Proceedings of the 2025 SICE Festival with Annual Conference
September 9-12, 2025, Chiang Mai, Thailand

Koopman-based Reinforcement Learning using Kernel Trick for Discrete-Time Nonlinear Systems with Noise

Ritsuki Nakahara† and Tomonori Sadamoto

Department of Mechanical Engineering and Intelligent Systems, Graduate School of Informatics and Engineering, The University of Electro-Communications, Tokyo, Japan
(E-mail: {r.nakahara, sadamoto}@uec.ac.jp)

Abstract: In this paper, we propose a reinforcement learning (RL) method using Koopman operator and Kernel Trick for discrete-time nonlinear systems with process and observation noise. This method applies a type of reinforcement learning technique to the linear system realized in a high-dimensional space by using the Koopman operator and the Kernel Trick without system identification. The effectiveness of the proposed method is investigated through a duffing oscillator under process and observation noise.

Keywords: Optimal Control, Data-driven Control, Reinforcement Learning, Koopman Operator, Kernel Trick

1. INTRODUCTION

In recent years, systems targeted by control engineering have become increasingly large-scale and complex. Due to this large-scale complexity, it is difficult to accurately identify their mathematical models. Because of this background, the importance of data-driven control methods, which design controllers based on only input-output data without fully understanding the internal structure and dynamic characteristics of systems, has been increasing. Also, since most real systems are nonlinear and contain noise, it is also necessary to address these issues.

One of the global linearization methods for nonlinear systems is the Koopman operator-based method (Koopman linearization) [1]. Systems globally linearized by the Koopman operator can be identified using only the input-output data of the target system and basis functions. There are primal and dual problems in this identification problem [1], and in the primal problem, the number of given basis functions affects the computational complexity of the algorithm. In the dual problem, the inner product calculation between states lifted to higher dimensions by basis functions is expressed by kernel functions. This method of calculating with kernel functions instead of inner product calculations is called the Kernel Trick [2], and the computational complexity of the dual problem algorithm using the Kernel Trick depends on the amount of data. Although data-driven controller design methods using this advantage have already been proposed [3], it is not yet clear for systems containing noise. Therefore, this research derives the dual problem corresponding to the primal problem of reinforcement learning using the Koopman operator for discrete-time nonlinear systems with noise, and clarifies the reinforcement learning method in the dual problem.

The structure of this paper is as follows. Section 2 shows the problem setting and the objectives of this paper. Section 3 derives the primal problem of Koopman-type reinforcement learning in discrete-time nonlinear systems with noise. Section 4 proposes a reinforcement learning method using the Kernel Trick for the dual problem corresponding to the primal problem of Koopman-type reinforcement learning. Section 5 verifies the effectiveness of the proposed method through numerical simulations, and Section 6 describes conclusions and future prospects.

Notation

The following notations are used in this paper.

$a_k \sim \mathcal{D}_W$	$\mathbb{E}[a_k] = 0$ $\mathbb{E}\left[a_k a_j^\top\right] = \begin{cases} W & (j = k) \\ 0 & (j \neq k) \end{cases}$
$\left[\varphi_j(\cdot)\right]_{j=1}^{j=n_z}$	$\left[\varphi_1(\cdot)^\top, \ldots, \varphi_{n_z}(\cdot)^\top\right]^\top$
$\mathrm{diag}(A_1, \ldots, A_n)$	A block diagonal matrix with $A_1, \ldots, A_n$ on the block diagonal
A^i	Matrix at the i-th iteration
$\mathrm{vec(A)}$	$\left[a_1^\top, \ldots, a_n^\top\right]^\top$ where a_i is the i-th column vector of A
$\otimes$	Kronecker product
$\odot$	Hadamard product
$\mathcal{N}(\mu, \Sigma)$	Normal distribution with mean μ and variance Σ

2. PROBLEM FORMULATION

Let us consider a discrete-time nonlinear system, described as

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, w_k) \\ y_k &= x_k + v_k, \end{aligned} \tag{1}$$

where $k \geq 0$ is the time step, $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^r \to \mathbb{R}^n$ is a locally Lipschitz mapping with $f(0, 0, 0) = 0$. Also, $x_k \in \mathbb{R}^n$ is the state, $y_k \in \mathbb{R}^n$ is the output, $u_k \in \mathbb{R}^m$ is the input, $w_k \in \mathbb{R}^r$ is the process noise, $v_k \in \mathbb{R}^n$ is the observation noise, where $w_k \sim \mathcal{D}_{W_w}$ and $v_k \sim \mathcal{D}_{W_v}$. In this paper, we make the following assumptions for the system in (1).

