Controlling Nonlinear Dynamical Systems with Linear Quadratic Regulator-based Policy Networks in Koopman space

Tomoharu Iwata and Yoshinobu Kawahara

***Abstract*— We propose a data-driven control method for nonlinear dynamical systems based on the Koopman operator theory. Existing Koopman-based control methods apply linear optimal control methods after system identification by approximating the original cost function in the Koopman space. Therefore, errors in system identification and cost approximation deteriorate the control performance. On the other hand, the proposed method directly maximizes the control performance with reinforcement learning, where a controller is modeled by a neural network that consists of a linear quadratic regulator and an encoder that embeds data into the Koopman space. We experimentally demonstrate the effectiveness of the proposed method over existing Koopman-based and reinforcement learning-based methods with two nonlinear dynamical systems.**

I. INTRODUCTION

Controlling nonlinear dynamical systems is an important task in a wide variety of fields, such as engineering, physics, epidemiology, and sociology [6], [18]. Recently, the Koopman operator theory [13], [23] has attracted attention for data-driven control. Based on the theory, a nonlinear dynamical system is lifted to the corresponding linear one in a possibly infinite-dimensional space by embedding states using a nonlinear function. Therefore, various methods developed for controlling linear dynamical systems can be extended to nonlinear [24], [17].

For the lifting, we need an appropriate encoder to embed data in a finite-dimensional subspace, where the functions span a function space invariant to the Koopman operator. We call this subspace the Koopman space. To automatically learn encoders from data, several methods based on neural networks have been proposed [29], [20], [31], [16], [1], [9]. These methods have shown to achieve high performance on predicting future values due to the high representation power of neural networks. With existing control methods based on the Koopman space [7], [17], [14], [10], [32], [21], [8], [22], system identification is firstly performed, and then, control methods for linear dynamics are applied while fixing the identified system and approximating the original cost function in the Koopman space. Since encoders and systems are trained to improve the prediction performance without considering control, errors in system identification and cost approximation cannot be corrected in learning controllers.

In this paper, we propose a method for learning encoders and dynamics in the Koopman space that directly maximizes the control performance based on reinforcement learning. We consider a task of making measurement vectors close to a predefined target value for a black-box nonlinear dynamical system. By representing the task as a linear quadratic problem in the Koopman space, we can obtain a controller based on a linear quadratic regulator (LQR) [15] that takes a Koopman embedding as input. We combine an encoder with the LQR for transforming it to a controller that takes a measurement vector as input. The controller can be seen as a neural network, in which an encoder and LQR are used as its layers. We use the controller as a policy network within the reinforcement learning framework, and train it by the policy gradient method [28] to maximize the rewards defined based on the control performance. Since the LQR layers, which include procedures for solving a Riccati equation, are differentiable, we can backpropagate the expected rewards to update the controller. Figure 1 illustrates our proposed method.

The remainder of this paper is organized as follows. In Section II, we briefly describe related works. In Section III, we explain the Koopman operator theory, on which the proposed method is based. In Section IV, we formulate our problem, and present our method. In Section V, we experimentally demonstrate that the proposed method achieves better control performance than the existing method using FitzHugh-Nagumo and Lorenz equations. Finally, we present concluding remarks and discuss future work in Section VI.

II. RELATED WORK

Optimal control theory has been studied for a long time [12]. LQRs are developed in the optimal control theory, and have been successfully used in a wide variety of applications. With LQRs, we can obtain the optimal control when dynamics is linear and the cost is quadratic. The proposed method incorporates LQRs for controlling nonlinear dynamical systems based on the Koopman operator theory.

Reinforcement learning methods are categorized into two types: model-free, and model-based. With neural networks, model-free reinforcement learning gives a flexible way to learn controllers, or policies [27]. The proposed method is model-free in a sense that a policy network is trained to maximize the expected reward. It can also be seen as model-based in a sense that it models linear dynamics and finds an LQR in the Koopman space.

T. Iwata is with NTT Communication Science Laboratories, Japan, tomoharu.iwata.gy@hco.ntt.co.jp

Y. Kawahara is with Institute of Mathematics for Industry, Kyushu University, Japan, and with Center for Advanced Intelligence Project, RIKEN, Japan, kawahara@imi.kyushu-u.ac.jp

