2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Stochastic Nonlinear Control via Finite-dimensional Spectral Dynamic Embedding

Tongzheng Ren[1], Zhaolin Ren[2], Na Li[2], and Bo Dai[3]

Abstract— **Optimal control is notoriously difficult for stochastic nonlinear systems. [1] introduced Spectral Dynamics Embedding for developing reinforcement learning methods for controlling an unknown system. It uses an infinite-dimensional feature to linearly represent the state-value function and exploits finite-dimensional truncation approximation for practical implementation. However, the finite-dimensional approximation properties in control have not been investigated even when the model is known. In this paper, we provide a tractable stochastic nonlinear control algorithm that exploits the nonlinear dynamics upon the finite-dimensional feature approximation, Spectral Dynamics Embedding Control (SDEC), with an in-depth theoretical analysis to characterize the *approximation error* induced by the finite-dimension truncation and *statistical error* induced by finite-sample approximation in both policy evaluation and policy optimization. We also empirically test the algorithm and compare the performance with Koopman-based methods and iLQR methods on the pendulum swingup problem.**

I. Introduction

Stochastic optimal nonlinear control—i.e. finding an optimal feedback policy to maximize cumulative rewards for a stochastic nonlinear system—has been a long-standing challenging problem in control literature [2], [3]. Various control techniques have been developed for nonlinear control, including gain scheduling [4], [5], feedback linearization [6], iterative linear-quadratic regulator [7], sliding model control [8], geometric control [9], back stepping [10], control Lyapunov functions [11], model-predictive control [12], and tools that leverage inequality approximation and optimization-based methods like sum-of-squares (SOS) programming [13]. Nonlinear control often focuses on the stability of closed-loop systems, while control optimality analysis is often heuristic or limited to special classes of systems. Moreover, almost all the methods have their own limitations, where they either lead to highly suboptimal solutions, can only be applied to a subclass of nonlinear systems satisfying special conditions, or require a large amount of computation and thus could only handle very small-scale systems.

To take advantage of the rich theory and tools developed for linear systems, kernel-based linearization has recently regained attraction. The representative approaches includes Koopman operator theory [14], [15], kernelized nonlinear regulator (KNR) [16], reproducing kernel Hilbert space (RKHS) dynamics embedding control [17], and optimal control with occupation kernel [18], among others. The Koopman operator lifts states into an infinite-dimensional space of known measurement functions, where the dynamics become linear in the new space and *separated additive* linear action effects. Alternatively, KNR [16] assumes the nonlinear dynamics lie in an infinite-dimensional RKHS, and hence are *linear* in the corresponding feature maps for states and actions. [17] represents the conditional transition probability of a Markov decision process (MDP) in a *pre-defined* RKHS so that calculations involved in solving the MDP could be done via inner products in the infinite-dimensional RKHS. [18] introduce the Liouville's equation in occupation kernel space to represent the trajectories, with which the optimal value can be reformulated as linear programming in the infinite-dimensional space under some strict assumptions.

Although kernelized linearization has brought a promising new perspective to nonlinear control, these representative approaches fall short in both computational and theoretical respects. *Computationally*, control in an infinite-dimensional space is intractable. Hence, a finite-dimensional approximation is necessary. Data-driven computational procedures for kernel selection and RKHS reparametrization have been proposed for the Koopman [15], conditional RKHS embeddings [17], and occupation kernel [18] (whereas KNR [16] focuses on the sample complexity of learning the dynamics while ignoring the computational challenges in the control aspect). However, these methods are inefficient in the sense that **i)**, the dynamics or trajectories are *presumed* to be lying in some RKHS with a *pre-defined* finitely-approximated kernel, which is a very strict assumption; in fact, finding good kernel representations for the dynamics is a challenging task; and **ii)**, the dynamics information for kernelization is *only* exploited through samples, and other structure information in the dynamics are ignored even when the dynamics formula is known explicitly. Meanwhile, *theoretically*, the optimality of control with finite-dimensional approximations—*i.e.*, the policy value gap between the finite-dimensional approximation and optimal policy in an infinite-dimensional RKHS—has largely been ignored and not been rigorously analyzed.

Recently, [1] provided a novel kernel linearization method, spectral dynamic embedding, by establishing the connection between stochastic nonlinear control models and linear Markov Decision Processes (MDPs), which exploits the random noise property to factorize the transition probability operator, and induce an infinite-dimensional space for *linearly* representing the *state-action value function* for any *arbitrary* policy. Spectral dynamic embedding bypasses the drawbacks of the existing kernel linearization methods in

[1]T. Ren is with Google Brain and University of Texas, Austin, Email: tongzheng@utexas.edu; [2] Z. Ren and N. Li are with Harvard University, Email: zhaolinren@g.harvard.edu, nali@seas.harvard.edu; [3]B. Dai is the corresponding author with Google Brain and Georgia Tech, Email: bodai@{google.com, cc.gatech.edu}.

the sense that **i)** the kernel is *automatically induced* by the system dynamics, which avoids the difficulty in deciding the kernel features and eliminates the modeling approximation induced by a predefined kernel; and **ii)** the kernel linearization and its finite random feature approximation is in *closed-form* with well-studied performance guarantees [19]. The superiority of spectral dynamic embedding has been justified empirically in the reinforcement learning setting [1] where system dynamics are unknown. However, there still lacks any end-to-end control algorithm with theoretical guarantees that utilizes the finite-dimensional approximation of spectral dynamics embedding. Indeed, in general, the finite-dimensional approximation error in policy evaluation and control optimality has not been investigated theoretically, motivating the present work.