Assumption 1: The function f in (1) is unknown.

† Ritsuki Nakahara is the presenter of this paper.

Assumption 2: In (1), the state x_k, process noise w_k, and observation noise v_k are unmeasurable, while the output y_k and input u_k are measurable.
Furthermore, the control input is determined by a deterministic policy

$$u_k = \pi(y_k), \tag{2}$$

which is designed to minimize the evaluation function

$$J(y_0;\pi) := \mathbb{E}\left[\sum_{k=0}^{\infty} r(y_k,\pi(y_k)) - \lambda^{\pi}\middle| y_0\right] \tag{3}$$

as much as possible. Here,

$$r(y_k u_k) := y_k^{\top} Q y_k + u_k^{\top} R u_k \tag{4}$$

$$\lambda^{\pi} := \lim_{N\to\infty} \frac{1}{N}\mathbb{E}\left[\sum_{k=1}^{N} r(y_k,\pi(y_k))\right] \tag{5}$$

where $Q \geq 0$ and $R > 0$ are given, and λ^{π} represents the average cost under the policy $\pi(y_k)$. In this research, we extend the method proposed in reference [4] to nonlinear systems using Koopman linearization. The details of this extension method will be discussed in Section 4, and in the next section, we first briefly describe the overview of the Koopman linearization method adopted in this paper.

3. KOOPMAN LINERIZATION

In this section, based on [5], we provide several assumptions to linearize the discrete-time nonlinear system in (1) and apply the proposed method. First, when globally linearizing the system in (1) using the Koopman operator, we choose basis functions that are smooth independent of u and w, and have linearity with respect to u and w. That is, for $j = 1, 2, \ldots$, we define basis functions $\varphi_j : \mathbb{R}^n \to \mathbb{R}$ as follows

$$z_k(y_k) := [y_k^{\top}, \varphi_1(y_k)^{\top}, \ldots, \varphi_{n_z}(y_k)^{\top}]^{\top}. \tag{6}$$

Note that $1 \ll n_z$. Hereafter, we denote $z_k(y_k)$ as z_k for simplifying the notation. Using the output equation of (1), we can write

$$z_k = \begin{bmatrix} x_k \\ [\varphi_j(x_k)]_{j=1}^{j=n_z} \end{bmatrix} + \begin{bmatrix} v_k \\ [\varphi_j(y_k)]_{j=1}^{j=n_z} - [\varphi_j(x_k)]_{j=1}^{j=n_z} \end{bmatrix}. \tag{7}$$

Here, we make the following assumptions for (1) and (7)

Assumption 3: For any $x_k \in \mathcal{X}$ and $v_k \sim \mathcal{D}_{W_v}$, assume that

$$[v_k^{\top}, (\varphi(x_k + v_k) - \varphi(x_k))^{\top}]^{\top} =: l_k \sim \mathcal{D}_{W_l} \tag{8}$$

holds.

Assumption 4: Under Assumption 3, the system in (1) globally linearized by the Koopman operator becomes

$$\begin{aligned} s_{k+1} &= A s_k + B u_k + C w_k \\ z_k &= s_k + l_k \end{aligned} \tag{9}$$

where $N_z := n + n_z$, $A \in \mathbb{R}^{N_z \times N_z}$, $B \in \mathbb{R}^{N_z \times m}$, $C \in \mathbb{R}^{N_z \times r}$, and s_k is

$$s_k := \begin{bmatrix} x_k \\ [\varphi_j(x_k)]_{j=1}^{j=n_z} \end{bmatrix}.$$

For the system in (9), we make the following assumption

Assumption 5: (A, B) is stabilizable.

