

---

# Continuous-time Value Function Approximation in Reproducing Kernel Hilbert Spaces

---

**Motoya Ohnishi**  
Keio Univ., KTH, RIKEN  
motoya.ohnishi@riken.jp

**Masahiro Yukawa**  
Keio Univ., RIKEN  
yukawa@elec.keio.ac.jp

**Mikael Johansson**  
KTH  
mikaelj@ee.kth.se

**Masashi Sugiyama**  
RIKEN, Univ. Tokyo  
masashi.sugiyama@riken.jp

## Abstract

Motivated by the success of reinforcement learning (RL) for discrete-time tasks such as AlphaGo and Atari games, there has been a recent surge of interest in using RL for continuous-time control of physical systems (cf. many challenging tasks in OpenAI Gym and DeepMind Control Suite). Since discretization of time is susceptible to error, it is methodologically more desirable to handle the system dynamics directly in continuous time. However, very few techniques exist for continuous-time RL and they lack flexibility in value function approximation. In this paper, we propose a novel framework for model-based continuous-time value function approximation in reproducing kernel Hilbert spaces. The resulting framework is so flexible that it can accommodate any kind of kernel-based approach, such as Gaussian processes and kernel adaptive filters, and it allows us to handle uncertainties and nonstationarity without prior knowledge about the environment or what basis functions to employ. We demonstrate the validity of the presented framework through experiments.

## 1 Introduction

Reinforcement learning (RL) [37, 20, 35] has been successful in a variety of applications such as AlphaGo and Atari games, particularly for discrete stochastic systems. Recently, application of RL to physical control tasks has also been gaining attention, because solving an optimal control problem (or the Hamilton-Jacobi-Bellman-Isaacs equation) [21] directly is computationally prohibitive for complex nonlinear system dynamics and/or cost functions.

In the physical world, states and actions are continuous, and many dynamical systems evolve in continuous time. OpenAI Gym [7] and DeepMind Control Suite [40] offer several representative examples of such physical tasks. When handling continuous-time (CT) systems, CT formulations are methodologically desirable over the use of discrete-time (DT) formulations with the small time intervals, since such discretization is susceptible to errors. In terms of computational complexities and the ease of analysis, CT formulations are also more advantageous over DT counterparts for control-theoretic analyses such as *stability* and *forward invariance* [14], which are useful for safety-critical applications. As we will show in this paper, our framework allows to constrain control inputs and/or states in a computationally efficient way.

One of the early examples of RL for CT systems [4] pointed out that Q learning is incapable of learning in continuous time and proposed advantage updating. Convergence proofs were given in [25] for systems described by stochastic differential equations (SDEs) [28] using a grid-based

Table 1: Relations to the existing approaches

	DT	DT stochastic (MDP)	CT	CT stochastic
Non kernel-based	(e.g. [5])	(e.g. [44])	(e.g. [8])	(e.g. [25])
<b>Kernel-based</b>	<b>(e.g. [9])</b>	<b>(e.g. [13])</b>	<b>(This work)</b>	<b>(This work)</b>

discretization of states and time. Stochastic differential dynamic programming and RL have also been studied in, for example, [43, 30, 42]. For continuous states and actions, function approximators are often employed instead of finely discretizing the state space to avoid the explosion of computational complexities. The work in [8] presented an application of CT-RL by function approximators such as Gaussian networks with fixed number of basis functions. In [45], it was mentioned that any continuously differentiable value function (VF) can be approximated by increasing the number of independent basis functions to infinity in CT scenarios, and a CT policy iteration was proposed.

However, without resorting to the theory of reproducing kernels [3], determining the number of basis functions and selecting the suitable basis function class cannot be performed systematically in general. Nonparametric learning is often desirable when no *a priori* knowledge about a suitable set of basis functions for learning is available. Kernel-based methods have many non-parametric learning algorithms, ranging from Gaussian processes (GPs) [32] to kernel adaptive filters (KFs) [22], which can provably deal with uncertainties and nonstationarity. While DT kernel-based RL was studied in [29, 49, 41, 36, 26, 13, 27], for example, and the Gaussian process temporal difference (GPTD) algorithm was presented in [9], no CT kernel-based RL has been proposed to our knowledge. Moreover, there is no unified framework in which existing kernel methods and their convergence/tracking analyses are straightforwardly applied to model-based VF approximation.

In this paper, we present a novel theoretical framework of model-based CT-VF approximation in reproducing kernel Hilbert spaces (RKHSs) [3] for systems described by SDEs. The RKHS for learning is defined through one-to-one correspondence to a user-defined RKHS in which the VF being obtained is lying. We then obtain the associated kernel to be used for learning. The resulting framework renders any kind of kernel-based methods applicable in model-based CT-VF approximation, including GPs [32] and KFs [22]. In addition, we propose an efficient barrier-certified policy update for CT systems, which implicitly enforces state constraints. Relations of our framework to the existing approaches for DT, DT stochastic (the Markov decision process (MDP)), CT, and CT stochastic systems are shown in Table 1. Our proposed framework covers model-based VF approximation working in RKHSs, including those for CT and CT stochastic systems. We verify the validity of the framework on the classical Mountain Car problem and a simulated inverted pendulum.

## 2 Problem setting

Throughout,  $\mathbb{R}$ ,  $\mathbb{Z}_{\geq 0}$ , and  $\mathbb{Z}_{> 0}$  are the sets of real numbers, nonnegative integers, and strictly positive integers, respectively. We suppose that the system dynamics described by the SDE [28],

$$dx = h(x(t), u(t))dt + \eta(x(t), u(t))dw, \quad (1)$$

is known or learned, where  $x(t) \in \mathbb{R}^{n_x}$ ,  $u(t) \in \mathcal{U} \subset \mathbb{R}^{n_u}$ , and  $w$  are the state, control, and a Brownian motion of dimensions  $n_x \in \mathbb{Z}_{> 0}$ ,  $n_u \in \mathbb{Z}_{> 0}$ , and  $n_w \in \mathbb{Z}_{> 0}$ , respectively,  $h : \mathbb{R}^{n_x} \times \mathcal{U} \rightarrow \mathbb{R}^{n_x}$  is the drift, and  $\eta : \mathbb{R}^{n_x} \times \mathcal{U} \rightarrow \mathbb{R}^{n_x \times n_w}$  is the diffusion. A Brownian motion can be considered as a process noise, and is known to satisfy the Markov property [28]. Given a policy  $\phi : \mathbb{R}^{n_x} \rightarrow \mathcal{U}$ , we define  $h^\phi(x) := h(x, \phi(x))$  and  $\eta^\phi(x) := \eta(x, \phi(x))$ , and make the following two assumptions.

**Assumption 1.** For any Lipschitz continuous policy  $\phi$ , both  $h^\phi(x)$  and  $\eta^\phi(x)$  are Lipschitz continuous, i.e., the stochastic process defined in (1) is an Itô diffusion [28, Definition 7.1.1], which has a pathwise unique solution for  $t \in [0, \infty)$ .