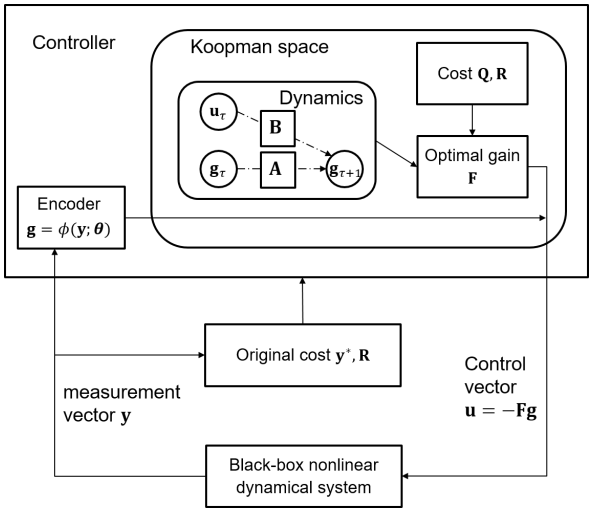


Fig. 1. Our proposed method. At each timestep, the controller receives a measurement vector from a black-box nonlinear dynamical system. The controller then chooses a control vector, which is subsequently sent to the system. The controller assumes linear dynamics and quadratic cost in the Koopman space. The cost is approximated from the original cost in the measurement space. The optimal gain is calculated from the dynamics and cost based on linear quadratic regulators. In the controller, a measurement vector is embedded in the Koopman space by the encoder. The control vector is determined using the optimal gain and Koopman embedding. The parameters of the controller, which consist of the dynamics in the Koopman space and the parameters of the encoder, are trained by minimizing the original cost using reinforcement learning.

III. Preliminaries: Koopman operator theory

We consider nonlinear discrete-time dynamical system,

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t), \tag{1}$$

where $\mathbf{x}_t \in \mathcal{X}$ is the state at timestep t. Koopman operator $\mathcal{A}$ is defined as an infinite-dimensional linear operator that acts on observables $g : \mathcal{X} \to \mathbb{R}$ (or $\mathbb{C}$) [13],

$$g(\mathbf{x}_{t+1}) = \mathcal{A}g(\mathbf{x}_t), \tag{2}$$

with which the analysis of nonlinear dynamics can be lifted to a linear (but infinite-dimensional) regime.

Although the existence of the Koopman operator is theoretically guaranteed in various situations, its practical use is limited by its infinite dimensionality. We can assume the restriction of $\mathcal{A}$ to a finite-dimensional subspace $\mathcal{G}$ [29]. If $\mathcal{G}$ is spanned by a finite number of functions $\{g_1, \dots, g_K\}$, then the restriction of $\mathcal{A}$ to $\mathcal{G}$, which we denote $\mathbf{A} \in \mathbb{R}^{K \times K}$, becomes a finite-dimensional operator,

$$\mathbf{g}_{t+1} = \mathbf{A}\mathbf{g}_t, \tag{3}$$

where $\mathbf{g}_t = [g_1(\mathbf{x}_t), \dots, g_K(\mathbf{x}_t)] \in \mathbb{R}^K$ is a vector of observables at timestep t.

IV. Proposed method

A. Problem formulation

We assume the following nonlinear dynamical system,

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t), \quad \mathbf{y}_t = h(\mathbf{x}_t), \tag{4}$$

where $\mathbf{x}_t \in \mathcal{X}$ is the state, $\mathbf{y}_t \in \mathbb{R}^D$ is the measurement vector, and $\mathbf{u}_t \in \mathbb{R}^J$ is the exogenous control vector at timestep t. Functions f and h are black-boxes, i.e., we do not know them, but we can select control vector $\mathbf{u}_t$ and observe measurement vector $\mathbf{y}_{t+1}$ for each timestep. Our aim is to find appropriate control sequence $\{\mathbf{u}_t\}_{t=1}^T$ that makes the measurement vectors close to a target $\mathbf{y}^*$. It is evaluated by the following cost function,

$$E = \sum_{t=1}^{T} \Big(\| \mathbf{y}_t - \mathbf{y}^* \|^2 + \mathbf{u}_t^\top \mathbf{R} \mathbf{u}_t \Big), \tag{5}$$

where the first term is the distance between the observed and target measurements, and the second term is the cost of controls defined by $\mathbf{R} \in \mathbb{R}^{J \times J}$.

B. System identification with control in Koopman space

According to the Koopman operator theory, we consider the following linear dynamics with control in a finite-dimensional Koopman space [26], [2],

$$\mathbf{g}_{t+1} = \mathbf{A}\mathbf{g}_t + \mathbf{B}\mathbf{u}_t, \tag{6}$$

where $\mathbf{g}_t \in \mathbb{R}^K$ is the observables at timestep t, $\mathbf{A} \in \mathbb{R}^{K \times K}$ is a finite-dimensional approximation of the Koopman operator, and $\mathbf{B} \in \mathbb{R}^{K \times J}$ represents the effect of a control vector on the observables at the next timestep.