Under these assumptions, with

$$\hat{Q} := \mathrm{diag}(Q, 0, \ldots, 0) \tag{10}$$

(4) can be expressed as

$$r(y_k, u_k) = z_k^{\top} \hat{Q} z_k + u_k^{\top} R u_k \tag{11}$$

which is in quadratic form. Therefore, (9) and (11) are equivalent to the LQR problem formulation when the control law is linear. In this case, the method from [4] can be used. This will be shown in the next section.

4. KOOPMAN TYPE REINFORCEMENT LEARNING FOR PRIMARY PROBLEM

4.1. Preliminaries

In this section, we present an overview of Koopman-based Average Off-PI for the primal problem, along with related theorems. This algorithm starts with a stable initial gain K^1, then repeats the calculation of average cost and Policy Iteration until satisfy $i = N$ or $\|K^{i+1} - K^i\| < \epsilon$, finally returning K^i and terminating. Next, we define the policy $\pi(y_k)$ in (2) as follows

$$\pi(y_k) := K z_k \tag{12}$$

Note that z_k depends on y_k. Since (9) is a linear system, based on [4], we provide the following lemma

Lemma 1: [4] Consider the system in (1), and assume that assumptions 1-5 hold. Let P^{π} satisfy

$$L^{\top} P^{\pi} L - P^{\pi} + K^{\top} R K + \hat{Q} = 0 \tag{13}$$

where $P^{\pi} \geq 0$, $L := A + BK$ and is stable. Then, considering λ^{π} in (5), with the value function defined as

$$V^{\pi}(y_k) := \mathbb{E}\left[\sum_{k'=k}^{\infty} r(y_{k'}, \pi(y_{k'})) - \lambda^{\pi}\middle| y_k\right] \tag{14}$$

we have

$$V^{\pi}(y_k) = z_k^{\top} P^{\pi} z_k \tag{15}$$

Furthermore, λ^{π} in (5) can be written as

$$\begin{aligned} \lambda^{\pi} =& \,\mathrm{tr}(C^{\top} P^{\pi} C W_w) + \mathrm{tr}(P^{\pi} W_l) \\ &+ \mathrm{tr}(K^{\top} B^{\top} P^{\pi} B K W_l) - \mathrm{tr}(L^{i\top} P^{\pi} L W_l) \end{aligned} \tag{16}$$

4.2. Calculation of Average Cost

In this section, we perform the calculation of (16). Since (9) is unknown, we cannot directly use the results obtained in Lemma 1. Therefore, we use

$$\bar{\lambda}^i = \frac{1}{\tau}\sum_{k=1}^{\tau} r_k^i \tag{17}$$

as the empirical average cost $\bar{\lambda}^i$. Using this $\bar{\lambda}^i$, we perform Policy Iteration in the next section.

4.3. Koopman-based Average Off-PI

In this section, we show Policy Iteration based on Section 4.2. While we have been considering the target policy $\pi(y_k) = Kz_k$, here we define a behavior policy $u_k = Kz_k + e_k$ with exploration noise and consider a system using this behavior policy. Note that $e_k \sim \mathcal{D}_{W_e}$. In this case, the random variables in the closed-loop system are $y_k, x_k, w_k, v_k, \eta_k$, and we denote the expectation with respect to $\mathcal{X} := \{y_k, x_k, w_k, v_k\}$ as $\mathbb{E}_{\mathcal{X}}$. From the system in (9), we obtain

$$\Theta^i \zeta^i = \Xi^i - \sigma_k \tag{18}$$

Here's the next part of the translation where

$$\Theta^i := \begin{bmatrix} D^i_{zz} - D^i_{zpzp} & 2D^i_{ze} & 2D^i_{eu} - D^i_{ee} \end{bmatrix} \tag{19}$$

$$\zeta^i := \begin{bmatrix} \mathrm{vec}(P^i)^\top & \mathrm{vec}(H^i)^\top & \mathrm{vec}(N^i)^\top \end{bmatrix}^\top \tag{20}$$

$$\Xi^i := \left[\xi_0, \ldots, \xi_{\tau'-1}\right]^\top \tag{21}$$

$$\sigma_k := -z_{k+1}^\top P^i z_{k+1} + \mathbb{E}\left[z_{k+1}^\top P^i z_{k+1} | y_k\right] \tag{22}$$