**Assumption 2.** The set  $\mathcal{X} \subset \mathbb{R}^{n_x}$  is compact with nonempty interior  $\text{int}(\mathcal{X})$ , and  $\text{int}(\mathcal{X})$  is invariant under the system (1) with any Lipschitz continuous policy  $\phi$ , i.e.,

$$P_x(x(t) \in \text{int}(\mathcal{X})) = 1, \quad \forall x \in \text{int}(\mathcal{X}), \quad \forall t \geq 0, \quad (2)$$

where  $P_x(x(t) \in \text{int}(\mathcal{X}))$  denotes the probability that  $x(t)$  lies in  $\text{int}(\mathcal{X})$  when starting from  $x(0) = x$ .

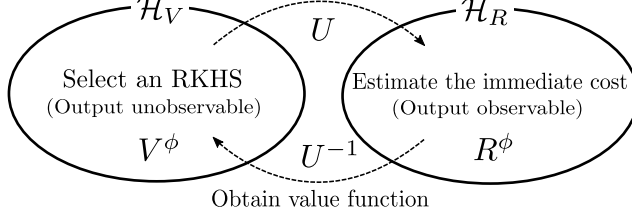


Figure 1: An illustration of the main ideas of our proposed framework. Given a system dynamics and an RKHS  $\mathcal{H}_V$  for the VF  $V^\phi$ , define  $\mathcal{H}_R$  under one-to-one correspondence to estimate an observable immediate cost function in  $\mathcal{H}_R$ , and obtain  $V^\phi$  by bringing it back to  $\mathcal{H}_V$ .

Assumption 2 implies that a solution of the system (1) stays in  $\text{int}(\mathcal{X})$  with probability one. We refer the readers to [15] for stochastic stability and invariance for SDEs.

In this paper, we consider the immediate cost function<sup>1</sup>  $R : \mathbb{R}^{n_x} \times \mathcal{U} \rightarrow \mathbb{R}$ , which is continuous and satisfies  $E_x \left[ \int_0^\infty e^{-\beta t} |R(x(t), u(t))| dt \right] < \infty$ , where  $E_x$  is the expectation for all trajectories (time evolutions of  $x(t)$ ) starting from  $x(0) = x$ , and  $\beta \geq 0$  is the discount factor. Note this boundedness implies that  $\beta > 0$  or that there exists a zero-cost state which is stochastically asymptotically stable [15]. Specifically, we consider the case where the immediate cost is *not* known *a priori* but is sequentially observed. Now, the VF associated with a policy  $\phi$  is given by

$$V^\phi(x) := E_x \left[ \int_0^\infty e^{-\beta t} R^\phi(x(t)) dt \right] < \infty, \quad (3)$$

where  $R^\phi(x(t)) := R(x(t), \phi(x(t)))$ .

The advantages of using CT formulations include a smooth control performance and an efficient policy update, and CT formulations require no elaborative partitioning of time [8]. In addition, our work shows that CT formulations make control-theoretic analyses easier and computationally more efficient and are more advantageous in terms of susceptibility to errors when the time interval is small. We mention that the CT formulation can still be considered in spite of the fact that the algorithm is implemented in discrete time.

With these problem settings in place, our goal is to estimate the CT-VF in an RKHS and improve policies. However, since the output  $V^\phi(x)$  is unobservable and the so-called double-sampling problem exists when approximating VFs (see e.g., [38, 16]), kernel-based supervised learning and its analysis cannot be directly applied to VF approximation in general. Motivated by this fact, we propose a novel model-based CT-VF approximation framework which enables us to conduct kernel-based VF approximation as supervised learning.

### 3 Model-based CT-VF approximation in RKHSs

In this section, we briefly present an overview of our framework; We take the following steps:

1. Select an RKHS  $\mathcal{H}_V$  which is supposed to contain  $V^\phi$  as one of its elements.
2. Construct another RKHS  $\mathcal{H}_R$  under one-to-one correspondence to  $\mathcal{H}_V$  through a certain bijective linear operator  $U : \mathcal{H}_V \rightarrow \mathcal{H}_R$  to be defined later in the next section.
3. Estimate the immediate cost function  $R^\phi$  in the RKHS  $\mathcal{H}_R$  by kernel-based supervised learning, and return its estimate  $\hat{R}^\phi$ .
4. An estimate of the VF  $V^\phi$  is immediately obtained by  $U^{-1}(\hat{R}^\phi)$ .

An illustration of our framework is depicted in Figure 1. Note we can avoid the double-sampling problem because the operator  $U$  is deterministic even though the system dynamics is stochastic. Therefore, under this framework, model-based CT-VF approximation in RKHSs can be derived, and convergence/tracking analyses of kernel-based supervised learning can also be applied to VF approximation.

---

**Algorithm 1** Model-based CT-VF Approximation in RKHSs with Barrier-Certified Policy Updates
 

---

**Estimate of the VF:**  $\hat{V}_n^\phi = U^{-1}(\hat{R}_n^\phi)$

**for**  $n \in \mathbb{Z}_{\geq 0}$  **do**

- Receive  $x_n \in \mathcal{X}$ ,  $\phi(x_n) \in \mathcal{U}$ , and  $R(x_n, \phi(x_n)) \in \mathbb{R}$

- Update the estimate  $\hat{R}_n^\phi$  of  $R^\phi$  by using some kernel-based method in  $\mathcal{H}_R$   $\triangleright$  e.g., Section 6

- Update the policy with barrier certificates when  $V^\phi$  is well estimated  $\triangleright$  e.g., (11)

**end for**

---

**Policy update while restricting certain regions of the state space** As mentioned above, one of the advantages of a CT framework is its affinity for control-theoretic analyses such as *stability* and *forward invariance*, which are useful for safety-critical applications. For example, suppose that we need to restrict the region of exploration in the state space to some set  $\mathcal{C} := \{x \in \mathcal{X} \mid b(x) \geq 0\}$ , where  $b : \mathcal{X} \rightarrow \mathbb{R}$  is smooth. This is often required for safety-critical applications.

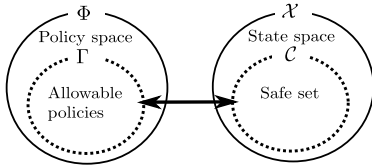


Figure 2: An illustration of barrier-certified policy updates. State constraints are implicitly enforced via barrier certificates.

To this end, control inputs must be properly constrained so that the state trajectory remains inside the set  $\mathcal{C}$ . In the safe RL context, there exists an idea of considering a smaller space of allowable policies (see [11] and references therein). To effectively constrain policies, we employ control barrier certificates (cf. [50, 48, 12, 46, 2, 1]). Without explicitly calculating the state trajectory over a long time horizon, it is known that any Lipschitz continuous policy satisfying control barrier certificates renders the set  $\mathcal{C}$  forward invariant [50], i.e., the state trajectory remains inside the set  $\mathcal{C}$ . In other words, we can implicitly enforce state constraints by satisfying barrier certificates when updating policies. Barrier-certified policy update was first introduced in [27] for DT systems, but is computationally more efficient in our CT scenario. This concept is illustrated in Figure 2, where  $\Phi$  is the space of Lipschitz continuous policies

$\phi : \mathcal{X} \rightarrow \mathcal{U}$ , and  $\Gamma$  is the space of barrier-certified allowable policies.