To realize a linear dynamics in the Koopman space, we obtain the observables as a function of measurement vector, $\phi : \mathbb{R}^D \to \mathbb{R}^K$,

$$\mathbf{g}_t = \phi(\mathbf{y}_t), \tag{7}$$

where ϕ is a feed-forward neural network, and we call it an encoder.

Suppose that we are given a set of N sequences of measurement and control vectors, $\mathcal{D} = \{\{(\mathbf{y}_{nt}, \mathbf{u}_{nt})\}_{t=1}^T\}_{n=1}^N$ using the dynamical system in Eq. (4), where $\mathbf{y}_{nt}$ and $\mathbf{u}_{nt}$ are the measurement and control vectors at timestep t in the nth sequence. Let $\psi : \mathbb{R}^K \to \mathbb{R}^D$ be is a decoder modeled by a feed-forward neural network that maps the observables into the measurement space. We can estimate $\mathbf{A}$, $\mathbf{B}$, and parameters of ϕ and ψ by minimizing the following objective function,

$$H = \sum_{n=1}^{N} \Bigg(\sum_{t=1}^{T} \| \mathbf{y}_{nt} - \psi(\mathbf{g}_{nt}) \|^2 + \eta \sum_{t=1}^{T-1} \| \mathbf{y}_{n,t+1} - \psi(\mathbf{A}\mathbf{g}_{nt} + \mathbf{B}\mathbf{u}_{nt}) \|^2 \Bigg), \tag{8}$$

where the first term is the reconstruction loss, the second term is the prediction loss when the next measurement vector is predicted through the Koopman space, and $\eta > 0$ is the hyperparameter.

C. Control via Koopman space

We approximate the cost in the measurement space in Eq. (5) by the cost in the Koopman space,

$$\tilde{E} = \sum_{t=1}^{T} \Big(\parallel \mathbf{g}_t - \mathbf{g}^* \parallel^2 + \mathbf{u}_t^\top \mathbf{R} \mathbf{u}_t \Big), \tag{9}$$

where $\mathbf{g}^* = \phi(\mathbf{y}^*)$ is the observables of target measurement $\mathbf{y}^*$. Here, we assume that if observables are closely located in the Koopman space, their measurement vectors are also closely located in the measurement space.

The approximated cost (9) can be rewritten in a quadratic form,

$$\tilde{E} = \sum_{t=1}^{T} \left(\mathbf{g}_t'^\top \mathbf{Q} \mathbf{g}_t' + \mathbf{u}_t^\top \mathbf{R} \mathbf{u}_t \right) + T \parallel \mathbf{g}^* \parallel^2, \tag{10}$$

where

$$\mathbf{g}_t' = \begin{bmatrix} \mathbf{g}_t \\ 1 \end{bmatrix} \in \mathbb{R}^{K+1}, \tag{11}$$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{I} & -\mathbf{g}^* \\ -\mathbf{g}^{*\top} & 0 \end{bmatrix} \in \mathbb{R}^{(K+1)\times(K+1)}. \tag{12}$$

Using Eq.(6), the dynamics of $\mathbf{g}_t'$ is given by

$$\mathbf{g}_{t+1}' = \mathbf{A}' \mathbf{g}_t' + \mathbf{B}' \mathbf{u}_t, \tag{13}$$

where

$$\mathbf{A}' = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathbb{R}^{(K+1)\times(K+1)}, \tag{14}$$

$$\mathbf{B}' = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{(K+1)\times J}. \tag{15}$$

Since the dynamics is linear in Eq. (6) and the cost is quadratic in Eq. (10), linear quadratic regulators (LQRs) are applicable in the Koopman space. With LQRs, the optimal control vector is given by

$$\mathbf{u}_t = -\mathbf{F} \mathbf{g}_t', \tag{16}$$

where $\mathbf{F} \in \mathbb{R}^{J\times(K+1)}$ is the optimal gain matrix calculated by

$$\mathbf{F} = (\mathbf{R} + \mathbf{B}'^\top \mathbf{P} \mathbf{B}')^{-1} (\mathbf{B}'^\top \mathbf{P} \mathbf{A}'). \tag{17}$$

$\mathbf{P} \in \mathbb{R}^{(K+1)\times(K+1)}$ is the solution of the following discrete-time Riccati equation,

$$\begin{aligned} \mathbf{P} &= \mathbf{A}'^\top \mathbf{P} \mathbf{A}' \\ &- (\mathbf{A}'^\top \mathbf{P} \mathbf{B}')(\mathbf{R} + \mathbf{B}'^\top \mathbf{P} \mathbf{B}')^{-1} (\mathbf{B}'^\top \mathbf{P} \mathbf{A}') + \mathbf{Q}. \end{aligned} \tag{18}$$

The Riccati equation can be solved by iterating Eq. (18) until convergence with initialization $\mathbf{P} = \mathbf{Q}$.