and $D^i_{zz}, D^{+i}_{zz} \in \mathbb{R}^{\tau' \times N_z^2}$, $D^i_{ze} \in \mathbb{R}^{\tau' \times mN_z}$, $D^i_{eu}, D^i_{ee} \in \mathbb{R}^{\tau' \times m^2}$, $H^i \in \mathbb{R}^{m \times N_z}$, $N^i \in \mathbb{R}^{m \times m}$, $\xi_k \in \mathbb{R}$ are

$$\begin{aligned}
D^i_{zz} &:= \left[z_0 \otimes z_0, \ldots, z_{\tau'-1} \otimes z_{\tau'-1}\right]^\top \\
D^i_{zpzp} &:= \left[z_1 \otimes z_1, \ldots, z_{\tau'} \otimes z_{\tau'}\right]^\top \\
D^i_{ze} &:= \left[z_0 \otimes e_0, \ldots, z_{\tau'-1} \otimes e_{\tau'-1}\right]^\top \\
D^i_{eu} &:= \left[e_0 \otimes u_0, \ldots, e_{\tau'-1} \otimes u_{\tau'-1}\right]^\top \\
D^i_{ee} &:= \left[e_0 \otimes e_0, \ldots, e_{\tau'-1} \otimes e_{\tau'-1}\right]^\top \\
H^i &:= B^\top P^i A \\
N^i &:= B^\top P^i B \\
\xi_k &:= y_k^\top Q y_k + (u_k - e_k)^\top R (u_k - e_k) - \bar{\lambda}^i
\end{aligned}$$

By solving (18) to obtain H^i, N^i in this way, we derive the control law as

$$K^{i+1} = -\left(\sum_{i'=0}^{i} (\hat{N}^{i'} + R)\right)^{-1} \left(\sum_{i'=0}^{i} \hat{H}^{i'}\right) \tag{23}$$

At the end of this section, by introducing an L_2 regularization term to this minimization problem, we get

$$\min_{\zeta^i} \frac{1}{2}\|\Theta^i \zeta^i - \Xi^i\|^2 + \frac{\alpha}{2}\|\zeta^i\|^2 \tag{24}$$

In this paper, we call the above equation the primal problem, and we refer to the procedure of obtaining a controller from (23) by solving (24) instead of (18) as Koopman-type reinforcement learning for the primal problem. Since the dimension of ζ depends on the dimension N_z of the state z composed of basis functions, the more bases we prepare to compensate for nonlinearity (increasing N_z), the more difficult it becomes to execute this algorithm for primal problem in real-time.

5. KOOPMAN TYPE REINFORCEMENT LEARNING FOR DUAL PROBLEM

In this section, we propose a method that is equivalent to the algorithm for primal problem and can utilize a sufficient number of basis functions. In the dual problem, since states z_k composed of basis functions are expressed through inner product calculations between outputs, the inner products can be replaced with kernel functions. Therefore, while executing the algorithm is difficult in the primal problem when given a vast number of basis functions, the dual problem algorithm can be executed as it depends on the amount of data. To obtain this advantage, we will derive the dual problem algorithm. First, the dual problem corresponding to (24) is as follows

$$\begin{aligned}
&\min_{\mathbf{a}^i} \frac{1}{2}\|\Theta^i \Theta^{i^\top} \mathbf{a}^i - \Xi^i\|^2 + \frac{\alpha}{2}\|\Theta^{i^\top} \mathbf{a}^i\|^2 \\
&\text{subject to} \quad \Theta^{i^\top} \mathbf{a}^i = \zeta^i
\end{aligned} \tag{25}$$

where $\mathbf{a}^i \in \mathbb{R}^{\tau'}$. The solution is given by

$$\mathbf{a}^i = \left(\Theta^i \Theta^{i^\top} + \alpha I\right)^{-1} \Xi^i \tag{26}$$