Our contributions. In this paper, we close the gap by justifying the finite random feature approximation of spectral dynamic embedding in nonlinear control rigorously, enabling the practical usage of the kernelized linearization in control. Our contributions lie in the following folds.

We first formalize a computational tractable stochastic nonlinear control algorithm with finite-dimension truncation of spectral dynamic embedding representation, *Spectral Dynamics Embedding Control (SDEC)*. Specifically, we first extract the finite-dimensional spectral dynamics embedding in *closed-form* through Monte-Carlo approximation as explained in Section III-A, and then, we conduct the dynamic programming for value function upon these representations through least square policy evaluation. The policy is improved by natural policy gradient for optimal control based on the obtained value functions. The concrete algorithm is derived in Section III-B. We note that the particular way of policy evaluation and policy update is actually compatible with cutting-edge deep reinforcement learning methods, such as Soft Actor-Critic [20], by using our proposed representation to approximate the critic function. In other words, one of the novelties of SDEC is that it exploits the known nonlinear dynamics to obtain a nature, inherent representation space which could be adopted by various dynamical programming or policy gradient based methods.

We then characterize the policy evaluation error with finite-dimensional truncation and finite-sample approximation in SDEC in Section IV-A. We further provide a rigorous optimality analysis for the policy obtained by SDEC in Section IV-B, which to the best of our knowledge is the first time this has been done, due to the challenging complications in stochastic nonlinear control. Specifically, we show the gap between optimal policy and the SDEC induced policy is inversely proportional to a polynomial dependency w.r.t. number of features and the number of samples used in dynamic programming. Lastly, we conduct a numerical study on a robotic pendulum control problem to justify our theoretical analysis in Section V.

II. Problem Setup and Preliminaries

In this section, we introduce the stochastic nonlinear control problem that will be studied in this paper and reformulate it as a MDP. We will also briefly introduce the background knowledge about reproducing kernel Hilbert space (RKHS) and random features.

A. Stochastic Nonlinear Control Problem in MDPs

We consider the standard discrete-time nonlinear control model with γ-discounted infinite horizon, defined by

$$s_{t+1} = f(s_t, a_t) + \epsilon_t, \quad \text{where} \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2 I_d), \tag{1}$$

such that $\gamma \in (0,1)$, $s \in \mathcal{S} \subset \mathbb{R}^d$ is the state, $a \in \mathcal{A}$ is the control action, and $\{\epsilon_t\}_{t=1}^{\infty}$ are independent Gaussian noises. The function $f(\cdot,\cdot) : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ describes the general nonlinear dynamics, and $r : \mathcal{S} \times \mathcal{A} \to [0,1]$ gives the reward function on the state and action pair. Here we assume the reward function $r(s,a)$ is bounded for any $(s,a) \in \mathcal{S} \times \mathcal{A}$.[1] Without loss of generality, we assume there is a fixed initial state s_0. Given a stationary policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ with $\Delta(\mathcal{A})$ as the space of probability measures over $\mathcal{A}$, the accumulated reward over infinite horizon is given by

$$J^\pi = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right], \tag{2}$$

where the expectation is w.r.t. the stochastic dynamics and the random policy. In this paper, we study the optimal *control/planning* problem which is to seek a policy π^* that maximizes (2), given the dynamics f and the reward function r. Note that the nonlinearity of f and r makes this optimal control problem difficult as reviewed in the introduction.

The above stochastic nonlinear optimal control problem can also be described via Markov Decision Process. Consider an episodic homogeneous-MDP, denoted by $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, r, \gamma \rangle$, where $P(\cdot|s,a) : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ describes the state transition distribution, where $\Delta(\mathcal{S})$ denotes the space of probability measures on the set $\mathcal{S}$. Then, the stochastic nonlinear control model (1) can be recast as an MDP with transition dynamics

$$P(s'|s,a) \propto \exp\left(-\frac{\|f(s,a) - s'\|_2^2}{2\sigma^2} \right). \tag{3}$$

Meanwhile, given a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, the corresponding Q^π-function is given by

$$Q^\pi(s,a) = \mathbb{E}_{P,\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \middle| s_0 = s, a_0 = a \right]. \tag{4}$$

It is straightforward to show the Bellman recursion for Q^π,

$$Q^\pi(s,a) = r(s,a) + \gamma \mathbb{E}_P \left[V^\pi(s') \right] \tag{5}$$

with $V^\pi(s) := \mathbb{E}_\pi \left[Q^\pi(s,a) \right]$. Equivalently, the accumulated reward J^π is $V^\pi(s_0)$. The goal can be reformulated as seeking the optimal policy $\pi^* = \operatorname{argmax}_\pi V^\pi(s_0)$.

