in figure 10(a) is inspired by *recurrent neural networks*; dLREN encodes N previous predicted trajectories in order to make a prediction on the next trajectory. *Radial basis function* (RBF) networks have advantages of good generalization and strong tolerance to input noise, RBF can add non-linearity to the encoder

without increasing network's depth. Therefore, the *kernel LREN* (kLREN) is considered in figure 10(b), with a RBF layer added before the encoder.

B. Controlling Non-Affine Systems

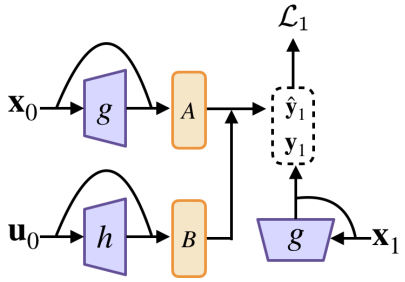


Fig. 11: Double encoder for input non-affine systems (DEINA) network architecture

For non-affine systems with inputs $\mathbf{u} \in \mathbb{R}^l$, following the approach shown in section I-C, equation 11 is modified so that:

$$\mathbf{y}_{k+1} = A\mathbf{y}_k + B\mathbf{q}_k \tag{19}$$

where $\mathbf{q} = \zeta(\mathbf{u})$, $\zeta : \mathbb{R}^l \to \mathbb{R}^o$. Matrix $B \in \mathbb{R}^{m \times o}$ maps the embedded inputs $\mathbf{q}$ to the same dimension as the embedded system states $\mathbf{y} \in \mathbb{R}^m$. This formulation corresponds with constraining the dynamic map ϕ on the extended state-space defined in equation 13 to be of the form:

$$\phi(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \psi(\mathbf{x}) \\ \zeta(\mathbf{u}) \end{bmatrix} \tag{20}$$

This formulation relaxes the strong linear constraint imposed by equation 11 on the input. KOOC is trivial if $\mathbf{x} \in \psi(\mathbf{x})$ and $\mathbf{u} \in \psi(\mathbf{u})$. Following this framework, we propose the *Double Encoder for Input Non-Affine systems* (DEINA) shown in figure 11, where the two encoders lift $\mathbf{x}$ and $\mathbf{u}$ to the embedded space. This network is, to our best knowledge, the only framework that utilizes DL for linearizing and controlling input non-affine systems.

C. Model Predictive Control

Linear *Model Predictive Control* (MPC) is a control strategy where the control input at each time step is obtained by solving a convex optimization problem. The optimisation time-window is typically smaller than the whole horizon, therefore MPC obtains sub-optimal solution than LQR. However, when the system is noisy, which is typical for the lifted states using methods discussed in section II, MPC offers more robust solutions than LQR. Koopman MPC operates on the lifted state, and the computation complexity can be rendered independent of the lifted state's dimension [3].

With much of the literature utilizing Koopman MPC to achieve control, it will be useful to bench mark approaches considered by this project against other methods. Future systems to be considered are the *Van der Pol oscillator* (affine), the *duffing oscillator* (affine) and the *bilinear motor* (non-affine), all of which have been investigated in literature [1], [3], [8] using none DL frameworks.

References

[1] M. Korda and Igor Mezic, "Optimal construction of Koopman eigenfunctions for prediction and control," Tech. Rep., 2019. arXiv: 1810.08733v2.

[2] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature Communications*, vol. 9, no. 1, 2018, issn: 20411723. doi: 10.1038/s41467-018-07210-0. arXiv: 1712.09707.

[3] M. Korda and I. Mezic, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018, issn: 00051098. doi: 10.1016/j.automatica.2018.03.046. arXiv: 1611.03537.

[4] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of Koopman eigenfunctions for control," no. April, 2017. arXiv: 1707.01146. [Online]. Available: http://arxiv.org/abs/1707.01146.

[5] B. O. Koopman, "Hamiltonian systems and transformation in hilbert space," *Proceedings of the National Academy of Sciences*, vol. 17, no. 5, pp. 315–318, May 1931. doi: 10.1073/pnas.17.5.315. [Online]. Available: https://doi.org/10.1073/pnas.17.5.315.

[6] E. Yeung, S. Kundu, and N. Hodas, "Learning deep neural network representations for koopman operators of nonlinear dynamical systems," *Proceedings of the American Control Conference*, vol. 2019-July, pp. 4832–4839, 2019, issn: 07431619. arXiv: 1708.06850.

[7] 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, issn: 15360040. doi: 10.1137/18M1177846. arXiv: 1712.01378.

[8] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," *PLoS ONE*, vol. 11, no. 2, pp. 1–19, 2016, issn: 19326203. doi: 10.1371/journal.pone.0150171. arXiv: 1510.03007.