A brief summary of the proposed model-based CT-VF approximation in RKHSs is given in Algorithm 1. In the next section, we present theoretical analyses of our framework.

## 4 Theoretical analyses

We presented the motivations and an overview of our framework in the previous section. In this section, we validate the proposed framework from theoretical viewpoints. Because the output  $V^\phi(x)$  of the VF is unobservable, we follow the strategy presented in the previous section. First, by properly identifying the RKHS  $\mathcal{H}_V$  which is supposed to contain the VF, we can implicitly restrict the class of the VF. If the VF  $V^\phi$  is twice continuously differentiable<sup>2</sup> over  $\text{int}(\mathcal{X}) \subset \mathcal{X}$ , we obtain the following Hamilton-Jacobi-Bellman-Isaacs equation [28]:

$$\beta V^\phi(x) = -\mathcal{G}(V^\phi)(x) + R^\phi(x), \quad x \in \text{int}(\mathcal{X}), \quad (4)$$

where the infinitesimal generator  $\mathcal{G}$  is defined as

$$\mathcal{G}(V^\phi)(x) := -\frac{1}{2} \text{tr} \left[ \frac{\partial^2 V^\phi(x)}{\partial x^2} A^\phi(x) \right] - \frac{\partial V^\phi(x)}{\partial x} h^\phi(x), \quad x \in \text{int}(\mathcal{X}). \quad (5)$$

Here,  $\text{tr}$  stands for the trace, and  $A^\phi(x) := A(x, \phi(x)) \in \mathbb{R}^{n_x \times n_x}$ , where  $A(x, u) = \eta(x, u)\eta(x, u)^\top$ . By employing a suitable RKHS such as a Gaussian RKHS for  $\mathcal{H}_V$ , we can guarantee twice continuous differentiability of an estimated VF. Note that functions in a Gaussian RKHS are smooth [23], and any continuous function on every compact subset of  $\mathbb{R}^{n_x}$  can be approximated with an arbitrary accuracy [34] in a Gaussian RKHS.

<sup>1</sup>The cost function might be obtained by the negation of the reward function.

<sup>2</sup> See, for example, [10, Chapter IV],[19], for more detailed arguments about the conditions under which twice continuous differentiability is guaranteed.

Next, we need to construct another RKHS  $\mathcal{H}_R$  which contains the immediate cost function  $R^\phi$  as one of its element. The relation between the VF and the immediate cost function is given by rewriting (4) as

$$R^\phi(x) = [\beta I_{\text{op}} + \mathcal{G}](V^\phi)(x), \quad x \in \text{int}(\mathcal{X}), \quad (6)$$

where  $I_{\text{op}}$  is the identity operator. To define the operator  $\beta I_{\text{op}} + \mathcal{G}$  over the whole  $\mathcal{X}$ , we use the following definition.

**Definition 1** ([52, Definition 1]). *Let  $I_s := \{\alpha := [\alpha^1, \alpha^2, \dots, \alpha^{n_x}]^\top \in \mathbb{Z}_{\geq 0}^{n_x} \mid \sum_{j=1}^{n_x} \alpha^j \leq s\}$  for  $s \in \mathbb{Z}_{\geq 0}$ ,  $n_x \in \mathbb{Z}_{> 0}$ . Define  $D^\alpha \varphi(x) = \frac{\partial^{\sum_{j=1}^{n_x} \alpha^j}}{\partial(x^1)^{\alpha^1} \partial(x^2)^{\alpha^2} \dots \partial(x^{n_x})^{\alpha^{n_x}}} \varphi(x)$ , where  $x := [x^1, x^2, \dots, x^{n_x}]^\top \in \mathbb{R}^{n_x}$ . If  $\mathcal{X} \subset \mathbb{R}^{n_x}$  is compact with nonempty interior  $\text{int}(\mathcal{X})$ ,  $C^s(\text{int}(\mathcal{X}))$  is the space of functions  $\varphi$  over  $\text{int}(\mathcal{X})$  such that  $D^\alpha \varphi$  is well defined and continuous over  $\text{int}(\mathcal{X})$  for each  $\alpha \in I_s$ . Define  $C^s(\mathcal{X})$  to be the space of continuous functions  $\varphi$  over  $\mathcal{X}$  such that  $\varphi|_{\text{int}(\mathcal{X})} \in C^s(\text{int}(\mathcal{X}))$  and that  $D^\alpha(\varphi|_{\text{int}(\mathcal{X})})$  has a continuous extension  $D^\alpha \varphi$  to  $\mathcal{X}$  for each  $\alpha \in I_s$ . If  $\kappa \in C^{2s}(\mathcal{X} \times \mathcal{X})$ , define  $(D^\alpha \kappa)_x(y) = \frac{\partial^{\sum_{j=1}^{n_x} \alpha^j}}{\partial(x^1)^{\alpha^1} \partial(x^2)^{\alpha^2} \dots \partial(x^{n_x})^{\alpha^{n_x}}} \kappa(y, x)$ ,  $\forall x, y \in \text{int}(\mathcal{X})$ .*

Now, suppose that  $\mathcal{H}_V$  is an RKHS associated with the reproducing kernel  $\kappa^V(\cdot, \cdot) \in C^{2 \times 2}(\mathcal{X} \times \mathcal{X})$ . Then, we define the operator  $U : \mathcal{H}_V \rightarrow \mathcal{H}_R := \{\varphi \mid \varphi(x) = U(\varphi^V)(x), \exists \varphi^V \in \mathcal{H}_V, \forall x \in \mathcal{X}\}$  as

$$\begin{aligned} U(\varphi^V)(x) &:= \beta \varphi^V(x) - [D^{e_1} \varphi^V(x), D^{e_2} \varphi^V(x), \dots, D^{e_{n_x}} \varphi^V(x)] h^\phi(x) \\ &\quad - \frac{1}{2} \sum_{m,n=1}^{n_x} A_{m,n}^\phi(x) D^{e_m + e_n} \varphi^V(x), \quad \forall \varphi^V \in \mathcal{H}_V, \forall x \in \mathcal{X}, \end{aligned} \quad (7)$$

where  $A_{m,n}^\phi(x)$  is the  $(m, n)$  entry of  $A^\phi(x)$ . Note here that  $U(\varphi^V)(x) = [\beta I_{\text{op}} + \mathcal{G}](\varphi^V)(x)$  over  $\text{int}(\mathcal{X})$ . We emphasize here that the *expected* value and the immediate cost are related through the *deterministic* operator  $U$ . The following main theorem states that  $\mathcal{H}_R$  is indeed an RKHS under Assumptions 1 and 2, and its corresponding reproducing kernel is obtained.