For an MDP with finite states and actions, optimal control can be obtained by solving dynamic program for the Q-function via via the Bellman relation (5). However, for continuous states and actions, representing the Q-function and

[1] The bounded assumption is for the purpose of theoretical analysis. Without the assumption, our methods could be applied but without theoretical guarantees. Indeed, our current results can not handle systems with potential unbounded rewards. For systems with bounded state and action space (e.g., inverted pendulum), such assumptions often hold. It is left as our future work to relax this assumption. Also note that we only need to assume $r(s,a)$ is bounded but not necessarily in $[0,1]$.

conducting dynamic programming (DP) in function space becomes the major issue for obtaining the optimal policy. In this paper, we will introduce spectral dynamic embedding, a novel kernelized linearization, to linearly represent the Q-function, and develop practical and provable method to learn/approximate the optimal control policy.

B. Reproducing Kernel Hilbert Space

In this section, we will introduce the necessary background of Reproducing Kernel Hilbert Spaces for understanding our proposed kernel embeddings.

Definition 1 ((Positive-Definite) Kernel [21]). *A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is said to be a kernel on the non-empty set $\mathcal{X}$ if there exists a Hilbert space $\mathcal{H}$ and a feature map $\psi(\cdot) : \mathcal{X} \to \mathcal{H}$ such that $\forall x, x' \in \mathcal{X}$, $k(x, x') = \langle \psi(x), \psi(x') \rangle_{\mathcal{H}}$. Moreover, the kernel is said to be positive definite if $\forall n \geqslant 1$, $\forall \{a_i\}_{i\in[n]} \subset \mathbb{R}$ and mutually distinct sets $\{x_i\}_{i\in[n]} \subset \mathcal{X}$, $\sum_{i\in[n]} \sum_{j\in[n]} a_i a_j k(x_i, x_j) > 0$.*

One can then define the RKHS associated with the kernel.

Definition 2 (Reproducing Kernel Hilbert Space [22]). *The Hilbert space $\mathcal{H}$ of an $\mathbb{R}$-valued function defined on a non-emptry set $\mathcal{X}$ is said to be a reproducing kernel Hilbert space (RKHS) if there is a kernel $k(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, such that*

1) *$\forall x \in \mathcal{X}$, $k(x, \cdot) \in \mathcal{H}$.*
2) *$\forall x \in \mathcal{X}, f \in \mathcal{H}$, we have that $\langle f, k(x, \cdot) \rangle_{\mathcal{H}} = f(x)$ (a.k.a the reproducing property), which also implies that $\langle k(x, \cdot), k(y, \cdot) \rangle_{\mathcal{H}} = k(x, y)$.*

Here $k(\cdot, \cdot)$ is called a (unique) reproducing kernel of $\mathcal{H}$.

The kernel associated with RKHS admits an important decomposition, as described in the following theorem.

Theorem 1 (Bochner [23]). *If $k(x, x')$ is a positive definite kernel, then there exists a set Ω, a measure $\rho(\omega)$ on Ω, and Fourier random feature $\psi_\omega(x) : \mathcal{X} \to \mathbb{C}$ such that*

$$k(x, x') = \mathbb{E}_{\rho(\omega)} \left[\psi_\omega(x) \psi_\omega(x') \right]. \tag{6}$$

It should be emphasized that the $\psi_\omega(\cdot)$ in Bochner decomposition may not be unique. The Bochner decomposition provides the random feature [24], which is applying Monte-Carlo approximation for (6), leading to a finite-dimension approximation for kernel methods. For example, for the Gaussian kernel, $k(x, x') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right)$, the corresponding $\rho(\omega)$ is a Gaussian proportional to $\exp\left(-\frac{\sigma^2\|\omega\|^2}{2}\right)$ with $\psi_\omega(x) = \exp\left(-\mathbf{i}\omega^\top x\right)$; for the Laplace kernel, the $\rho(\omega)$ is a Cauchy distribution with the same $\psi_\omega(\cdot)$. Please refer to Table 1 in [25] for more examples.

III. Control with Spectral Dynamic Embedding

We first introduce spectral dynamic embedding [1], a novel kernelized linearization, by which the Q-function for arbitrary policy can be represented *linearly*, therefore, the policy evaluation and control can be conducted within the linear space. This is significantly different from existing kernel linearization methods, which are designed for linearizing the dynamic model. We then propose the corresponding finite-dimensional approximated linear space for tractable optimal control. Specifically, we execute dynamic programming for policy evaluation in the linear function space obtained by the spectral dynamics embedding, upon which we improve the policy with natural policy gradient, leading to *Spectral Dynamics Embedding Control (SDEC)* in Algorithm 1.

A. Spectral Dynamics Embedding

As we discussed in Section II, we have recast the stochastic nonlinear control model as an MDP. By recognizing the transition operator (3) as a Gaussian kernel and using Theorem 1, we can further decompose the transition dynamics and reward in a *linear representation* as follows. Due to the space limitation, we defer all the proofs to our online report [26].

