

Learning Feature Maps of the Koopman Operator: A Subspace Viewpoint

Yingzhao Lian and Colin N. Jones

Abstract—The Koopman operator was recently shown to be a useful method for nonlinear system identification and controller design. However, the scalability of current data-driven approaches is limited by the selection of feature maps. In this paper, we present a new data-driven framework for learning feature maps of the Koopman operator by introducing a novel separation method. The approach provides a flexible interface between diverse machine learning algorithms and well-developed linear subspace identification methods, as well as demonstrating a connection between the Koopman operator and observability. The proposed data-driven approach is tested by learning stable nonlinear dynamics generating hand-written characters, as well as a bilinear DC motor model.

I. INTRODUCTION

The Koopman operator enables the evolution of a nonlinear dynamic system to be represented via linear dynamics, albeit of an infinite dimension. It has been used extensively for the analysis of complex dynamic systems in fields such as fluid mechanics [26] and molecular physics [36] and in recent years it has been proposed as an approach that allows for some linear design tools to be used for nonlinear controller design [17, 19, 30, 35].

Central to this interest is the Dynamic Mode Decomposition (DMD) [27] algorithm and its extensions (e.g. [34, 4]), which are data-driven methods that allow the identification of the Koopman operator from data. In general, an *embedding* (or *lifting*), in which a dictionary of feature maps mapping the original state to a higher dimensional feature space, plays a central role in all applications of the Koopman operator in control engineering. This dictionary of feature maps is normally assumed as prior knowledge, however, as the dimension of the system increases, the number of feature maps that need to be considered tends to increase exponentially. Even though there are methods which learn the feature maps, e.g., [21, 31], the learned maps are normally less interpretable.

The aforementioned difficulty in defining feature maps is the key motivation of this work, in which we propose a novel data-driven method relating Koopman operators to subspace identification methods. Specifically, a novel separation method is applied to divide the original optimization problem into two sub-problems where the system dynamics is first learned by subspace identification methods and then the feature maps are learned. A key benefit of this approach is

This work has received support from the Swiss National Science Foundation under the RISK project (Risk Aware Data Driven Demand Response, grant number 200021_175627)

Yingzhao Lian and Colin N. Jones are with Automatic Laboratory, Ecole Polytechnique Fédérale de Lausanne, Switzerland. {yingzhao.lian, colin.jones}@epfl.ch

that the learning of feature maps allows for the full flexibility of machine learning techniques.

In the following sections, we will first introduce Koopman operator theory and the separation method in Section II. The subspace identification methods and Gaussian process regression will be introduced in Section III to learn the Koopman operator and the feature maps. Section IV will elaborate the connection between the Koopman operator and observability. Finally, the proposed approach is tested by learning stable nonlinear dynamics of hand-written characters and by modeling bilinear DC motor dynamics in Section V.

The major contributions of this paper are summarized as follows:

- Demonstrate the relation between the Koopman operator and subspace identification approaches with the proposed separation method.
- The proposed method provides a flexible interface connecting subspace identification methods and diverse machine learning techniques. Meanwhile, it also improves the scalability of data-driven Koopman operator algorithms by selecting optimal feature maps.
- Show the connection between the Koopman operator and observability, which provides a novel viewpoint of regularization of Koopman operator algorithms.

II. LEARNING THE KOOPMAN OPERATOR: THE SEPARATION METHOD

For the sake of simplicity, we will explain the separation method with an autonomous dynamic system. The incorporation of control inputs into the framework will be discussed in Section III-B

A. Koopman Operator

We consider an autonomous, discrete-time nonlinear dynamical system

$$x^+ = F(x) \quad (1)$$

where $F : \mathcal{M} \rightarrow \mathcal{M}$ is the system update equation and $\mathcal{M} \subseteq \mathbb{R}^n$ is the state space. Our goal in this section is to demonstrate how the Koopman operator can be used to develop an infinite-dimensional linear dynamic system that can be used to compute the state evolution of this system.

Given the function space \mathcal{F} consisting of all functions mapping $\mathcal{M} \rightarrow \mathbb{R}$, called ‘observables’, the Koopman operator [14, 15] applied to the observable $f \in \mathcal{F}$ is defined as

$$\mathcal{K}f = f \circ F$$

The Koopman operator defines a new dynamical system in the function space \mathcal{F} that governs the evolution of the observables.