From (19) and (20), the above equation can be expressed as

$$\mathrm{vec}(P^i) = (D^i_{zpzp} - D^i_{zz})\mathbf{a}^i \tag{27}$$

$$\mathrm{vec}(H^i) = 2D^i_{ze}\mathbf{a}^i \tag{28}$$

$$\mathrm{vec}(N^i) = (2D^i_{eu} - D^i_{ee})\mathbf{a}^i \tag{29}$$

Here, $\Theta^i \Theta^{i\top}$ can be expanded as

$$\Theta^i \Theta^{i\top} \tag{30}$$

$$\begin{aligned}
=&D^i_{zz}(D^i_{zz})^\top - D^i_{zz}(D^i_{zpzp})^\top - D^i_{zpzp}(D^i_{zz})^\top \\
&+ D^i_{zpzp}(D^i_{zpzp})^\top + 4D^i_{ze}(D^i_{ze})^\top + 4D^i_{eu}(D^i_{eu})^\top \\
&- 2D^i_{eu}(D^i_{ee})^\top - 2D^i_{ee}(D^i_{eu})^\top + 4D^i_{ee}(D^i_{ee})^\top
\end{aligned} \tag{31}$$

where

$$D^i_{zz}(D^i_{zz})^\top = (Z^+ Z^\top) \odot (ZZ^\top) \tag{32}$$

$$D^i_{zpzp}(D^i_{zz})^\top = (Z^+ Z^\top) \odot (Z^+ Z^\top) \tag{33}$$

$$D^i_{zz}(D^i_{zpzp})^\top = (Z(Z^+)^\top) \odot (Z(Z^+)^\top) \tag{34}$$

$$D^i_{zpzp}(D^i_{zpzp})^\top = (Z^+(Z^+)^\top) \odot (Z^+(Z^+)^\top) \tag{35}$$

$$D^i_{ze}(D^i_{ze})^\top = (ZZ^\top) \odot (EE^\top) \tag{36}$$

$$D^i_{eu}(D^i_{eu})^\top = (EE^\top) \odot (UU^\top) \tag{37}$$

$$D^i_{eu}(D^i_{ee})^\top = (EU^\top) \odot (EE^\top) \tag{38}$$

$$D^i_{ee}(D^i_{eu})^\top = (EE^\top) \odot (EU^\top) \tag{39}$$

$$D^i_{ee}(D^i_{ee})^\top = (EE^\top) \odot (EE^\top) \tag{40}$$

and Z, Z^+, U, E are defined as

$$Z := [z_0, \ldots, z_{\tau'-1}]^\top \in \mathbb{R}^{\tau' \times N_z} \tag{41}$$

$$Z^+ := [z_1, \ldots, z_{\tau'}]^\top \in \mathbb{R}^{\tau' \times N_z} \tag{42}$$

$$U := [u_0, \ldots, u_{\tau'-1}]^\top \in \mathbb{R}^{\tau' \times m} \tag{43}$$

$$E := [e_0, \ldots, e_{\tau'-1}]^\top \in \mathbb{R}^{\tau' \times m} \tag{44}$$

Each element of $ZZ^\top, Z^+Z^\top, Z^+(Z^+)^\top, H^i z_k$ is an inner product between vectors projected onto a high-dimensional space by basis functions. Then, we define a positive semidefinite kernel function $\mathcal{K} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ and replace each element with kernel functions. $\mathcal{K}$ is given as the sum of a linear kernel function and any positive semidefinite kernel function K_φ, namely

$$\mathcal{K}(y, y') = y^\top y' + K_\varphi(y, y')$$

Finally, the control law can be obtained from the following equation

$$u_k^{i+1} = \left(\sum_{i'=1}^{i} \left(R + \mathrm{vec}^{-1}((2D_{eu}^i - D_{ee}^i)\mathbf{a}^{i'}) \right) \right)^{-1} \cdot \left(\sum_{i'=1}^{i} [\mathcal{K}(y_k, y_0) \otimes e_0, \ldots, \mathcal{K}(y_k, y_{\tau'-1}) \otimes e_{\tau'-1}]\, \mathbf{a}^{i'} \right) \tag{45}$$

Its pseudocode is given below

Algorithm 1 : Koopman-based Off-Policy Iteration for Dual Problem using Kernel Trick