Proposition 2. *Denote $\theta_r = [0, 0, 1]^\top$,*

$$\begin{aligned} \phi_\omega(f(s,a)) = \Big[&\tfrac{g_\alpha(f(s,a))}{\alpha^d} \cos\left(\tfrac{\omega^\top f(s,a)}{\sqrt{1-\alpha^2}}\right), \\ &\tfrac{g_\alpha(f(s,a))}{\alpha^d} \sin\left(\tfrac{\omega^\top f(s,a)}{\sqrt{1-\alpha^2}}\right), r(s,a) \Big], \end{aligned} \tag{7}$$

$\mu_\omega(s') = p(s')[\cos(\sqrt{1-\alpha^2}\omega^\top s'), \sin(\sqrt{1-\alpha^2}\omega^\top s'), 0]^\top$, *where* $g_\alpha(f(s,a)) := \exp\left(\frac{\alpha^2\|f(s,a)\|^2}{2(1-\alpha^2)\sigma^2}\right)$, $\omega \sim \mathcal{N}(0, \sigma^{-2} I_d)$, $\forall \alpha \in [0, 1)$ *and* $p(s') = \frac{\alpha^d}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{\|\alpha s'\|^2}{2\sigma^2}\right)$ *is a Gaussian distribution for s' with standard deviation $\frac{\sigma}{\alpha}$, then*

$$\begin{aligned} P(s'|s,a) =& \mathbb{E}_{\omega\sim\mathcal{N}(0,\sigma^{-2}I_d)} \left[\phi_\omega(f(s,a))^\top \mu_\omega(s')\right] \\ :=& \langle \phi_\omega(f(s,a)), \mu_\omega(s') \rangle_{\mathcal{N}(0,\sigma^{-2}I_d)}, \end{aligned} \tag{8a}$$

$$\begin{aligned} r(s,a) =& \mathbb{E}_{\omega\sim\mathcal{N}(0,\sigma^{-2}I_d)} \left[\phi_\omega(f(s,a))^\top \theta_r\right] \\ :=& \langle \phi_\omega(f(s,a)), \theta_r \rangle_{\mathcal{N}(0,\sigma^{-2}I_d)}, \end{aligned} \tag{8b}$$

The $\phi_\omega(\cdot)$ is named as *Spectral Dynamic Embedding*. For convenience and without confusion, we abuse the notation $\langle \cdot, \cdot \rangle_{\mathcal{N}(0,\sigma^2 I_d)}$ as $\langle \cdot, \cdot \rangle$, and $\phi_\omega(\cdot)$ as $\phi(\cdot)$ in the paper. We emphasize the decomposition in [1] is a special case of in (8) with $\alpha = 1$. The tunable α provides benefits for the analysis and may also be used to improve empirical performance.

The most significant benefit of the decomposition is the property that it induces a function space composed by $\{\phi_\omega(s,a)\}$ with $\omega \sim \mathcal{N}\left(0, \sigma^2 I_d\right)$, where the Q-function for arbitrary policy can be linearly represented. This is shown in this next result, with proof in our online report [26].

Proposition 3 (Proposition 2.3 [27]). *For any policy, there exist weights θ^π such that the corresponding state-action value function $Q^\pi(s,a) = \langle \phi_\omega(f(s,a)), \theta^\pi_\omega \rangle$.*

In fact, the MDP with the factorization in (8) is an instantiation of the *linear MDP* [27], [28]. However, instead of assuming the factorization of the dynamics with finite dimension ϕ in linear MDPs, the spectral dynamics embedding decomposes general stochastic nonlinear model explicitly through infinite-dimensional kernel view.

Immediately, for the stochastic nonlinear control model (1) with an arbitrary dynamics $f(s,a)$, the spectral dynamic embedding $\phi_\omega(\cdot)$ provides a linear space, in which we can conceptually conduct dynamic programming for policy evaluation and optimal control with global optimal guarantee.

Algorithm 1: Spectral Dynamics Embedding Control (SDEC)

Spectral Dynamic Embedding Generation

1 Sample i.i.d. $\{\omega_i\}_{i\in[m]}$ where $\omega_i \sim \mathcal{N}(0, \sigma^{-2} I_d)$ and construct the feature

$$\phi(s,a) = [g(f(s,a))\sin(\omega_i^\top f(s,a)), \; g(f(s,a))\cos(\omega_i^\top f(s,a))]_{i\in[m]}.$$

Initialize $\theta_0 = 0$ and $\pi_0(a|s) \propto \exp(\phi(s,a)^\top \theta_0)$.

Least Square Policy Evaluation

2 **for** $k = 0, 1, \cdots, K$ **do**

3 Sample i.i.d. $\{(s_i, a_i, s_i'), a_i'\}_{i\in[n]}$ where $(s_i, a_i) \sim \nu_{\pi_k}$, $s_i' = f(s_i, a_i) + \varepsilon$, ν_{π_k} is the stationary distribution of π_k, $\varepsilon \sim \mathcal{N}(0, \sigma^2 I_d)$, $a_i' \sim \pi_k(s_i')$.