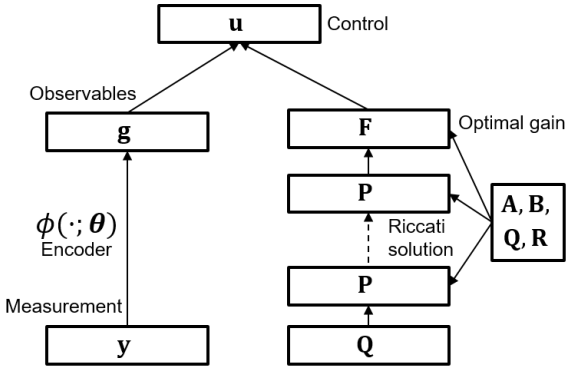


Fig. 2. Our policy network π. The parameters in π to be estimated are $\mathbf{\Psi} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\theta}\}$, where $\mathbf{A}$ and $\mathbf{B}$ are the dynamics in the Koopman space in Eq. (6), and $\boldsymbol{\theta}$ is the parameters of encoder ϕ. First, solution $\mathbf{P}$ of the Riccati equation in Eq. (18) is solved by iterating Eq. (18) using $\mathbf{A}$, $\mathbf{B}$, $\mathbf{Q}$, and $\mathbf{R}$ with initialization $\mathbf{P} = \mathbf{Q}$. Second, optimal gain matrix $\mathbf{F}$ is calculated using $\mathbf{P}$, $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{R}$ by Eq. (17). Third, measurement vector $\mathbf{y}$ is transformed to observable $\mathbf{g}$ in the Koopman space by encoder ϕ with parameter $\boldsymbol{\theta}$. Finally, control vector $\mathbf{u}$ is obtained using $\mathbf{F}$ and $\mathbf{g}$ by Eq. (16). Our policy network can be seen as a neural network with parameters $\mathbf{\Psi}$ that takes measurement vector $\mathbf{y}$ as input and outputs control vector $\mathbf{u}$.

D. Improving control with reinforcement learning

If we perfectly identify the dynamical system $\mathbf{A}, \mathbf{B}, \phi$ via the Koopman space, and the cost in the Koopman space in Eq. (9) is exactly equivalent to that in the measurement space in Eq. (5), the control based on LQRs in Eq. (16) is optimal. However, the system identification is difficult due to its nonlinearity and measurement noise. In addition, the cost in the Koopman space might not approximate that in the measurement space well when these two spaces have far different metrics.

Let $\boldsymbol{\theta}$ be parameters of encoder ϕ. To improving control via the Koopman space, we propose to tune parameters $\mathbf{\Psi} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\theta}\}$ using reinforcement learning, such that the cost in the measurement is minimized when an LQR is adopted in the Koopman space. Our problem in Section IV-A can be defined on a Markov decision process (MDP), where the current measurement vector $\mathbf{y}_t$ is a state of the MDP, the control vector $\mathbf{u}_t$ is an action, and the cost at a timestep

$$E(\mathbf{y}, \mathbf{u}) = \parallel \mathbf{y} - \mathbf{y}^* \parallel^2 + \mathbf{u}^\top \mathbf{R} \mathbf{u}, \tag{19}$$

is a negative reward. Let $\pi : \mathbb{R}^D \to \mathbb{R}^J$ be a controller, or policy in the MDP, that takes a measurement vector as input, and outputs an appropriate control vector. We model policy π using $\mathbf{A}$, $\mathbf{B}$ and ϕ considering an LQR in the Koopman space as in Eq. (16),

$$\pi(\mathbf{y}; \mathbf{\Psi}) = -\mathbf{F}(\mathbf{A}, \mathbf{B}) [\phi(\mathbf{y})^\top 1]^\top, \tag{20}$$

where $\mathbf{F}(\mathbf{A}, \mathbf{B})$ is the optimal gain matrix obtained by Eqs. (17,18), which is a function of $\mathbf{A}$ and $\mathbf{B}$. Figure 2 illustrates our policy network π. Since Eq. (17) and the iteration of Eq. (18) are differentiable, we can backpropagate the rewards through our policy network in Eq. (20) to update parameters $\mathbf{\Psi}$.