As the Koopman operator is an operator on a function space, \mathcal{K} is in general infinite-dimensional, but critically it is *linear* even when the dynamics F are non-linear and as such, we call an observable $\phi \in \mathcal{F}$ an eigenfunction associated with the eigenvalue $\lambda \in \mathbb{C}$ if $\mathcal{K}\phi = \lambda\phi$. From this we can see that the eigenfunctions (or linear combinations of the eigenfunctions) evolve linearly along the trajectories of our nonlinear system (1)

$$\phi(x^+) = \phi(F(x)) = \mathcal{K}\phi(x) = \lambda\phi(x) . \quad (2)$$

Instead of tracking the state x of our system, we track the evolution of a set of observables f along the state trajectories. Specifically, given an observable in the span of the eigenfunctions $f = \sum v_k(f)\phi_k$, where the weights $v_k(f)$ are called the Koopman modes of f , we notice that the evaluation of the observable at the current $f(x)$ is a *linear* function of the Koopman eigenfunctions evaluated at the current state

$$f(x) = \sum c_k(f)\phi_k(x)$$

Note that the development here was done for scalar-valued observables for simplicity, but is easily extended to vector-valued functions. The vector form is assumed in the rest of the paper.

Finite Approximation

As the method developed here is learning the Koopman operator from a finite amount of data, a finite approximation is taken. We define a finite set of observables, called the ‘performance outputs’, $\{f_i\} \subset \mathcal{F}$, which define the states and/or outputs of interest for the dynamic system being identified. We assume that the identity function $\mathcal{I}(x) = x$ is contained in the set of performance outputs, which implies that the states of the system is a linear combination of the eigenfunctions.

We assume that there is a finite number of eigenfunctions, written in matrix form as $\Phi = [\phi_1 \ \phi_2 \ \dots]^T$. As the problem is now finite, we can write the Koopman operator as a matrix $\mathcal{K} = K$ and the evolution of the eigenfunctions along the system trajectory as

$$\Phi(x^+) = K\Phi(x)$$

The observables, or performance outputs, are then a linear function of $\Phi(x)$

$$f(x) = V\Phi(x)$$

where $f := [f_1 \ f_2 \ \dots]^T$ and V is a matrix of Koopman modes.

B. Separation Method

We assume the availability of a set of measurements of the observables / performance outputs of the system along system trajectories $\{x_i, y_i\}$, where $y_i = f(x_i)$ are the performance outputs. We pose a least-squares optimization problem to fit the Koopman modes, eigenvalues and eigenfunctions

$$\begin{aligned} \min_{K, V, \Phi} \sum_i \|y_i - V\Phi(x_i)\|^2 + \|\Phi(x_i) - K\Phi(x_{i-1})\|^2 \quad (3) \\ \text{s.t. } \Phi \in \mathcal{F} \end{aligned}$$

If the eigenfunctions Φ are assumed to be in the span of a finite dictionary of known basis functions $\Psi = [\psi_1, \psi_2, \dots, \psi_n]^T$, then (3) can be formulated as a convex least-squares problem, which can be easily solved to find the Koopman modes and eigenvalues V and K . This is the extended dynamic mode decomposition (EDMD) [34]¹ approach, which has been shown to be effective if a good dictionary of functions Ψ is available. However, unlike the optimization problem defined in EDMD and [17] where the Koopman mode and Koopman operator are learned separately with a fixed dictionary of feature maps, we here demonstrate that all three can be learned from data.

Assumption 1: There is a class of universal approximator \mathcal{U} such that for $\forall f \in \mathcal{F}$ and $\forall \epsilon > 0$, $\exists \psi \in \mathcal{U}$ such that $\|f - \psi\|_\infty < \epsilon$, where $\|\cdot\|_\infty$ denotes infinity norm.

Assumption 1 implies that \mathcal{F} is dense in \mathcal{U} . We note that Gaussian process regression and infinitely-wide neural networks are examples of such classes of functions.

Optimization problem (3) is equivalent to the following optimization problem almost everywhere

$$\begin{aligned} \min_{K, V, \{\hat{x}\}, \Psi} \sum_i \|y_i - V\hat{x}_i\|^2 + \|\hat{x}_i - K\hat{x}_{i-1}\|^2 \quad (4) \\ \text{s.t. } \hat{x}_i = \Psi(x_i) \quad \text{for all } x_i \\ \Psi \in \mathcal{U} \end{aligned}$$

This optimization problem admits a relaxation² as:

$$\begin{aligned} \min_{K, V, \{\hat{x}\}, \Psi \in \mathcal{U}} \sum_i \overbrace{\|y_i - V\hat{x}_i\|^2 + \|\hat{x}_i - K\hat{x}_{i-1}\|^2}^1 \quad (5) \\ + \underbrace{\|\hat{x}_i - \Psi(x_i)\|^2}_2 \end{aligned}$$

As indicated in the assumption, the second component of this optimization problem admits an arbitrarily small error and therefore this relaxation is arbitrarily close. Hence, the optimization problems can now be separated into two problems, which can then be solved accordingly.