4 Initialize $\hat{w}_{k,0} = 0$.

5 **for** $t = 0, 1, \cdots, T-1$ **do**

6 Solve

$$\hat{w}_{k,t+1} = \arg\min_{w} \tag{9}$$

7 $$\left\{\sum_{i\in[n]} \left(\phi(s_i,a_i)^\top w - r(s_i,a_i) - \gamma \phi(s_i', a_i')^\top \hat{w}_{k,t}\right)^2\right\}$$

8 **end**

Natural Policy Gradient for Control

9 Update $\theta_{k+1} = \theta_k + \eta \hat{w}_{k,T}$ and

$$\pi_{k+1}(a|s) \propto \exp(\phi(s,a)^\top \theta_{k+1}). \tag{10}$$

10 **end**

B. Control with Finite-dimensional Approximation

Although the spectral dynamic embedding in Proposition 2 provides a linear space to represent the family of Q-function, there is still a major challenge to be overcome for practical implementation. Specifically, the dimension of $\phi_\omega(\cdot)$ is *infinite* with $\omega \sim \mathcal{N}\left(0, \sigma^{-2} I_d\right)$, which is computational intractable. [1] suggested the finite-dimensional random feature, which is the Monte-Carlo approximation for the Bochner decomposition [24], and demonstrated strong empirical performances for reinforcement learning. However, there is no formal control algorithm established with the finite-dimensional approximation, and their regret analysis ignores the approximation error from finite-dimensional truncation, leaving gaps between the theoretical justification and the empirical success.

In this section, we first formalize the Spectral Dynamics Embedding Control (SDEC) algorithm, implementing the dynamic programming efficiently in a principled way as shown in Algorithm 1, whose theoretical property will be analyzed in the next section.

In SDEC, there are three main components,

1) *Generating spectral dynamic embedding (Line 1).* Following Proposition 2, we construct finite-dimensional $[\phi_{\omega_i}(s,a)]_{i=1}^m \in \mathbb{R}^{m\times 1}$ by Monte-Carlo approximation with $\omega_i \sim \mathcal{N}\left(0, \sigma^2 I\right)$, which will be used for representing the state-value functions.
2) *Policy evaluation (Line 4 to 8).* We conduct the least square policy evaluation for estimating the state-value function of current policy upon the generated finite-dimensional truncation features. We sample from the stationary distribution $\nu_\pi(s,a)$ from dynamics under current policy π and solve a series of least square regression (9) to learn a $Q^\pi(s,a) = \phi(s,a)^\top w^\pi$, with $w^\pi \in \mathbb{R}^{m\times 1}$ by minimizing a Bellman recursion type loss.
3) *Policy update (Line 9).* Once we have the state-value function for current policy estimated well in step 2), we will update the policy by natural policy gradient or mirror descent in (10), *i.e.*,

$$\pi^{t+1}(a|s) = \underset{\pi(\cdot|s)\in\Delta(\mathcal{A})}{\operatorname{argmax}} \; \left\langle \pi, \phi(s,a)^\top w^{\pi_t} \right\rangle + \frac{1}{\eta} KL(\pi||\pi_t).$$

Once the finite-dimensional spectral dynamic embedding has been generated in step 1), the algorithm alternates between step 2) and step 3) to improve the policy.

We emphasize that although we tailored the natural policy gradient with least square policy evaluation upon the proposed spectral dynamic embedding in Algorithm 1, the spectral dynamic embedding is also compatible to other planning methods, and we leave the algorithm design and theoretical analysis as our future work.

Remark 1 (Beyond Gaussian noise). *Although we derive SDEC mainly for the stochastic nonlinear dynamics with Gaussian noise, the method can be easily extended for more flexible noise model by considering*

$$\zeta(s') = f(s,a) + \epsilon, \quad \textit{with} \quad \epsilon \sim \mathcal{N}\left(0, \sigma^2 I_d\right), \tag{11}$$

where $\zeta(\cdot)$ is a nonlinear model. When ζ is invertible, the model can be understood as $s' = \zeta^{-1}(f(s,a) + \epsilon)$, therefore the noise is no longer Gaussian w.r.t. s'. We emphasize SDEC is still applicable to (11) for arbitrary $\zeta(\cdot)$, which generalizes the method beyond Gaussian noise.

IV. Theoretical Analysis

The major difficulty in analyzing the optimality of the policy induced by SDEC is the fact that after finite-dimensional truncation, the transition operator constructed by the random feature $\widehat{P}(s'|s,a) := \frac{1}{m}\sum_{i=1}^m \phi_{\omega_i}(f(s,a))\,\mu_{\omega_i}(s')$ with $\{\omega_i\}_{i=1}^m \sim \mathcal{N}\left(0, \sigma^{-2} I_d\right)$ is no longer a valid distribution, *i.e.*, it lacks non-negativity and normalization, which induces a *pseudo-MDP* [29] as the approximation. As a consequence, the value function for the pseudo-MDP is not bounded. Then, the vanilla proof strategy used in majority of the literature since [30], *i.e.*, analyzing the optimality gap between policies through simulation lemma, is no longer applicable. We bypass the difficulty from pseudo-MDP, and provide rigorous investigation of the impact of the approximation error for policy evaluation and optimization in SDEC, filling the long-standing gap. We defer all proofs to our online report [26].

We first specify the assumptions, under which we derive our theoretical results below. These assumptions are commonly used in the literature [27], [31]–[33].