**Theorem 1.** *Under Assumptions 1 and 2, suppose that  $\mathcal{H}_V$  is an RKHS associated with the reproducing kernel  $\kappa^V(\cdot, \cdot) \in C^{2 \times 2}(\mathcal{X} \times \mathcal{X})$ . Suppose also that (i)  $\beta > 0$ , or that (ii)  $\mathcal{H}_V$  is a Gaussian RKHS, and there exists a point  $x_{t \rightarrow \infty} \in \text{int}(\mathcal{X})$  which is stochastically asymptotically stable over  $\text{int}(\mathcal{X})$ , i.e.,  $P_x \left( \lim_{t \rightarrow \infty} x(t) = x_{t \rightarrow \infty} \right) = 1$  for any starting point  $x \in \text{int}(\mathcal{X})$ . Then, the following statements hold.*

(a) *The space  $\mathcal{H}_R := \{\varphi \mid \varphi(x) = U(\varphi^V)(x), \exists \varphi^V \in \mathcal{H}_V, \forall x \in \mathcal{X}\}$  is an isomorphic Hilbert space of  $\mathcal{H}_V$  equipped with the inner product defined by*

$$\langle \varphi_1, \varphi_2 \rangle_{\mathcal{H}_R} := \langle \varphi_1^V, \varphi_2^V \rangle_{\mathcal{H}_V}, \quad \varphi_i(x) := U(\varphi_i^V)(x), \quad \forall x \in \mathcal{X}, \quad i \in \{1, 2\}, \quad (8)$$

where the operator  $U$  is defined in (7).

(b) *The Hilbert space  $\mathcal{H}_R$  has the reproducing kernel given by*

$$\kappa(x, y) := U(K(\cdot, y))(x), \quad x, y \in \mathcal{X}, \quad (9)$$

where

$$\begin{aligned} K(x, y) &= \beta \kappa^V(x, y) - [(D^{e_1} \kappa^V)_y(x), (D^{e_2} \kappa^V)_y(x), \dots, (D^{e_{n_x}} \kappa^V)_y(x)] h^\phi(y) \\ &\quad - \frac{1}{2} \sum_{m,n=1}^{n_x} A_{m,n}^\phi(y) (D^{e_m + e_n} \kappa^V)_y(x). \end{aligned} \quad (10)$$

*Proof.* See Appendices A and B in the supplementary document.  $\square$

Under Assumptions 1 and 2, Theorem 1 implies that the VF  $V^\phi$  can be uniquely determined by the immediate cost function  $R^\phi$  for a policy  $\phi$  if the VF is in an RKHS of a particular class. In fact, the relation between the VF and the immediate cost function in (4) is based on the assumption that the VF is twice continuously differentiable over  $\text{int}(\mathcal{X})$ , and the verification theorem (cf. [10]) states that, when the immediate cost function and a twice continuously differentiable function satisfying the relation (4) are given under certain conditions, the twice continuously differentiable function is indeed the VF. In Theorem 1, on the other hand, we first restrict the class of the VF by identifying

an RKHS  $\mathcal{H}_V$ , and then approximate the immediate cost function in the RKHS  $\mathcal{H}_R$  any element of which satisfies the relation (4). Because the immediate cost  $R^\phi(x(t))$  is observable, we can employ any kernel-based supervised learning to estimate the function  $R^\phi$  in the RKHS  $\mathcal{H}_R$ , such as GPs and KFs, as elaborated later in Section 6.

In the RKHS  $\mathcal{H}_R$ , an estimate of  $R^\phi$  at time instant  $n \in \mathbb{Z}_{\geq 0}$  is given by  $\hat{R}_n^\phi(x) = \sum_i^r c_i \kappa(x, x_i)$ ,  $c_i \in \mathbb{R}$ ,  $r \in \mathbb{Z}_{\geq 0}$ , where  $\{x_i\}_{i \in \{1, 2, \dots, r\}} \subset \mathcal{X}$  is the set of samples, and the reproducing kernel  $\kappa$  is defined in (9). An estimate of the VF  $V^\phi$  at time instant  $n \in \mathbb{Z}_{\geq 0}$  is thus immediately obtained by  $\hat{V}_n^\phi(x) = U^{-1}(\hat{R}_n^\phi)(x) = \sum_{i=1}^r c_i K(x, x_i)$ , where  $K$  is defined in (10).

Note, when the system dynamics is described by an ordinary differential equation (i.e.,  $\eta = 0$ ), the assumptions that  $V^\phi$  is twice continuously differentiable and that  $\kappa^V(\cdot, \cdot) \in C^{2 \times 2}(\mathcal{X} \times \mathcal{X})$  are relaxed to that  $V^\phi$  is continuously differentiable and that  $\kappa^V(\cdot, \cdot) \in C^{2 \times 1}(\mathcal{X} \times \mathcal{X})$ , respectively.

As an illustrative example of Theorem 1, we show the case of the linear-quadratic regulator (LQR) below.

**Special case: linear-quadratic regulator** Consider a *linear* feedback  $\phi_{\text{LQR}}$ , i.e.,  $\phi_{\text{LQR}}(x) = -F_{\text{LQR}}x$ ,  $F_{\text{LQR}} \in \mathbb{R}^{n_u \times n_x}$ , and a linear system  $\dot{x} := \frac{dx}{dt} = A_{\text{LQR}}x + B_{\text{LQR}}u$ , where  $A_{\text{LQR}} \in \mathbb{R}^{n_x \times n_x}$  and  $B_{\text{LQR}} \in \mathbb{R}^{n_x \times n_u}$  are matrices. In this case, we know that the value function  $V^{\phi_{\text{LQR}}}$  becomes quadratic with respect to the state variable (cf. [53]). Therefore, we employ an RKHS with a quadratic kernel for  $\mathcal{H}_V$ , i.e.,  $\kappa^V(x, y) = (x^\top y)^2$ . If we assume that the input space  $\mathcal{X}$  is so *large* that the set  $\text{span}\{A_{\text{sym}} | A_{\text{sym}} = xx^\top, \exists x \in \mathcal{X}\}$  accommodates any real symmetric matrix, we obtain  $\mathcal{H}_V = \{\mathcal{X} \ni x \mapsto x^\top A_{\text{sym}} x | A_{\text{sym}} \text{ is symmetric}\}$ .

Moreover, it follows from the product rule of the directional derivative [6] that  $K(x, y) = -x^\top \bar{A}_{\text{LQR}} y x^\top y - x^\top y x^\top \bar{A}_{\text{LQR}} y = x^\top (-\bar{A}_{\text{LQR}} y y^\top - y y^\top \bar{A}_{\text{LQR}}) x$ , where  $\bar{A}_{\text{LQR}} := A_{\text{LQR}} - B_{\text{LQR}} F_{\text{LQR}}$ . Note  $A_{\text{value}}(y) := -\bar{A}_{\text{LQR}} y y^\top - y y^\top \bar{A}_{\text{LQR}}$  is symmetric, implying  $K(\cdot, y) \in \mathcal{H}_V$ , and we obtain  $\kappa(x, y) = -x^\top (\bar{A}_{\text{LQR}}^\top A_{\text{value}}(y) + A_{\text{value}}(y) \bar{A}_{\text{LQR}}) x$ . Because  $A_{\text{cost}}(y) := -\bar{A}_{\text{LQR}}^\top A_{\text{value}}(y) - A_{\text{value}}(y) \bar{A}_{\text{LQR}}$  is symmetric, it follows that  $\kappa(\cdot, y) \in \mathcal{H}_V$ . If  $\bar{A}_{\text{LQR}}$  is stable (Hurwitz), from Theorem 1, the one-to-one correspondence between  $\mathcal{H}_V$  and  $\mathcal{H}_R$  thus implies that  $\mathcal{H}_V = \mathcal{H}_R$ . Therefore, we can fully approximate the immediate cost function  $R^{\phi_{\text{LQR}}}$  in  $\mathcal{H}_R$  if  $R^{\phi_{\text{LQR}}}$  is quadratic with respect to the state variable.