Problem 1 is solved first

$$\min_{K, V, \{\hat{x}\}} \sum_i \|y_i - V\hat{x}_i\|^2 + \|\hat{x}_i - K\hat{x}_{i-1}\|^2 \quad (6)$$

¹If the basis functions are identity functions of previous measurements, then it is a DMD approach

²Relaxation means that every feasible solution in optimization (5) admits a feasible solution in optimization (4) by enforcing $\hat{x}_i = \Psi(x_i)$

which produces a lifted state sequence \hat{x}_i , as well as the matrices defining the linear dynamics K and V .

Problem 2 is then solved in order to find a mapping from the true system state x_i to the lifted system state \hat{x}_i at the sampled points. The optimization is a classic regression problem.

$$\min_{\Psi \in \mathcal{U}} \sum_i \|\hat{x}_i - \Psi(x_i)\|^2 \quad (7)$$

In general, the key of this separation method is that regardless of the optimal state sequence generated in Problem (6), the optimal value of Problem (7) will be arbitrarily close to zero as a result of Assumption 1. As a result, the loss contributed by sub-optimization Problem (6) dominates the loss from sub-optimization Problem (7), such that Problem (7) is negligible in the original optimization Problem (5). Therefore, without loss of generality, we are able to apply any flexible regression algorithm that meets this requirement.

In this paper, the sub-optimization Problem (7) will be solved by Gaussian process regression which has zero loss and provides an exact interpolation³.

It is worthwhile noting that EDMD is a special case of the proposed method when the dictionary of feature maps consists of infinite basis functions equipped with a reproducing Hilbert kernel space (RHKS) [28]. This will give rise to a kernel linear regression, which has been proven to be the same as the mean function of Gaussian process regression used in this paper. [13]

III. RECONSTRUCTION OF THE KOOPMAN OPERATOR

In this section, we will solve the two sub-optimization problems (6) and (7). We will use the subspace identification method to solve Problem (6) and Gaussian Process regression to solve Problem (7). The relevant techniques for solving the corresponding problems are introduced below and we will conclude this section by bringing all the components together.

A. Subspace Identification

Subspace identification is a class of system identification methods that identify linear state space models of the form⁴:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + Du_k + v_k \end{aligned} \quad (8)$$

where $y_k \in \mathbb{R}^{n_y}$, $x_k \in \mathbb{R}^{n_x}$, $u_k \in \mathbb{R}^{n_u}$, $w_k \in \mathbb{R}^{n_x}$ and $v_k \in \mathbb{R}^{n_y}$ are the system outputs, states, control inputs, system noise and measurement noise. Most subspace identification methods fall in the unifying framework proposed by Van Overschee and De Moor [32]. In general, these algorithms obtain the Kalman filter states from input-output data, and then the state space model is computed by optimization with respect to the special structure in the filtering matrices. The key

³Exact interpolation is guaranteed if the measurement noise is assumed to be zero (Section III-C), however, this could lead to overfitting in practice, therefore, a non-zero measurement noise is used in our implementation.

⁴See, e.g., [33] or [23] for an overview of subspace identification techniques.

component of most of the subspace identification methods is the estimation of the extended observability matrix:

$$\Gamma_f = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{f-1} \end{bmatrix}. \quad (9)$$

In this work, we use the sequential PARSIM algorithm [24].

B. Subspace Identification Based Observable Dynamics

Problem (6) is minimizing the mean squared error of the noise terms w_k and v_k in the following model

$$\begin{aligned} \hat{x}_{k+1} &= K\hat{x}_k + w_k \\ y_k &= V\hat{x}_k + v_k \end{aligned}, \quad (10)$$

which indicates that we can apply a sub-space identification method to identify the corresponding matrices K , V and the states $\{\hat{x}\}$. Based on this observation, we can incorporate control inputs as in general subspace identification methods, where the dynamics are taken to be affine with respect to control input u . This is a heuristic used in [17], and has been shown to provide good performance when modelling systems with control inputs.

The corresponding observable dynamics are

$$\begin{aligned} \hat{x}_{k+1} &= K\hat{x}_k + Bu_k \\ y_k &= V\hat{x}_k + Du_k \end{aligned}. \quad (11)$$

To determine the order of the system, dubbed n_f , which is also the number of eigenfunctions, one might choose different methods, such as cross validation [10] or complexity criteria [1]. We are able to choose different weightings in Problem (6) to achieve an optimal realization up to order n_f [11], which is also the optimal finite approximation of observable dynamics up to order n_f . Because Problem (7) admits an arbitrary small loss, the learnt feature maps are therefore optimal features maps governing the optimal realization up to order n_f .