We estimate the parameters of the policy network $\mathbf{\Psi}$ using the policy gradient method [28]. Let

$$r_t(\mathbf{y}_t, \mathbf{u}_t) = -\sum_{\tau=t}^{T} \gamma^{\tau-t} E(\mathbf{y}_\tau, \mathbf{u}_\tau) \tag{21}$$

be the cumulative discounted reward at timestep t with discount factor γ. With the reinforcement learning framework, the objective function to be maximized is the expected cumulative discounted reward,

$$J(\mathbf{\Psi}) = \sum_{t=1}^{T} \mathbb{E}_{\mathbf{y}_t, \mathbf{u}_t}[r_t(\mathbf{y}_t, \mathbf{u}_t)], \tag{22}$$

where $\mathbb{E}$ is the expectation when measurement $\mathbf{y}_t$ is generated from the dynamical system in Eq. (4), and control $\mathbf{u}_t$ is generated from the following policy distribution,

$$p(\mathbf{u}_t|\mathbf{y}_t; \mathbf{\Psi}) = \mathcal{N}(\pi(\mathbf{y}_t; \mathbf{\Psi}), \sigma^2 \mathbf{I}). \tag{23}$$

Here, $\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$ is the normal distribution with mean $\boldsymbol{\mu}$ and covariance $\mathbf{\Sigma}$. With the policy gradient theorem, the gradient of objective function J with respect to parameters $\mathbf{\Psi}$ is given by

$$\frac{\partial J(\mathbf{\Psi})}{\partial \mathbf{\Psi}} = \sum_{t=1}^{T} \mathbb{E}_{\mathbf{y}_t, \mathbf{u}_t} \left[r_t(\mathbf{y}_t, \mathbf{u}_t) \frac{\partial \log p(\mathbf{u}_t|\mathbf{y}_t; \mathbf{\Psi})}{\partial \mathbf{\Psi}} \right]. \tag{24}$$

The training procedure is shown in Algorithm 1. The expectation in Eqs. (22,24) is approximated by the Monte Carlo method using sampled measurement vectors, control vectors, and cumulative discount rewards. We use the average of the discounted sum of the future rewards as a baseline to reduce the variance [30], [4].

Algorithm 1 Training procedure of our policy network.

Input: Black-box dynamical system, target $\mathbf{y}^*$, control cost $\mathbf{R}$, number of iterations to solve the Riccati equation L, control variance σ^2, discount factor γ.
Output: Trained parameters $\mathbf{\Psi}$.

```
Pretrain parameters Ψ by minimizing H in Eq. (8) using sequences of measurement and control vectors D.
while End condition is satisfied do
    Initialize the solution of the Riccati equation P = Q.
    for ℓ ∈ {1, ··· , L} do
        Iterate Eq. (18) to solve the Riccati equation.
    end for
    Obtain optimal gain matrix F by Eq. (17).
    Randomly sample initial measurement vector y_1.
    for t ∈ {1, ··· , T} do
        Calculate the mean of the control distribution π(y_t; Ψ) by Eq. (20).
        Sample control u_t according to the control distribution in Eq. (23).
        Calculate netagive reward E(y_t, u_t) by Eq. (19).
        Obtain next measurement vector y_{t+1} from the system with sampled control u_t.
    end for
    for t ∈ {1, ··· , T} do
        Calculate cumulative discounted rewards r_t(y_t, u_t) by Eq. (21).
    end for
    Calculate J in Eq. (22) and its gradient in Eq. (24) using sampled measurement vectors y_t, control vectors u_t, and cumulative discount rewards r_t for t ∈ {1, ··· , T}.
    Update model parameters Ψ using J and its gradient using a stochastic gradient method.
end while
```

V. Experiments

A. Dynamical systems

We evaluated the proposed method using two nonlinear dynamical systems: FitzHugh-Nagumo and Lorenz equations. The FitzHugh-Nagumo equation [3], [25] is a model of an excitable system such as a neuron. We added scalar control variable u_1 to the FitzHugh-Nagumo equation as follows,

$$\frac{dx_1}{dt} = x_1 - \frac{x_1^3}{3} - x_2 + I, \quad \frac{dx_2}{dt} = c(x_1 - a - bx_2) + u_1, \tag{25}$$

where we used $a = 0.7$, $b = 0.8$, $c = 0.08$, and $I = 0.8$. The control variable was bounded a continuous value $[-2.5, 2.5]$. The Lorenz equation [19] is a chaotic nonlinear system, which was derived as as a model for atmospheric convection. We added scalar control variable u_1 to the Lorenz equation as follows,