Suppose that the immediate cost function is given by  $R^{\phi_{\text{LQR}}}(x) = \sum_{i=1}^r c_i \kappa(x, x_i) = x^\top \bar{A}_{\text{cost}} x$ . Then, the estimated value function will be given by  $V^{\phi_{\text{LQR}}}(x) = U^{-1}(R^{\phi_{\text{LQR}}})(x) = \sum_{i=1}^r c_i K(x, x_i) = -x^\top \bar{A}_{\text{value}} x$ , where  $\bar{A}_{\text{LQR}}^\top \bar{A}_{\text{value}} + \bar{A}_{\text{value}} \bar{A}_{\text{LQR}} + \bar{A}_{\text{cost}} = 0$ , which is indeed the well-known Lyapunov equation [53]. Unlike Gaussian-kernel cases, we only require a *finite* number of parameters to fully approximate the immediate cost function, and hence is analytically obtainable.

**Barrier-certified policy updates under CT formulation** Next, we show that the CT formulation makes barrier-certified policy updates computationally more efficient under certain conditions. Assume that the system dynamics is affine in the control, i.e.,  $h(x, u) = f(x) + g(x)u$ , and  $\eta = 0$ , where  $f: \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$  and  $g: \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x \times n_u}$ , and that the immediate cost  $R(x, u)$  is given by  $Q(x) + \frac{1}{2}u^\top M u$ , where  $Q: \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ , and  $M \in \mathbb{R}^{n_u \times n_u}$  is a positive definite matrix. Then, any Lipschitz continuous policy  $\phi: \mathcal{X} \rightarrow \mathcal{U}$  satisfying  $\phi(x) \in \mathcal{S}(x) := \left\{ u \in \mathcal{U} \mid \frac{\partial b(x)}{\partial x} f(x) + \frac{\partial b(x)}{\partial x} g(x)u + \alpha(b(x)) \geq 0 \right\}$  renders the set  $\mathcal{C}$  forward invariant [50], i.e., the state trajectory remains inside the set  $\mathcal{C}$ , where  $\alpha: \mathbb{R} \rightarrow \mathbb{R}$  is strictly increasing and  $\alpha(0) = 0$ . Taking this constraint into account, the barrier-certified greedy policy update is given by

$$\phi^+(x) = \underset{u \in \mathcal{S}(x)}{\operatorname{argmin}} \left[ \frac{1}{2} u^\top M u + \frac{\partial V^\phi(x)}{\partial x} g(x)u \right], \quad (11)$$

which is, by virtue of the CT formulation, a quadratic programming (QP) problem at  $x$  when  $\mathcal{U} \subset \mathbb{R}^{n_u}$  defines affine constraints (see Appendix C in the supplementary document). The space of allowable policies is thus given by  $\Gamma := \{\phi \in \Phi \mid \phi(x) \in \mathcal{S}(x), \forall x \in \mathcal{X}\}$ . When  $\eta \neq 0$  and the dynamics is learned by GPs as in [30], the work in [47] provides a barrier certificate for uncertain dynamics. Note, one can also employ a function approximator or add noises to the greedily updated policy to avoid

unstable performance and promote exploration (see e.g., [8]). To see if the updated policy  $\phi^+$  remains in the space of Lipschitz continuous policies  $\Phi$ , i.e.,  $\Gamma \subset \Phi$ , we present the following proposition.

**Proposition 1.** *Assume the conditions in Theorem 1. Assume also that  $\mathcal{U} \subset \mathbb{R}^{n_u}$  defines affine constraints, and that  $f, g, \alpha$ , and the derivative of  $b$  are Lipschitz continuous over  $\mathcal{X}$ . Then, the policy  $\phi^+$  defined in (11) is Lipschitz continuous over  $\mathcal{X}$  if the width of a feasible set<sup>3</sup> is strictly larger than zero over  $\mathcal{X}$ .*

*Proof.* See Appendix D in the supplementary document.  $\square$

Note, if  $\mathcal{U} \subset \mathbb{R}^{n_x}$  defines the bounds of each entry of control inputs, it defines affine constraints. Lastly, the width of a feasible set is strictly larger than zero if  $\mathcal{U}$  is sufficiently large and  $\frac{\partial b(x)}{\partial x} g(x) \neq 0$ . We will further clarify the relations of the proposed theoretical framework to existing works below.

## 5 Relations to existing works

First, our proposed framework takes advantage of the capability of learning complicated functions and nonparametric flexibility of RKHSs, and reproduces some of the existing *model-based* DT-VF approximation techniques (see Appendix E in the supplementary document). Note that some of the existing DT-VF approximations in RKHSs, such as GPTD [9], also work for model-free cases (see [27] for model-free adaptive DT action-value function approximation, for example). Second, since the RKHS  $\mathcal{H}_R$  for learning is explicitly defined in our framework, any kernel-based method and its convergence/tracking analyses are directly applicable. While, for example, the work in [17], which aims at attaining a sparse representation of the unknown function in an online fashion in RKHSs, was extended to the policy evaluation [18] by addressing the double-sampling problem, our framework does not suffer from the double-sampling problem, and hence any kernel-based online learning (e.g., [17, 51, 39]) can be straightforwardly applied. Third, when the time interval is small, DT formulations become susceptible to errors, while CT formulations are immune to the choice of the time interval. Note, on the other hand, a larger time interval poorly represents the system dynamics evolving in continuous time. Lastly, barrier certificates are efficiently incorporated in our CT framework through QPs under certain conditions, and state constraints are implicitly taken into account. Stochastic optimal control such as the work in [43, 42] requires sample trajectories over predefined finite time horizons and the value is computed along the trajectories while the VF is estimated in an RKHS even without having to follow the trajectory in our framework.

## 6 Applications and practical implementation

We apply the theory presented in the previous section to the Gaussian kernel case and introduce CTGP as an example, and present a practical implementation. Assume that  $A(x, u) \in \mathbb{R}^{n_x \times n_x}$  is diagonal,

for simplicity. The Gaussian kernel is given by  $\kappa^V(x, y) := \frac{1}{(2\pi\sigma^2)^{L/2}} \exp\left(-\frac{\|x - y\|_{\mathbb{R}^{n_x}}^2}{2\sigma^2}\right)$ ,

$x, y \in \mathcal{X}$ ,  $\sigma > 0$ . Given Gaussian kernel  $\kappa^V(x, y)$ , the reproducing kernel  $\kappa(x, y)$  defined in (9) is derived as  $\kappa(x, y) = a(x, y)\kappa^V(x, y)$ , where  $a : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  is a real-valued function (see Appendix F in the supplementary document for the explicit form of  $a(x, y)$ ).