There are also other approaches proposed to incorporate control inputs into the Koopman operator, such as [22, 35]. Most consider control inputs as part of the states, which fits our proposed approach seamlessly. However, we chose the method used in [17], because it is the most compatible with the well-developed controller design methods in linear systems, especially model predictive control (MPC).

C. Gaussian Process

A Gaussian Process $\mathcal{GP}(\mu, k)$ is an infinite-dimensional distribution over the space of smooth real-valued functions $f : \mathbb{R}^N \rightarrow \mathbb{R}$, specialized by a priori mean $\mu : \mathbb{R}^N \rightarrow \mathbb{R}$ and covariance functions $k : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$ [25], which is also called the kernel function. By definition, function values at a finite set of inputs $[x_1, x_2, x_3 \dots, x_n]$ follows a multi-variate Gaussian distribution $\mathcal{N}(\mu_X, K_{XX})$, where $\mu_X = [\mu(x_1), \dots, \mu(x_n)]^T$ and $K_{XX} = [k(x_i, x_j)]_{i,j=1}^n$. In general, $K_{A,B}$ denotes the cross-covariance between set A and B . If the measurement is contaminated by Gaussian

observation noise, $p(y(x)|f(x)) \sim \mathcal{N}(f(x), \sigma^2)$ with σ^2 as measurement noise variance, then the predictive distribution at any point $x^* \in \mathbb{R}^N$ given data $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$ is

$$\begin{aligned} p(f(x^*)|\mathcal{D}) &\sim \mathcal{GP}(\mu_{f|\mathcal{D}}(x^*), k_{f|\mathcal{D}}(x^*, x^*)) \quad (12) \\ \mu_{f|\mathcal{D}}(x^*) &= \mu(x^*) + K_{x^*X} \hat{K}_{XX}^{-1} y \\ k_{f|\mathcal{D}}(x^*, x^*) &= K_{x^*x^*} - K_{x^*X} \hat{K}_{XX}^{-1} K_{Xx^*}^T \end{aligned}$$

where $\hat{K}_{XX} = K_{XX} + \sigma^2 I$ and $y = [y_1, y_2, \dots, y_n]^T$. This defines a scalar-valued regression, for vector-valued regression, various methods have been proposed [3, 6, 20]. For simplicity, we will do vector-valued regression via a scalar-valued regression in each dimension.

D. Gaussian Process based Feature Maps

From Problem (6), we estimated the value of observables at states x_i as $\hat{x}_i = \Psi(x_i)$. Take one observable $\psi_k(\cdot)$ from the set Ψ for example, and the output of this function at x_i is equal to \hat{x}_i if there is no noise. Then, referring to (12), the expectation of prediction at any $x^* \in \mathbb{R}^{n_x}$ is

$$\mu_{\psi_k|X, \hat{X}}(x^*) = \mu(x^*) + K_{x^*X} K_{XX}^{-1} \hat{X},$$

where X is the matrix stacking all the state measurements x_i and \hat{X} is the matrix stacking the corresponding \hat{x}_i . In particular, we choose $\mu(\cdot) \equiv 0$ to make full use of the flexibility of Gaussian process regression.

E. Discussion

With the solutions of both sub-optimization problems, we get the matrices K, B, V , and D (system (11)) and feature maps $\{\psi_j\}_{j=1}^{n_f}$, where n_f is the order of the observable dynamics as well as the number of feature maps. The interaction between state space dynamics and observable dynamics (*i.e.* feature space dynamics) are summarized as follows:

$$\begin{aligned} \hat{x} &= \psi(x) && \text{state space to observables} \\ \hat{x}_{k+1} &= K\hat{x}_k + Bu_k && \text{observables dynamics} \\ y_k &= V\hat{x}_k + Du_k && \text{observables to state space} \end{aligned}$$

while the dynamics of the ground truth system (which is unknown) is given by

$$x_{k+1} = F(x_k, u_k) \quad \text{state-space dynamics}$$

From this we can see that if we know, or can estimate, the state of the system x_0 , we can then simulate the future evolution of the system observables (which includes the future state x_k) by first evaluating a nonlinear lifting function $\psi(x_0)$, and then iterating a linear dynamic system.

In general, the learned feature maps are not necessarily the eigenfunctions, however, by diagonalization of matrix K , we can recover the eigenfunctions, eigenvalues and the corresponding Koopman modes of the performance outputs [7]. Moreover, it is noteworthy that the state x is a very general concept, one could use previous control inputs or even value function of the MPC controller as a state.

We conclude this section by listing the practical significance of the proposed method.