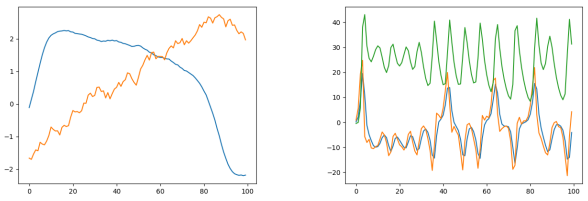
$$\begin{aligned} &\frac{dx_1}{dt} = \sigma(x_2 - x_1), \quad \frac{dx_2}{dt} = x_1(\rho - x_3) - x_2 + u_1, \\ &\frac{dx_3}{dt} = x_1 x_2 - \beta x_3, \end{aligned} \tag{26}$$

where we used $\sigma = 10$, $\rho = 28$, and $\beta = 8/3$. The control variable was bounded a continuous value $[-50, 50]$.

Figure 3 shows examples of measurement vector sequences with random control. For both systems, measurement vectors were obtained by $\mathbf{y}_t = \mathbf{x}_t + \epsilon$, where ϵ was Gaussian noise with mean $\mathbf{0}$ and standard deviation 10^{-2}. For the cost function in Eq. (5), we used zero target measurement vector $\mathbf{y}^* = \mathbf{0}$, control cost $\mathbf{R} = 10^{-2}$, and timesteps $T = 100$.

B. Proposed method setting

We used four-layered feed-forward neural networks with 128 hidden units, ReLU activation, and Koopman space $K = 64$ for encoder ϕ and decoder ψ. We pretrained parameters $\mathbf{\Psi}$ as well as parameters of decoder ψ by minimizing H in Eq. (8) with regularization hyperparameter $\eta = 1$ using 100 sequences of measurement and control vectors $\mathcal{D}$ with random initialization and random control as described in Section IV-B. For the optimization, we used Adam [11] with learning rate 10^{-3}, batch size one, and 1,000 training epochs. Then, parameters $\mathbf{\Psi}$ were tuned by maximizing J in Eq. (22) using the policy gradient method as described in Section IV-D. The maximum number of iterations for solving the Riccati



(a) FitzHugh-Nagumo (b) Lorenz

Fig. 3. Measurement vector sequences with random control.

equation in Eq. (18) was 20. The discount factor in Eq (21) was $\gamma = 0.99$. The variance of control σ^2 in Eq. (23) was set to half of the bounded width of the control variable, i.e., 2.5 for FitzHugh-Nagumo, and 50 for Lorenz equations. For the optimization, we used Adam with learning rate 10^{-3}, batch size 10, and maximum number of iterations 5,000. 50 simulation runs were evaluated for each iteration for early stopping.

C. Comparing methods

We compared the proposed method with the following five methods: NDMD, NDMD+C, NDMD+D, DMD, DMD+RL, and RL, which were based on the Koopman operator theory and/or reinforcement learning. The comparing methods were trained in the same way with the proposed method.

NDMD is neural dynamic mode decomposition [29], where system identification in the Koopman space was conducted as in Section IV-B assuming the neural network-based encoder and decoder. The control sequences were obtained by an LQR as in Section IV-C. NDMD corresponds to the proposed method without reinforcement learning procedures in Section IV-D.

NDMD+C is NDMD with a regularizer to improve controllability as [7]. A linear dynamical system is controllable if controllability matrix $\mathbf{C} = [\mathbf{B} \ \mathbf{AB} \ \mathbf{A}^2\mathbf{B} \ \cdots \ \mathbf{A}^{K-1}\mathbf{B}]$ has full row rank. Since the number of non-zero singular values is equal to the rank of the matrix, we added regularization term $\frac{1}{K}\sum_{k=1}^{K}\frac{1}{\tilde{s}_k}$ to H in Eq. (8), where $\tilde{s}_k = \frac{s_k}{\sum_{k'=1}^{K} s_{k'}}$ is the normalized singular value, and s_k is the singular value of controllability matrix $\mathbf{C}$.

NDMD+D is NDMD with a regularizer to improve cost function approximation as [17] by making distances in the Koopman space close to those in the measurement space. We added regularization term $\sum_{n=1}^{N}\sum_{t=1}^{T}\sum_{t'=1}^{T} \| \| \mathbf{y}_{nt} - \mathbf{y}_{nt'} \|^2 - \| \phi(\mathbf{y}_{nt}) - \phi(\mathbf{y}_{nt'}) \|^2 \|^2$ to H in Eq. (8).

DMD is dynamic mode decomposition, where system identification in the Koopman space was conducted without neural network-based encoders and decoders assuming the Koopman space was the same with the measurement space $\mathbf{g}_t = \mathbf{y}_t$. The control sequences were obtained by an LQR.