**CTGP** One of the celebrated properties of GPs is their Bayesian formulation, which enables us to deal with uncertainty through credible intervals. Suppose that the observation  $d$  at time instant  $n \in \mathbb{Z}_{\geq 0}$  contains some noise  $\epsilon \in \mathbb{R}$ , i.e.,  $d(x) = R^\phi(x) + \epsilon$ ,  $\epsilon \sim \mathcal{N}(0, \mu_o^2)$ ,  $\mu_o \geq 0$ . Given data  $d_N := [d(x_0), d(x_1), \dots, d(x_N)]^\top$  for some  $N \in \mathbb{Z}_{\geq 0}$ , we can employ GP regression to obtain the mean  $m(x_*)$  and the variance  $\mu^2(x_*)$  of  $\hat{R}^\phi(x_*)$  at a point  $x_* \in \mathcal{X}$  as

$$m(x_*) = k_*^\top (G + \mu_o^2 I)^{-1} d_N, \quad \mu^2(x_*) = \kappa(x_*, x_*) - k_*^\top (G + \mu_o^2 I)^{-1} k_*, \quad (12)$$

where  $I$  is the identity matrix,  $k_* := [\kappa(x_*, x_0), \kappa(x_*, x_1), \dots, \kappa(x_*, x_N)]^\top$ , and the  $(i, j)$  entry of  $G \in \mathbb{R}^{(N+1) \times (N+1)}$  is  $\kappa(x_{i-1}, x_{j-1})$ . Then, by the existence of the inverse operator  $U^{-1}$ , the mean  $m^V(x_*)$  and the variance  $\mu^{V^2}(x_*)$  of  $\hat{V}^\phi(x_*)$  at a point  $x_* \in \mathcal{X}$  can be given by

$$m^V(x_*) = K_*^{V^\top} (G + \mu_o^2 I)^{-1} d_N, \quad \mu^{V^2}(x_*) = \kappa^V(x_*, x_*) - K_*^{V^\top} (G + \mu_o^2 I)^{-1} K_*^V, \quad (13)$$

<sup>3</sup>*Width* indicates how much control margin is left for the strictest constraint, as defined in [24, Equation 21].

Table 2: Comparisons of the cumulative costs and numbers of times the observed velocities became lower than  $-0.05$  with and without barrier certificates

	CTKF	GPTD_1	DTKF_1	CTGP	GPTD_20	DTKF_20
Cumulative cost	114.2	299.1	299.1	82.2	89.2	90.4
With barrier	0 (times)	0 (times)	0 (times)	0 (times)	0 (times)	0 (times)
Without barrier	0 (times)	0 (times)	0 (times)	10 (times)	20 (times)	20 (times)

where  $K_*^V := [K(x_*, x_0), K(x_*, x_1), \dots, K(x_*, x_N)]^T$  (see Appendix G in the supplementary document for more details).

## 7 Numerical Experiment

In this section, we first show the validity of the proposed CT framework and its advantage over DT counterparts when the time interval is small, and then compare CTGP and GPTD for RL on a simulated inverted pendulum. In both of the experiments, the coherence-based sparsification [33] in the RKHS  $\mathcal{H}_R$  is employed to curb the growth of the dictionary size.

**Policy evaluations: comparison of CT and DT approaches** We show that our CT approaches are advantageous over DT counterparts in terms of susceptibility to errors, by using MountainCarContinuous-v0 in OpenAI Gym [7] as the environment. The state is given by  $x(t) := [x(t), v(t)]^T \in \mathbb{R}^2$ , where  $x(t)$  and  $v(t)$  are the position and the velocity of the car, and the dynamics is given by  $dx = \begin{bmatrix} v(t) \\ -0.0025 \cos(3x(t)) \end{bmatrix} dt + \begin{bmatrix} 0 \\ 0.0015 \end{bmatrix} u(t)dt$ , where  $u(t) \in [-1.0, 1.0]$ .

The position and the velocity are clipped to  $[-0.07, 0.07]$  and  $[-1.2, 0.6]$ , respectively, and the goal is to reach the position  $x = 0.45$ . In the simulation, the control cycle (i.e., the frequency of applying control inputs and observing the states and costs) is set to 1.0 second. The observed immediate cost is given by  $R(x(t), u(t)) + \epsilon = 1 + 0.001u^2(t) + \epsilon$  for  $x(t) < 0.45$  and  $R(x(t), u(t)) + \epsilon = 0.001u^2(t) + \epsilon$  for  $x(t) \geq 0.45$ , where  $\epsilon \sim \mathcal{N}(0, 0.1^2)$ . Note the immediate cost for the DT cases is given by  $(R(x(t), u(t)) + \epsilon)\Delta t$ , where  $\Delta t$  is the time interval. For policy evaluations, we use the policy obtained by RL based on the cross-entropy methods<sup>4</sup>, and the four methods, CTGP, KF-based CT-VF approximation (CTKF), GPTD, and KF-based DT-VF approximation (DTKF), are used to learn value functions associated with the policy by executing the current policy for five episodes, each of which terminates whenever  $t = 300$  or  $x(t) \geq 0.45$ . GP-based techniques are expected to handle the random component  $\epsilon$  added to the immediate cost. The new policies are then obtained by the barrier-certified policy updates under CT formulations, and these policies are evaluated for five times. Here, the barrier function is given by  $b(x) = 0.05 + v$ , which prevents the velocity from becoming lower than  $-0.05$ . Figure 3 compares the value functions<sup>5</sup> learned by each method for the time intervals  $\Delta t = 20.0$  and  $\Delta t = 1.0$ . We observe that the DT approaches are sensitive to the choice of  $\Delta t$ . Table 2 compares the cumulative costs averaged over five episodes for each method and for different time intervals and the numbers of times we observed the velocity being lower than  $-0.05$  when the barrier certificate is employed and unemployed. (Numbers associated with the algorithm names indicate the lengths of the time intervals.) Note that the cumulative costs are calculated by summing up the immediate costs multiplied by the duration of each control cycle, i.e., we discretized the immediate cost based on the control cycle. It is observed that the CT approach is immune to the choice of  $\Delta t$  while the performance of the DT approach degrades when the time interval becomes small, and that the barrier-certified policy updates work efficiently.

**Reinforcement learning: inverted pendulum** We show the advantage of CTGP over GPTD when the time interval for the estimation is small. Let the state  $x(t) := [\theta(t), \omega(t)]^T \in \mathbb{R}^2$  consists of the angle  $\theta(t)$  and the angular velocity  $\omega(t)$  of an inverted pendulum, and we consider the dynamics:

$$dx = \begin{bmatrix} \omega(t) \\ \frac{g}{\ell} \sin(\theta(t)) - \frac{\rho}{m\ell^2} \omega(t) \end{bmatrix} dt + \begin{bmatrix} 0 \\ \frac{1}{m\ell^2} \end{bmatrix} u(t)dt + 0.01I dw, \text{ where } g = 9.8, m = 1, \ell = 1, \rho = 0.01. \text{ The Brownian motion may come from outer disturbances and/or modeling error. In the simulation, the time interval } \Delta t \text{ is set to 0.01 seconds, and the simulated dynamics evolves}$$

<sup>4</sup>We used the code in <https://github.com/udacity/deep-reinforcement-learning/blob/master/cross-entropy/CEM.ipynb> offered by Udacity. The code is based on PyTorch [31].