- The proposed algorithm is fully data-driven in comparison requiring minimal a priori knowledge of the systems.
- In comparison with EDMD, the proposed algorithms provides a reduced order model with an optimal selection of feature maps. which improves the scalability in terms of the dimension of the system.
- Due to linear dynamics in the feature space, the proposed algorithm enables a linear controller design scheme in the feature even for a nonlinear system.
- Because of the proposed separation method, any supervised regression algorithms satisfying Assumption 1 (e.g. neural networks) can be used for learning the feature maps. Therefore, the proposed algorithm provides a flexible interface between linear control theory and diverse supervised learning algorithms.

IV. CONNECTION BETWEEN KOOPMAN OPERATOR AND OBSERVABILITY

In this section, we demonstrate the strong connection between the Koopman operator and observability by answering the following question:

Why do we learn these eigenfunctions but not the others?

The eigenfunctions of the Koopman operator enjoy a special algebraic structure, in particular, the eigenfunctions that vanish nowhere form an Abelian group [7]. Specifically, if ϕ_1, ϕ_2 are eigenfunctions with eigenvalues λ_1, λ_2 respectively, then $\phi_1 \phi_2$ is also an eigenfunction with eigenvalue $\lambda_1 \lambda_2$. Based on this observation, the eigenfunctions related to an unstable mode or to a slow dynamic mode should dominate the learning problem, which is also the reason why Dynamic Mode Decomposition (DMD) tends to be numerically ill-conditioned. In order to learn the family of eigenfunctions, [16] proposed to construct a large family of possible eigenfunctions. However, we argue that instead of learning all the possible observable dynamics, we should select the eigenfunctions that have strong relations with the observables that are of special interest, which are termed ‘performance outputs’ in this paper. For example, if we control the temperature of a room, then we should learn the eigenfunctions that could properly reproduce the temperature dynamics but not the others. We thereby propose to use a well-developed concept in control science, observability, to quantify/regularize and to select the possible eigenfunctions.

For the sake of simplicity, we elaborate on this idea with a discrete-time, autonomous dynamical system. Assume that $\{\phi_k\}_{k=1}^K$ is the set of eigenfunctions with corresponding eigenvalues $\{\lambda_k\}$. Let f denote an observable that we focus on, and let its Koopman mode corresponding to ϕ_k be v_k , then

$$\mathcal{K}^i f = \hat{V} \Lambda^i \Phi(x), \quad (13)$$

where $\hat{V} = [v_1, v_2, \dots, v_K]$, $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_K)$ and $\Phi(x) = [\phi_1(x), \phi_2(x), \dots, \phi_K(x)]^T$. Given a sequence of N_i evaluations $y_i = f(x_i)$ of the observable f along the

state sequence $\{x_i\}$, we have

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N_i} \end{bmatrix} = \hat{\Gamma} \Phi(x_0) = \begin{bmatrix} \hat{V} \Lambda^0 \\ \hat{V} \Lambda^1 \\ \vdots \\ \hat{V} \Lambda^{N_i-1} \end{bmatrix} \Phi(x_0). \quad (14)$$

Since the value of the eigenfunctions are fixed for a fixed x_0 , the dynamics of the observable is determined by $\hat{\Gamma}$. This matrix leads to two observations:

- 1) The matrix $\hat{\Gamma}$ is exactly an extended observability matrix appearing in equation (9).
- 2) The dynamics of an observable depends both on the Koopman mode and the eigenvalues. Even though a eigenfunction may have a large eigenvalue, if it has a small Koopman mode with respect to the observable f (i.e. this eigenfunction has lower observability in f), then the dynamics corresponding this eigenfunction is less relevant to the outputs that we would like to predict.

The observations above suggest an answer to the question we posed at the beginning of this section. Because the quality of the reconstruction of the performance outputs are considered in the original optimization problem (3), the eigenvalues learned from the proposed algorithms reflect a balance between state evolution and performance outputs reconstruction. The key piece of a subspace identification method is exactly estimating the extended observability matrix, from which the most observable components are extracted by singular value decomposition (SVD).

We conclude this section by pointing out the major significant differences between EDMD and our algorithm as follows:

- 1) Our algorithm does not assume a dictionary of basis functions is known explicitly, which results in a more scalable approach and one that is less sensitive to tuning parameters (the selection of the dictionary).
- 2) Our algorithm considers the evolution of the observables explicitly, and the corresponding learned feature maps are optimal with respect to their observability.

Remark 1: We note that the DMD approach [26] starts from a Krylov space and develops a matrix that is in observable canonical form, which also indicates a link between observability and the identification of the Koopman operator.