DMD+RL is the DMD with reinforcement learning procedures in Section IV-D. DMD+RL corresponds to the proposed method without encoders and decoders.

TABLE I

AVERAGE TEST COSTS AND THEIR STANDARD ERRORS.

	FitzHugh-Nagumo	Lorenz
Ours	**83.1 ± 2.9**	**5151.8 ± 786.8**
NDMD	443.9 ± 90.6	83152.3 ± 521.2
NDMD+C	343.2 ± 4.9	86526.7 ± 24.0
NDMD+D	940.3 ± 293.4	84631.8 ± 1171.8
DMD	312.8 ± 1.5	73073.3 ± 1982.8
DMD+RL	211.7 ± 15.3	37590.1 ± 3059.8
RL	85.7 ± 2.6	24310.1 ± 4131.8

RL is reinforcement learning using the policy gradient method with a policy of a feed-forward neural network. The policy network takes a measurement vector as input, and outputs a control vector. We used a four-layered feed-forward neural network with 128 hidden units, and ReLU activation. RL corresponds to the proposed method with a black-box policy that considers neither Koopman spaces nor LQRs.

D. Results

Table I shows test costs averaged over ten experiments. Figure 4 show examples of controlled dynamics by the proposed method, NDMD, and RL. The proposed method achieved the lowest test cost with both systems. The test cost of NDMD was high because system identification of nonlinear dynamical systems was difficult, and the cost in the Koopman space could not approximate that in the measurement space appropriately. Even with regularizers (NDMD+C/D), the performance was not improved well. On the other hand, the proposed method directly minimized the cost in the measurement space with the reinforcement learning framework. The test cost of the DMD was high because it could not model the nonlinear dynamical systems. By incorporating the reinforcement learning framework with DMD+RL, the performance was improved compared with DMD. However, the performance of DMD+RL was worse than that of the proposed method. This result indicates that it is important to use nonlinear transformation by encoders to the Koopman space for controlling nonlinear dynamical systems. RL achieved better performance than the other comparing methods due to its direct minimization of the cost in the measurement space, and the high representation power of neural networks. With RL, the policy network was fully modeled by a neural network as a black-box function. In contrast, with the proposed method, the policy network was modeled by a neural network incorporating the knowledge on the Koopman operator theory and linear quadratic regulators. Therefore, the proposed method obtained better controllers than RL.

VI. CONCLUSION

We proposed a method to learn a controller for nonlinear dynamical systems. With the proposed method, a policy network is modeled based on a linear quadratic regulator in a Koopman space, and it is trained by minimizing the expected cost with reinforcement learning. Although our results are encouraging, we must extend our approach in several future directions. First, we will apply it to other

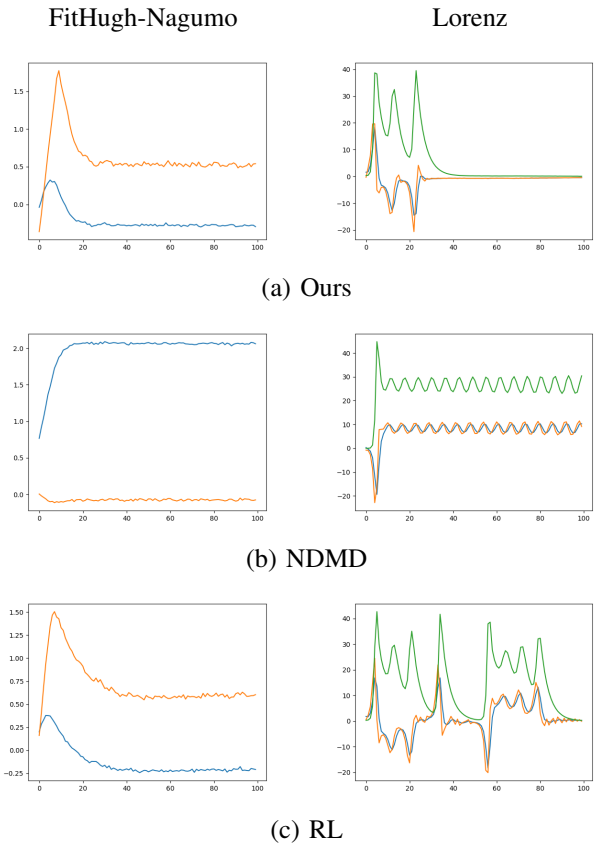


Fig. 4. Examples of controlled dynamics of (Left) FitzHugh-Nagumo and (Right) Lorenz equations by (a) our method, (b) NDMD, and (c) RL.

nonlinear dynamical systems, such as fluid and robots. Second, we plan to improve our proposed method using reinforcement learning techniques, which includes actor-critic methods [5]. Third, we will incorporate other control methods for linear dynamical systems than linear quadratic regulators in our framework based on the Koopman operator theory and reinforcement learning.