Assumption 1 (Regularity Condition for Dynamics and Reward)**.** *$\|f(s,a)\| \leqslant c_f$, and $r(s,a) \leqslant c_r$ for all $s \in \mathcal{S}, a \in \mathcal{A}$. For the ease of presentation, we omit the polynomial dependency on c_f and c_r and focus on the dependency of other terms of interest.*

Assumption 2 (Regularity Condition for Feature)**.** *The features are linearly independent.*

Assumption 3 (Regularity Condition for Stationary Distribution [33])**.** *The stationary distribution ν_π for all policy π has full support, and satisfies the following conditions:*
$\lambda_{\min}\left(\mathbb{E}_{\nu_\pi}\left[\phi(s,a)\phi(s,a)^\top\right]\right) \geqslant \lambda_1$,
$\lambda_{\min}\left(\mathbb{E}_{\nu_\pi}\left[\phi(s,a)\left(\phi(s,a)-\gamma\mathbb{E}_{\nu_\pi}\phi(s',a')\right)^\top\right]\right) \geqslant \lambda_2$,
where $\lambda_1, \lambda_2 > 0$.

A. Error Analysis for Policy Evaluation

For notation simplicity, we omit π and use ν to denote the stationary distribution throughout this section. Our analysis starts from the error for policy evaluation (Line 4 to 8 in Algorithm 1). We decompose the error into two parts, one is the irreducible error due to the limitation of our basis (*i.e.*, finite m in Line 1), and one is the statistical error due to the finite number of samples we use (*i.e.*, finite n in Line 4). Deferring proof details to the online report, our main result here is the following.

Theorem 4. *Let $T = \Theta(\log n)$. With probability at least $1-\delta$, we have that*

$$\left\|Q^\pi - \hat{Q}_T^\pi\right\|_\nu = \tilde{O}\left(\frac{1}{(1-\gamma)^2\sqrt{m}} + \frac{m^3}{(1-\gamma)\lambda_1^2\lambda_2\sqrt{n}}\right). \quad (12)$$

Theorem 4 provides the estimation error of Q with least square policy evaluation under the $\|\cdot\|_\nu$ norm. It can be used to provide an estimation error of J^π shown in the following corollary. Meanwhile, it also performs as a cornerstone of control optimality analysis, as we will show in the next section. The bound also reveals a fundamental tradeoff between the approximation error and statistical error: presumably, a larger number of m will make the finite kernel truncation capable of approximating the original infinite-dimensional function space better but it also requires a larger number of samples n in order to train the weights well.

Corollary 5. *With high probability, we have that*

$$\left|J^\pi - \hat{J}_T^\pi\right| = \quad (13)$$
$$\tilde{O}\left(\sqrt{\max_{s,a}\frac{\mu(s)\pi(a|s)}{\nu(s,a)}}\left(\frac{1}{(1-\gamma)^2\sqrt{m}} + \frac{m^3}{(1-\gamma)\lambda_1^2\lambda_2\sqrt{n}}\right)\right)$$

B. Error Analysis of Natural Policy Gradient for Control

We state here the performance guarantee for the control optimality with natural policy gradient.

Theorem 6. *Let $\eta = \Theta\left(\lambda_2 m^{-1}\sqrt{\log|\mathcal{A}|}\right)$. With high probability, we have that*

$$\mathbb{E}\left[\min_{k<K}\left\{V^{\pi^*} - V^{\pi_k}\right\}\right] = \tilde{O}\left(\frac{m}{\lambda_2}\sqrt{\frac{\log|\mathcal{A}|}{K}}\right. \quad (14)$$
$$\left. + \frac{1}{1-\gamma}\sqrt{\max_{s,a,\pi,k}\frac{d^{\pi^*}(s)\pi(a|s)}{\nu_{\pi_k}(s,a)}}\left(\frac{1}{(1-\gamma)^2\sqrt{m}} + \frac{m^3}{(1-\gamma)\lambda_1^2\lambda_2\sqrt{n}}\right)\right)$$

We emphasize that Theorem 6 characterizes the gap between optimal policy and the solution provided by SDEC, taking in account of finite-step in policy optimization (K), finite-dimension (m) and finite-sample (n) approximation in policy evaluation, respectively, which, to the best of our knowledge, is established for the first time. As we can see, with m increases, we can reduce the approximation error, but the optimization and sample complexity will also increase accordingly. We can further balance the terms for the optimal dimension of features.

V. Simulations

In this section, we compare SDEC with iterative LQR (iLQR) [7] and Koopman-based control [15], two well-known alternatives for nonlinear control.

A. Pendulum environment

The pendulum swingup problem is a classical nonlinear control task. The system comprises a pendulum attached at one end to a fixed point, and the other end being free. The goal of the control task is to apply torque on the free end to swing it into an upright position, with its center of gravity right above the fixed point. In our simulations, we use the Pendulum-v1 dynamics from the OpenAI gym.