V. VALIDATION

In this section, we will run system identification on two systems, including one autonomous system and one system with control inputs. The Gaussian Process Regression toolbox in MATLAB [18] is used for the GP regression implementation. Both examples uses automatic relevance deterministic (ARD) kernel, whose hyperparameters are optimized through maximal likelihood.

A. Learning Stable Hand-written Character Dynamics

Learning by demonstration is a technique that enables robots to reproduce tasks by providing demonstrations [5].

Different approaches have been proposed to tackle this problem [2, 37, 8], however, the global stability of the learned dynamics is relatively hard to ensure. In [12], a Gaussian mixture model (GMM) is applied to ensure the global stability where the learned dynamical system admits a Lyapunov function. This approach is not scalable because GMM suffers from the curse of dimensionality. The method proposed in this work gives a possible stable and scalable solution.

We assume the desired task admits an autonomous, discrete-time dynamical system such that the state x evolves according to $x_{k+1} = F(x_k)$. By applying the Koopman operator, there exists linear observable dynamics, such that $\hat{x}_{k+1} = A\hat{x}_k$. The model identified has the form

$$\begin{aligned} \hat{x}_0 &= \psi(x_0) && \text{states space to observables} \\ \hat{x}_{k+1} &= A\hat{x}_k && \text{observables dynamics} \\ x_k &= C\hat{x}_k && \text{observables to state space} \end{aligned}$$

In order to ensure global stability of the system, we need to ensure that the observable dynamics are globally stable, which is true if and only if [29]

$$A^T A \preceq I, \quad (15)$$

where I is the identity matrix and \preceq stands for generalized inequality with respect to the positive semi-definite cone. Therefore, once we have the estimated extended observability matrix Γ_f (9) from the subspace identification method, we integrate the stability constraint and cast the optimization problem as the following semi-definite programming (SDP) problem:

$$\begin{aligned} \min_A & \left\| \Gamma_f \begin{bmatrix} A \\ -\mathbf{1}^T \end{bmatrix} \right\|_F \\ \text{s.t.} & \begin{bmatrix} I & A^T \\ A & I \end{bmatrix} \succeq 0 \end{aligned}$$

where $\mathbf{1}$ is a vector of ones.

The effectiveness of this procedure is demonstrated via the modelling of character dynamics, whose data comes from [12]. A few demonstrations of hand-written characters have been recorded and the proposed method is applied to learn the trajectory of the dynamics such that we can rewrite the character automatically. Figure 1 and Figure 2 show the learned dynamics of character r and N respectively, from which we can observe that the prediction based on the learned dynamics properly reproduce the desired character from a different initial point. Moreover, the estimated vector field also demonstrates that the end point is a global asymptotically stable attractor.

B. Bilinear Motor

In this section, we apply the proposed method to identify the Koopman operator of a bilinear model of a DC motor [9].

$$\begin{aligned} \dot{x}_1 &= -(R_a/L_a)x_1 + (k_m/L_a)x_2u + u_a/L_a \\ \dot{x}_2 &= -(B/J)x_2 + (k_m/J)x_1u - \tau_l/J \\ y &= x_2 \end{aligned}$$

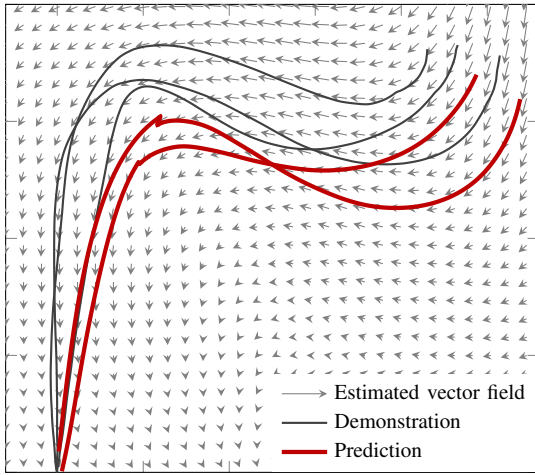


Fig. 1. Learned 18th order dynamics of the character ‘r’