Acknowledgement

This work was partially supported by JST CREST Grant No. JPMJCR1913.

References

[1] O. Azencot, N. B. Erichson, V. Lin, and M. W. Mahoney. Forecasting sequential data using consistent Koopman autoencoders. In *International Conference on Machine Learning*, 2020.

[2] D. Bruder, B. Gillespie, C. D. Remy, and R. Vasudevan. Modeling and control of soft robots using the Koopman operator and model predictive control. In *Robotics: Science and Systems XV*, 2019.

[3] R. FitzHugh. Mathematical models of threshold phenomena in the nerve membrane. *The Bulletin of Mathematical Biophysics*, 17(4):257–278, 1955.

[4] E. Greensmith, P. L. Bartlett, and J. Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5:1471–1530, 2004.

[5] I. Grondman, L. Busoniu, G. A. Lopes, and R. Babuska. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 42(6):1291–1307, 2012.

[6] S. J. Guastello. *Chaos, catastrophe, and human affairs: Applications of nonlinear dynamics to work, organizations, and social evolution*. Psychology Press, 2013.

[7] Y. Han, W. Hao, and U. Vaidya. Deep learning of Koopman representation for control. In *IEEE Conference on Decision and Control*, pages 1890–1895, 2020.

[8] B. Huang, X. Ma, and U. Vaidya. Feedback stabilization using Koopman operator. In *IEEE Conference on Decision and Control*, pages 6434–6439, 2018.

[9] T. Iwata and Y. Kawahara. Neural dynamic mode decomposition for end-to-end modeling of nonlinear dynamics. *arXiv preprint arXiv:2012.06191*, 2020.

[10] E. Kaiser, J. N. Kutz, and S. L. Brunton. Data-driven discovery of Koopman eigenfunctions for control. *arXiv preprint arXiv:1707.01146*, 2017.

[11] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[12] D. E. Kirk. *Optimal control theory: an introduction*. Courier Corporation, 2004.

[13] B. O. Koopman. Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences of the United States of America*, 17(5):315–318, 1931.

[14] M. Korda and I. Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160, 2018.

[15] H. Kwakernaak and R. Sivan. *Linear optimal control systems*, volume 1. Wiley-interscience New York, 1972.

[16] K. Lee and K. T. Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404:108973, 2020.

[17] Y. Li, H. He, J. Wu, D. Katabi, and A. Torralba. Learning compositional Koopman operators for model-based control. In *International Conference on Learning Representations*, 2019.

[18] W.-m. Liu, H. W. Hethcote, and S. A. Levin. Dynamical behavior of epidemiological models with nonlinear incidence rates. *Journal of Mathematical Biology*, 25(4):359–380, 1987.

[19] E. N. Lorenz. Deterministic nonperiodic flow. *Journal of Atmospheric Aciences*, 20(2):130–141, 1963.

[20] B. Lusch, J. N. Kutz, and S. L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):1–10, 2018.

[21] X. Ma, B. Huang, and U. Vaidya. Optimal quadratic regulation of nonlinear system using Koopman operator. In *American Control Conference*, pages 4911–4916, 2019.

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

[23] I. Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1-3):309–325, 2005.

[24] J. Morton, A. Jameson, M. J. Kochenderfer, and F. Witherden. Deep dynamical modeling and control of unsteady fluid flows. In *Advances in Neural Information Processing Systems*, pages 9258–9268, 2018.

[25] J. Nagumo, S. Arimoto, and S. Yoshizawa. An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10):2061–2070, 1962.

[26] J. L. Proctor, S. L. Brunton, and J. N. Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, 2016.

[27] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[28] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, 12:1057–1063, 1999.

[29] N. Takeishi, Y. Kawahara, and T. Yairi. Learning Koopman invariant subspaces for dynamic mode decomposition. In *Advances in Neural Information Processing Systems*, pages 1130–1140, 2017.

[30] L. Weaver and N. Tao. The optimal reward baseline for gradient-based reinforcement learning. In *Conference on Uncertainty in Artificial Intelligence*, pages 538–545, 2001.

[31] E. Yeung, S. Kundu, and N. Hodas. Learning deep neural network representations for Koopman operators of nonlinear dynamical systems. In *American Control Conference*, pages 4832–4839, 2019.

[32] P. You, J. Pang, and E. Yeung. Deep Koopman controller synthesis for cyber-resilient market-based frequency regulation. *IFAC-PapersOnLine*, 51(28):720–725, 2018.