<sup>5</sup>We used "jet colormaps" in Python Matplotlib for illustrating the value functions.



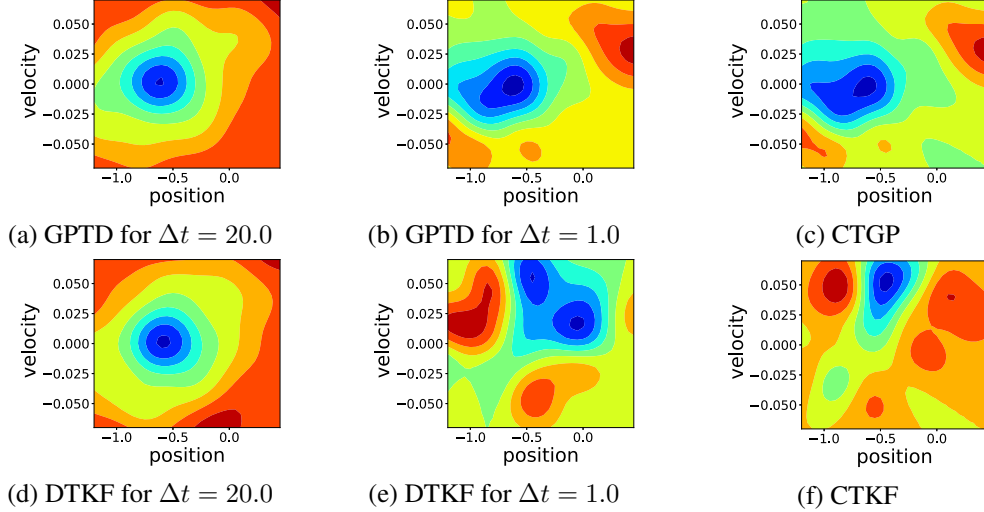


Figure 3: Illustrations of the value functions obtained by CTGP, CTKF, GPTD, and DTKF for time intervals  $\Delta t = 20.0$  and  $\Delta t = 1.0$ . The policy is obtained by RL based on the cross-entropy method. CT approaches are not affected by the choice of  $\Delta t$ .

by  $\Delta x = h(x(t), u(t))\Delta t + \sqrt{\Delta t}\eta(x(t), u(t))\epsilon_w$ , where  $\epsilon_w \sim \mathcal{N}(0, I)$ . In this experiment, the task is to stabilize the inverted pendulum at  $\theta = 0$ . The observed immediate cost is given by  $R(x(t), u(t)) + \epsilon = 1/(1 + e^{-10(\theta(t) - \pi/16)}) + 100/(1 + e^{-10(\theta(t) - \pi/6)}) + 0.05u^2(t) + \epsilon$ , where  $\epsilon \sim \mathcal{N}(0, 0.1^2)$ . A trajectory associated with the current policy is generated to learn the VF. The trajectory terminates when  $|\theta(t)| > \pi/4$  and restarts from a random initial angle. After 10 seconds, the policy is updated. To evaluate the current policy, average time over five episodes in which the pendulum stays up ( $|\theta(t)| \leq \pi/4$ ) when initialized as  $\theta(0) \in [-\pi/6, \pi/6]$  is used. Figure 4 compares this average time of CTGP and GPTD up to five updates with standard deviations until when stable policy improvement becomes difficult without some heuristic techniques such as adding noises to policies. Note that the same seed for the random number generator is used for the initializations of both of the two approaches. It is observed that GPTD fails to improve policies. The large credible interval of CTGP is due to the random initialization of the state.

## 8 Conclusion and future work

We presented a novel theoretical framework that renders the CT-VF approximation problem simultaneously solvable in an RKHS by conducting kernel-based supervised learning for the immediate cost function in the properly defined RKHS. Our CT framework is compatible with rich theories of control, including control barrier certificates for safety-critical applications. The validity of the proposed framework and its advantage over DT counterparts when the time interval is small were verified experimentally on the classical Mountain Car problem and a simulated inverted pendulum.

There are several possible directions to explore as future works; First, we can employ the state-of-the-art kernel methods within our theoretical framework or use other variants of RL, such as actor-critic methods, to improve practical performances. Second, we can consider uncertainties in value function approximation by virtue of the RKHS-based formulation, which might be used for safety verifications. Lastly, it is worth further explorations of advantages of CT formulations for physical tasks.

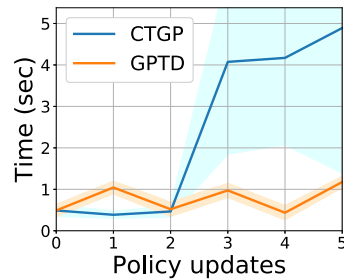


Figure 4: Comparison of time up to which the pendulum stays up between CTGP and GPTD for the inverted pendulum ( $\pm$  std. deviation).

## Acknowledgments

This work was partially conducted when M. Ohnishi was at the GRITS Lab, Georgia Institute of Technology; M. Ohnishi thanks the members of the GRITS Lab, including Dr. Li Wang, and Prof. Magnus Egerstedt for discussions regarding barrier functions. M. Yukawa was supported in part by KAKENHI 18H01446 and 15H02757, M. Johansson was supported in part by the Swedish Research Council and by the Knut and Alice Wallenberg Foundation, and M. Sugiyama was supported in part by KAKENHI 17H00757. Lastly, the authors thank all of the anonymous reviewers for their very insightful comments.