1: **Initialization:** Set $i \leftarrow 1$ and choose a stable gain K^1. Also, provide a positive semidefinite kernel function $\mathcal{K}$ and a small termination threshold $\epsilon > 0$.
2: **Average Cost Calculation:** Perform sampling for τ steps and use the obtained data to calculate the average cost $\bar{\lambda}^i$ from (17).
3: **Policy Evaluation and Improvement:** Apply the behavior policy after τ'' steps, and acquire the output y, input u and exploration noise e data before and after its application. Repeat this τ' times, and from the τ' pieces of data obtained, determine $\mathbf{a}^i$ from (26). Calculate the input u^{i+1} for the next iteration from (45).
4: **Convergence Check:** If $\|\mathbf{a}^{i+1} - \mathbf{a}^i\| \leq \epsilon$ is satisfied, terminate. Otherwise, set $i \leftarrow i+1$ and return to step 2.

6. NUMERICAL RESULT

In this section, we verify the effectiveness of the proposed **Algorithm 1**. The control target system is a discretized Duffing oscillator [1] with the equilibrium point $(1, 0)$ shifted to the origin. Given as follows

$$\begin{aligned} \begin{bmatrix} p_{k+1} \\ q_{k+1} \end{bmatrix} &= \begin{bmatrix} p_k + \delta t q_k \\ q_k + \delta t(-p_k(p_k+1)(p_k+2) - 0.5q_k + u_k) \end{bmatrix} \\ &\quad + \delta t w_k \\ y_k &= x_k + v_k \end{aligned}$$

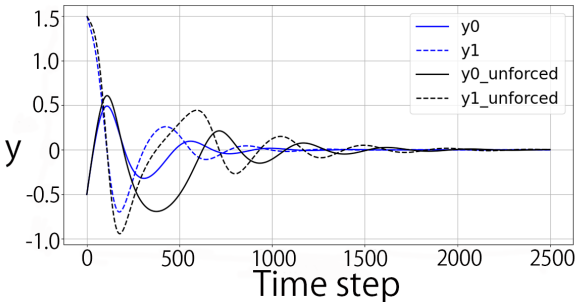

Fig. 1 The black line does the free response, and the solid blue line does the closed-loop response using the controller designed via **Algorithm 1**.

where $x_k = [p_k \; q_k]^\top \in \mathbb{R}^2$ and w_k and v_k follows $\mathcal{N}(0, 1)$. $Q \in \mathbb{R}^{2\times 2}$ and $R \in \mathbb{R}$ are given by

$$Q = \mathrm{diag}(1, 1), \quad R = 1$$

The initial state is set to $x_0 = [-0.5 \; 1.5]^\top$, iteration threshold $\epsilon = 0.0001$, maximum iteration count $N = 5$, initial gain $K^1 = 0$, exploration noise $e \sim \mathcal{N}(0, 0.0001)$, regularization parameter $\alpha = 1$, and sampling numbers are set to $\tau = 2000$, $\tau' = 2000$, with sampling interval $\tau'' = 10$. The kernel function $\mathcal{K}(y, y')$ is given by

$$\mathcal{K}(y, y') = y^\top y' + y^\top y' \exp(\|y - y'\|^2).$$

Under these settings, we obtained the control law using **Algorithm 1**, and the control results are shown in Fig. 1. The obtained control law asymptotically stabilizes the system to the origin, and the control results show faster convergence compared to the free response.

7. CONCLUSION

In this paper, we proposed a Koopman-type reinforcement learning method using Kernel Trick. Since the theoretical analysis of the proposed method is not sufficient, future work includes conducting a thorough theoretical analysis of the proposed method.

REFERENCES

[1] A. Mauroy, Y. Susuki, I. Mezić, *Koopman operator in systems and control*, Springer, 2020.

[2] M. Seeger, "Gaussian processes for machine learning", *International journal of neural systems*, Vol.14, No.02, pp.69–106, 2004.

[3] A. Kikuya, T. Sadamoto, "Koopman-based reinforcement learning using kernel trick for discrete-time input-affine nonlinear systems", in *Proc. of the Japan Joint Automatic Control Conference (in Japanese)*, pp.484–489, 2023.

[4] F. A. Yaghmaie, F. Gustafsson, "Using reinforcement learning for model-free linear quadratic control with process and measurement noises", in *Proc. of Conference on Decision and Control*, pp.6510–6517, 2019.

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