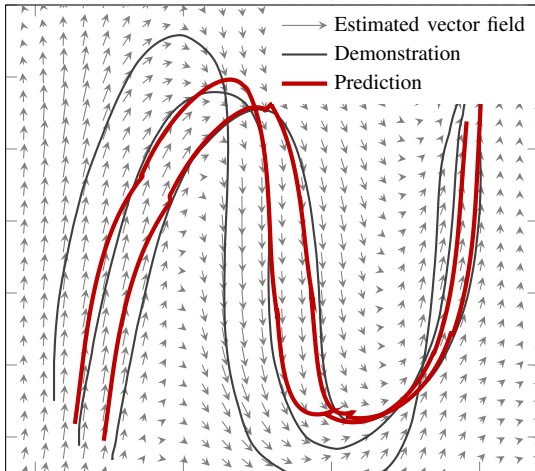


Fig. 2. Learned 22th order dynamics of the character ‘N’

where x_1 is the rotor current, x_2 is the angular velocity and the control input u is the stator current. The parameters are $L_a = 0.314$, $R_a = 12.345$, $k_m = 0.253$, $J = 0.00441$, $B = 0.00732$, $\tau_l = 1.47$, and $u_a = 60$. This model is also used in [17], where 103 basis functions are used to model the feature maps based on EDMD algorithm. However, using the technique developed in this paper, only 30 feature maps are needed to reproduce a good model, which demonstrates a much higher scalability than the EDMD algorithm. The validation of the model is shown in Figure 3, where also a local linearization of the model is used for comparison.

We run tests with 100 randomly sampled initial states and run a 1.5 second simulation with a random control input sequence. The root mean square error (RMSE) for the model proposed by this paper is 0.021 while the RMSE for the model from [17] is 0.137 for the same initial states. In conclusion, our proposed method has much higher scalability and performance than current EDMD approaches.

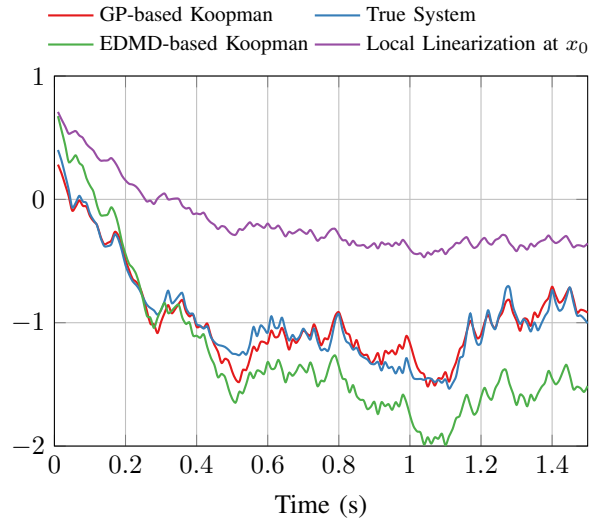


Fig. 3. Comparison of the prediction given Koopman operator and the real state evolution

VI. CONCLUSION

This paper has proposed a novel data-driven method for learning feature maps of Koopman operator, which results in a significant performance and scalability increase. In addition, the method offers a flexible interface between subspace identification methods and diverse machine learning techniques for this class of problems.

REFERENCES

- [1] Hirotugu Akaike. “A new look at the statistical model identification”. In: *Selected Papers of Hirotugu Akaike*. Springer, 1974, pp. 215–222.
- [2] Jacopo Aleotti and Stefano Caselli. “Robust trajectory learning and approximation for robot programming by demonstration”. In: *Robotics and Autonomous Systems* 54.5 (2006), pp. 409–413.
- [3] Mauricio A Álvarez and Neil D Lawrence. “Computationally efficient convolved multiple output Gaussian processes”. In: *Journal of Machine Learning Research* 12.May (2011), pp. 1459–1500.
- [4] Hassan Arbabi and Igor Mezić. “Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator”. In: *SIAM Journal on Applied Dynamical Systems* 16.4 (2017), pp. 2096–2126.
- [5] A Billard et al. “Survey: Robot programming by demonstration”. In: *Handbook of Robotics* 59.BOOK_CHAP (2008).
- [6] Edwin V Bonilla, Kian M Chai, and Christopher Williams. “Multi-task Gaussian process prediction”. In: *Advances in neural information processing systems*. 2008, pp. 153–160.
- [7] Marko Budišić, Ryan Mohr, and Igor Mezić. “Applied koopmanism”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 22.4 (2012), p. 047510.