B. Details of SDEC implementation

In our empirical implementation of SDEC, we combine spectral dynamical embedding with Soft Actor-Critic (SAC) [20]. Specifically, we use random features to parameterize the critic, and use this as the critic in SAC. In SAC, it is necessary to maintain a parameterized function $Q(s,a)$ which estimates the soft Q-value (which includes not just the reward but also an entropy term encouraging exploration). For a given policy π, the soft Q-value satisfies the relationship $Q^\pi(s_t,a_t) = r(s_t,a_t) + \gamma\mathbb{E}_{s_{t+1}\sim p}[V(s_{t+1})]$, where $V(s_t) = \mathbb{E}_{a_t\sim\pi}[Q(s_t,a_t) - \alpha\log\pi(a_t \mid s_t)]$. Based on the spectral dynamic embedding proposed in our paper, we parameterize the Q^π-function as $Q^\pi(s,a) = r(s,a) + \tilde{\phi}_\omega(s,a)^\top\tilde{\theta}^\pi$ where

$$\tilde{\phi}_\omega(s,a) = \left[\cos(\omega_1^\top s' + b_1), \cdots, \cos(\omega_m^\top s' + b_m)\right]^\top,$$

where $s' = f(s,a)$, with $\{\omega_i\}_{i=1}^m \sim \mathcal{N}(0, I_d)$ and $\{b_i\}$ drawn iid from $\mathrm{Unif}([0,2\pi])$. The $\tilde{\theta}^\pi \in \mathbb{R}^m$ is updated using (mini-batch) gradient descent for (9). We then use this Q-function for the actor update in SAC.

C. Performance versus other nonlinear control algorithms

We compare our algorithm against iterative LQR (iLQR) and Koopman-based control. For iLQR, we used the implementation in [34], where we added a log barrier function to account for the input constraint in the OpenAI Pendulum-v1 environment. Meanwhile for the Koopman-based control, we adapted an implementation called Deep KoopmanU with Control (DKUC) proposed in the paper [35], where we combine the dynamics learned from DKUC with MPC to enforce the input constraint. Apart from the noiseless setting, we also considered the setting where a noise term $\epsilon \sim \mathcal{N}(0,\sigma^2)$ is added to the angular acceleration at each step, where $\sigma \in \{1,2,3\}$. To evaluate the performance, we computed the average total episodic reward for each algorithm across 100 randomly chosen episodes, where each episode comprises 200 time-steps. For SDEC and DKUC, we trained using 4 different random seeds, and both algorithms have access to 400 episodes worth of environment interaction.

The performance of the various algorithms in both the noiseless and noisy setting (with $\sigma = 1$) can be found in Table I. For algorithms with inherent stochasticity (SDEC

TABLE I

PERFORMANCE COMPARISON.

	Pendulum	Pendulum (noisy, $\sigma = 1$)
SDEC	$\mathbf{-279.0\,(\pm 31.8)}$	$\mathbf{-252.0(\pm 2.2)}$
iLQR	-1084.7	-1330.7 ($\pm$35.3)
DKUC	-1090.9 ($\pm$35.9)	-1050.2 ($\pm$25.3)

and DKUC), the mean and standard deviation (in brackets) over 4 random seeds is shown. Throughout the simulations presented in this section, the number of random features used in SDEC is 512. We observe that the SDEC strongly outperforms iLQR and DKUC in both the noiseless and noisy setting (higher reward is better; in this case the negative sign is due to the rewards representing negative cost).

We further plot the evaluation performance of SDEC during the course of its learning in Figure 1. In both the noiseless and noisy settings, the performance of SDEC continuously improves until it saturated after around 300 episodes.

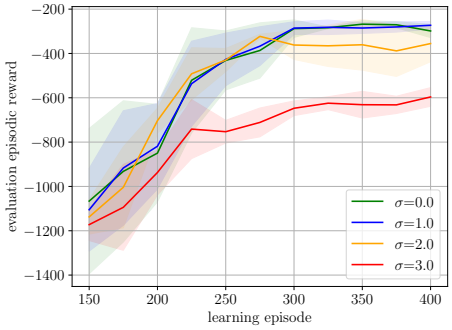

Fig. 1. SDEC performances with varying noise levels $\sigma \in \{0, 1, 2, 3\}$. The evaluation is performed every 25 learning episodes on a fresh evaluation set of 100 episodes, and the y-axis represents the average episodic reward on the evaluation set. The shaded regions represent a 1 standard deviation confidence interval (across 4 random seeds).

VI. CONCLUSION

There is a long-standing gap between the theoretical understanding and empirical success of the kernelized linearization control with spectral dynamic embedding, *i.e.*, the error induced by finite-dimensional approximation has not be clearly analyzed. We close this gap by exploiting a novel analysis method, which could be of independent interest, and characterizing the finite-dimensional approximation effect in both policy evaluation and optimization, theoretically validating the usage of SDEC in practice.

REFERENCES

[1] T. Ren, T. Zhang, C. Szepesvári, and B. Dai, "A free lunch from the noise: Provable and practical exploration for representation learning," in *Uncertainty in Artificial Intelligence*. PMLR, 2022, pp. 1686–1696.

[2] J.-J. E. Slotine, W. Li, *et al.*, *Applied nonlinear control*. Prentice hall Englewood Cliffs, NJ, 1991, vol. 199, no. 1.

[3] H. Khalil, *Nonlinear Control*. Pearson Education Limited, 2015.

[4] D. A. Lawrence and W. J. Rugh, "Gain scheduling dynamic linear controllers for a nonlinear plant," *Automatica*, vol. 31, no. 3, 1995.

[5] W. J. Rugh and J. S. Shamma, "Research on gain scheduling," *Automatica*, vol. 36, no. 10, pp. 1401–1425, 2000.