## References

- [1] A. Agrawal and K. Sreenath. “Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation”. In: *Proc. RSS*. 2017.
- [2] A. D. Ames et al. “Control barrier function based quadratic programs with application to automotive safety systems”. In: *arXiv preprint arXiv:1609.06408* (2016).
- [3] N. Aronszajn. “Theory of reproducing kernels”. In: *Trans. Amer. Math. Soc.* 68.3 (1950), pp. 337–404.
- [4] L. Baird. “Reinforcement learning in continuous time: Advantage updating”. In: *Proc. IEEE ICNN*. Vol. 4. 1994, pp. 2448–2453.
- [5] L. Baird. “Residual algorithms: Reinforcement learning with function approximation”. In: *Proc. ICML*. 1995, pp. 30–37.
- [6] J. Bonet and R. D. Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge University Press, 1997.
- [7] G. Brockman et al. “OpenAI Gym”. In: *arXiv preprint arXiv:1606.01540* (2016).
- [8] K. Doya. “Reinforcement learning in continuous time and space”. In: *Neural Computation* 12.1 (2000), pp. 219–245.
- [9] Y. Engel, S. Mannor, and R. Meir. “Reinforcement learning with Gaussian processes”. In: *Proc. ICML*. 2005, pp. 201–208.
- [10] W. H. Fleming and H. M. Soner. *Controlled Markov processes and viscosity solutions*. Vol. 25. Springer Science & Business Media, 2006.
- [11] J. Garcia and F. Fernández. “A comprehensive survey on safe reinforcement learning”. In: *J. Mach. Learn. Res.* 16.1 (2015), pp. 1437–1480.
- [12] P. Glotfelter, J. Cortés, and M. Egerstedt. “Nonsmooth barrier functions with applications to multi-robot systems”. In: *IEEE Control Systems Letters* 1.2 (2017), pp. 310–315.
- [13] S. Grunewalder et al. “Modelling transition dynamics in MDPs with RKHS embeddings”. In: *Proc. ICML*. 2012.
- [14] H. K. Khalil. “Nonlinear systems”. In: *Prentice-Hall* 3 (1996).
- [15] R. Khasminskii. *Stochastic stability of differential equations*. Vol. 66. Springer Science & Business Media, 2011.
- [16] V. R. Konda, J. N. Tsitsiklis, et al. “Convergence rate of linear two-time-scale stochastic approximation”. In: *The Annals of Applied Probability* 14.2 (2004), pp. 796–819.
- [17] A. Koppel et al. “Parsimonious online learning with kernels via sparse projections in function space”. In: *Proc. IEEE ICASSP*. 2017, pp. 4671–4675.
- [18] A. Koppel et al. “Policy evaluation in continuous MDPs with efficient kernelized gradient temporal difference”. In: *IEEE Trans. Automatic Control* (submitted) (2017).
- [19] N. V. Krylov. *Controlled diffusion processes*. Vol. 14. Springer Science & Business Media, 2008.
- [20] F. L. Lewis and D. Vrabie. “Reinforcement learning and adaptive dynamic programming for feedback control”. In: *IEEE Circuits and Systems Magazine* 9.3 (2009).
- [21] D. Liberzon. *Calculus of variations and optimal control theory: a concise introduction*. Princeton University Press, 2011.
- [22] W. Liu, J. Príncipe, and S. Haykin. *Kernel adaptive filtering*. New Jersey: Wiley, 2010.
- [23] H. Q. Minh. “Some properties of Gaussian reproducing kernel Hilbert spaces and their implications for function approximation and learning theory”. In: *Constructive Approximation* 32.2 (2010), pp. 307–338.

- [24] B. Morris, M. J. Powell, and A. D. Ames. “Sufficient conditions for the Lipschitz continuity of QP-based multi-objective control of humanoid robots”. In: *Proc. CDC*. 2013, pp. 2920–2926.
- [25] R. Munos and P. Bourguine. “Reinforcement learning for continuous stochastic control problems”. In: *Proc. NIPS*. 1998, pp. 1029–1035.
- [26] Y. Nishiyama et al. “Hilbert space embeddings of POMDPs”. In: *arXiv preprint arXiv:1210.4887* (2012).
- [27] M. Ohnishi et al. “Barrier-certified adaptive reinforcement learning with applications to brushbot navigation”. In: *arXiv preprint arXiv:1801.09627* (2018).
- [28] B. Øksendal. *Stochastic differential equations*. Springer, 2003.
- [29] D. Ormoneit and Š. Sen. “Kernel-based reinforcement learning”. In: *Machine Learning* 49.2 (2002), pp. 161–178.
- [30] Y. Pan and E. Theodorou. “Probabilistic differential dynamic programming”. In: *Proc. NIPS*. 2014, pp. 1907–1915.
- [31] A. Paszke et al. “Automatic differentiation in PyTorch”. In: (2017).
- [32] C. E. Rasmussen and C. K. Williams. *Gaussian processes for machine learning*. Vol. 1. MIT press Cambridge, 2006.
- [33] C. Richard, J. Bermudez, and P. Honeine. “Online prediction of time series data with kernels”. In: *IEEE Trans. Signal Process.* 57.3 (2009), pp. 1058–1067.
- [34] I. Steinwart. “On the influence of the kernel on the consistency of support vector machines”. In: *J. Mach. Learn. Res.* 2 (2001), pp. 67–93.
- [35] M. Sugiyama. *Statistical reinforcement learning: modern machine learning approaches*. CRC Press, 2015.
- [36] W. Sun and J. A. Bagnell. “Online Bellman residual and temporal difference algorithms with predictive error guarantees”. In: *Proc. IJCAI*. 2016.
- [37] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT Press, 1998.
- [38] R. S. Sutton, H. R. Maei, and C. Szepesvári. “A convergent  $O(n)$  temporal-difference algorithm for off-policy learning with linear function approximation”. In: *Advances in Neural Information Processing Systems*. 2009, pp. 1609–1616.
- [39] M. Takizawa and M. Yukawa. “Adaptive nonlinear estimation based on parallel projection along affine subspaces in reproducing kernel Hilbert space”. In: *IEEE Trans. Signal Processing* 63.16 (2015), pp. 4257–4269.
- [40] Y. Tassa et al. “DeepMind Control Suite”. In: *arXiv preprint arXiv:1801.00690* (2018).
- [41] G. Taylor and R. Parr. “Kernelized value function approximation for reinforcement learning”. In: *Proc. ICML*. 2009, pp. 1017–1024.
- [42] E. Theodorou, J. Buchli, and S. Schaal. “Reinforcement learning of motor skills in high dimensions: A path integral approach”. In: *Proc. IEEE ICRA*. 2010, pp. 2397–2403.
- [43] E. Theodorou, Y. Tassa, and E. Todorov. “Stochastic differential dynamic programming”. In: *Proc. IEEE ACC*. 2010, pp. 1125–1132.
- [44] J. N. Tsitsiklis and B. Van R. “Analysis of temporal-difference learning with function approximation”. In: *Proc. NIPS*. 1997, pp. 1075–1081.
- [45] K. G. Vamvoudakis and F. L. Lewis. “Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem”. In: *Automatica* 46.5 (2010), pp. 878–888.
- [46] L. Wang, A. D. Ames, and M. Egerstedt. “Safety barrier certificates for collisions-free multi-robot systems”. In: *IEEE Trans. Robotics* (2017).
- [47] L. Wang, E. A. Theodorou, and M. Egerstedt. “Safe learning of quadrotor dynamics using barrier certificates”. In: *arXiv preprint arXiv:1710.05472* (2017).
- [48] P. Wieland and F. Allgöwer. “Constructive safety using control barrier functions”. In: *Proc. IFAC* 40.12 (2007), pp. 462–467.
- [49] X. Xu, D. Hu, and X. Lu. “Kernel-based least squares policy iteration for reinforcement learning”. In: *IEEE Trans. Neural Networks* 18.4 (2007), pp. 973–992.
- [50] X. Xu et al. “Robustness of control barrier functions for safety critical control”. In: *Proc. IFAC* 48.27 (2015), pp. 54–61.
- [51] M. Yukawa. “Multikernel Adaptive Filtering”. In: *IEEE Trans. Signal Processing* 60.9 (2012), pp. 4672–4682.

- [52] DX. Zhou. “Derivative reproducing properties for kernel methods in learning theory”. In: *Journal of Computational and Applied Mathematics* 220.1-2 (2008), pp. 456–463.
- [53] K. Zhou, J. C. Doyle, K. Glover, et al. *Robust and optimal control*. Vol. 40. Prentice Hall, 1996.