- [8] Adam Coates, Pieter Abbeel, and Andrew Y Ng. “Learning for control from multiple demonstrations”. In: *Proceedings of the 25th international conference on Machine learning*. ACM. 2008, pp. 144–151.
- [9] S Daniel-Berhe and H Unbehauen. “Experimental physical parameter estimation of a thyristor driven DC-motor using the HMF-method”. In: *Control Engineering Practice* 6.5 (1998), pp. 615–626.
- [10] Bradley Efron. “The estimation of prediction error: covariance penalties and cross-validation”. In: *Journal of the American Statistical Association* 99.467 (2004), pp. 619–632.
- [11] Tony Gustafsson. “Subspace-based system identification: weighting and pre-filtering of instruments”. In: *Automatica* 38.3 (2002), pp. 433–443.
- [12] S Mohammad Khansari-Zadeh and Aude Billard. “Learning stable nonlinear dynamical systems with gaussian mixture models”. In: *IEEE Transactions on Robotics* 27.5 (2011), pp. 943–957.
- [13] George Kimeldorf and Grace Wahba. “Some results on Tchebycheffian spline functions”. In: *Journal of mathematical analysis and applications* 33.1 (1971), pp. 82–95.
- [14] Bernard O Koopman. “Hamiltonian systems and transformation in Hilbert space”. In: *Proceedings of the National Academy of Sciences* 17.5 (1931), pp. 315–318.
- [15] BO Koopman and J v Neumann. “Dynamical systems of continuous spectra”. In: *Proceedings of the National Academy of Sciences* 18.3 (1932), pp. 255–263.
- [16] Milan Korda and Igor Mezić. “Learning Koopman eigenfunctions for prediction and control: the transient case”. In: *arXiv preprint arXiv:1810.08733* (2018).
- [17] Milan Korda and Igor Mezić. “Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control”. In: *Automatica* 93 (2018), pp. 149–160.
- [18] MATLAB. *version 9.4.0 (R2010a)*. Natick, Massachusetts: The MathWorks Inc., 2018.
- [19] Alexandre Mauroy and Jorge Goncalves. “Linear identification of nonlinear systems: A lifting technique based on the Koopman operator”. In: *arXiv preprint arXiv:1605.04457* (2016).
- [20] Charles A Micchelli and Massimiliano Pontil. “On learning vector-valued functions”. In: *Neural computation* 17.1 (2005), pp. 177–204.
- [21] Samuel E Otto and Clarence W Rowley. “Linearly-recurrent autoencoder networks for learning dynamics”. In: *arXiv preprint arXiv:1712.01378* (2017).
- [22] Joshua L Proctor, Steven L Brunton, and J Nathan Kutz. “Generalizing Koopman theory to allow for inputs and control”. In: *SIAM Journal on Applied Dynamical Systems* 17.1 (2018), pp. 909–930.
- [23] S Joe Qin. “An overview of subspace identification”. In: *Computers & chemical engineering* 30.10-12 (2006), pp. 1502–1513.
- [24] S Joe Qin, Weilu Lin, and Lennart Ljung. “A novel subspace identification approach with enforced causal models”. In: *Automatica* 41.12 (2005), pp. 2043–2053.
- [25] Carl Edward Rasmussen. “Gaussian processes in machine learning”. In: *Advanced lectures on machine learning*. Springer, 2004, pp. 63–71.
- [26] Clarence W Rowley et al. “Spectral analysis of nonlinear flows”. In: *Journal of fluid mechanics* 641 (2009), pp. 115–127.
- [27] Peter J Schmid. “Dynamic mode decomposition of numerical and experimental data”. In: *Journal of fluid mechanics* 656 (2010), pp. 5–28.
- [28] Matthias Seeger. “Gaussian processes for machine learning”. In: *International journal of neural systems* 14.02 (2004), pp. 69–106.
- [29] Jean-Jacques E Slotine, Weiping Li, et al. *Applied nonlinear control*. Vol. 199. 1. Prentice hall Englewood Cliffs, NJ, 1991.
- [30] Amit Surana and Andrzej Banaszuk. “Linear observer synthesis for nonlinear systems using Koopman operator framework”. In: *IFAC-PapersOnLine* 49.18 (2016), pp. 716–723.
- [31] Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. “Learning Koopman invariant subspaces for dynamic mode decomposition”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 1130–1140.
- [32] Peter Van Overschee and Bart De Moor. “A unifying theorem for three subspace system identification algorithms”. In: *Automatica* 31.12 (1995), pp. 1853–1864.
- [33] Peter Van Overschee and BL De Moor. *Subspace identification for linear systems: Theory—Implementation—Applications*. Springer Science & Business Media, 2012.
- [34] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. “A data-driven approximation of the koopman operator: Extending dynamic mode decomposition”. In: *Journal of Nonlinear Science* 25.6 (2015), pp. 1307–1346.
- [35] Matthew O Williams et al. “Extending data-driven Koopman analysis to actuated systems”. In: *IFAC Symposium on Nonlinear Control Systems (NOLCOS)*. 2016.
- [36] Hao Wu et al. “Variational Koopman models: slow collective variables and molecular kinetics from short off-equilibrium simulations”. In: *The Journal of Chemical Physics* 146.15 (2017), p. 154104.
- [37] Katsu Yamane, James J Kuffner, and Jessica K Hodgins. “Synthesizing animations of human manipulation tasks”. In: *ACM Transactions on Graphics (TOG)* 23.3 (2004), pp. 532–539.