[6] B. Charlet, J. Lévine, and R. Marino, "On dynamic feedback linearization," *Systems & Control Letters*, vol. 13, no. 2, pp. 143–151, 1989.

[7] A. Sideris and J. E. Bobrow, "An efficient sequential linear quadratic algorithm for solving nonlinear optimal control problems," in *Proceedings of American Control Conference*. IEEE, 2005, pp. 2275–2280.

[8] C. Edwards and S. Spurgeon, *Sliding mode control: theory and applications*. Crc Press, 1998.

[9] R. Brockett, "The early days of geometric nonlinear control," *Automatica*, vol. 50, no. 9, pp. 2203–2224, 2014.

[10] P. V. Kokotovic, "The joy of feedback: nonlinear and adaptive," *IEEE Control Systems Magazine*, vol. 12, no. 3, pp. 7–17, 1992.

[11] J. A. Primbs, V. Nevistić, and J. C. Doyle, "Nonlinear optimal control: A control lyapunov function and receding horizon perspective," *Asian Journal of Control*, vol. 1, no. 1, pp. 14–24, 1999.

[12] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.

[13] S. Prajna, A. Papachristodoulou, P. Seiler, and P. A. Parrilo, "Sostools and its control applications," in *Positive polynomials in control*. Springer, 2005, pp. 273–292.

[14] B. O. Koopman, "Hamiltonian systems and transformation in hilbert space," *Proceedings of the national academy of sciences of the united states of america*, vol. 17, no. 5, p. 315, 1931.

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

[16] S. Kakade, A. Krishnamurthy, K. Lowrey, M. Ohnishi, and W. Sun, "Information theoretic regret bounds for online nonlinear control," *Adv. in Neural Info. Processing Systems*, vol. 33, pp. 15 312–15 325, 2020.

[17] S. Grunewalder, G. Lever, L. Baldassarre, M. Pontil, and A. Gretton, "Modelling transition dynamics in mdps with rkhs embeddings," *arXiv preprint arXiv:1206.4655*, 2012.

[18] R. Kamalapurkar and J. A. Rosenfeld, "An occupation kernel approach to optimal control," *arXiv preprint arXiv:2106.00663*, 2021.

[19] A. Rahimi and B. Recht, "Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning," *Advances in neural information processing systems*, vol. 21, 2008.

[20] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, 2018.

[21] M. A. Álvarez, L. Rosasco, N. D. Lawrence, *et al.*, "Kernels for vector-valued functions: A review," *Foundations and Trends® in Machine Learning*, vol. 4, no. 3, pp. 195–266, 2012.

[22] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American mathematical society*, vol. 68, no. 3, pp. 337–404, 1950.

[23] A. Devinatz, "Integral representations of positive definite functions," *Transactions of the American Mathematical Society*, vol. 74, no. 1, pp. 56–77, 1953.

[24] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," *Adv. in Neural Info. Processing Systems*, vol. 20, 2007.

[25] B. Dai, B. Xie, N. He, Y. Liang, A. Raj, M.-F. Balcan, and L. Song, "Scalable kernel methods via doubly stochastic gradients," *Adv. in Neural Info. Processing Systems*, vol. 27, pp. 3041–3049, 2014.

[26] T. Ren, Z. Ren, N. Li, and B. Dai, "Stochastic nonlinear control via finite-dimensional spectral dynamic embedding," *arXiv preprint arXiv:2304.03907*, 2023.

[27] C. Jin, Z. Yang, Z. Wang, and M. I. Jordan, "Provably efficient reinforcement learning with linear function approximation," in *Conference on Learning Theory*. PMLR, 2020, pp. 2137–2143.

[28] L. Yang and M. Wang, "Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound," in *International Conference on Machine Learning*. PMLR, 2020, pp. 10 746–10 756.

[29] H. Yao, C. Szepesvári, B. A. Pires, and X. Zhang, "Pseudo-mdps and factored linear action models," in *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, 2014, pp. 1–9.

[30] M. Kearns and S. Singh, "Near-optimal reinforcement learning in polynomial time," *Machine learning*, vol. 49, no. 2, pp. 209–232, 2002.

[31] H. Yu and D. P. Bertsekas, "New error bounds for approximations from projected linear equations," in *European Workshop on Reinforcement Learning*. Springer, 2008, pp. 253–267.

[32] A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan, "On the theory of policy gradient methods: Optimality, approximation, and distribution shift," *J Mach Learn Res*, vol. 22, no. 98, pp. 1–76, 2021.

[33] Y. Abbasi-Yadkori, P. Bartlett, K. Bhatia, N. Lazic, C. Szepesvari, and G. Weisz, "Politex: Regret bounds for policy iteration using expert prediction," in *International Conference on Machine Learning*, 2019.

[34] V. Roulet, S. Srinivasa, M. Fazel, and Z. Harchaoui, "Iterative linear quadratic optimization for nonlinear control: Differentiable programming algorithmic templates," *arXiv preprint arXiv:2207.06362*, 2022.

[35] H. Shi and M. Q.-H. Meng, "Deep koopman operator with control for nonlinear systems," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7700–7707, 2022.