2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

Stochastic Optimal Control via Hilbert Space Embeddings of Distributions

Adam J. Thorpe, *Student Member, IEEE,* Meeko M. K. Oishi, *Senior Member, IEEE*

2021 60th IEEE Conference on Decision and Control (CDC) | 978-1-6654-3659-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/CDC45484.2021.9682801

***Abstract*— Kernel embeddings of distributions have recently gained significant attention in the machine learning community as a data-driven technique for representing probability distributions. Broadly, these techniques enable efficient computation of expectations by representing integral operators as elements in a reproducing kernel Hilbert space. We apply these techniques to the area of stochastic optimal control theory and present a method to compute approximately optimal policies for stochastic systems with arbitrary disturbances. Our approach reduces the optimization problem to a linear program, which can easily be solved via the Lagrangian dual, without resorting to gradient-based optimization algorithms. We focus on discrete-time dynamic programming, and demonstrate our proposed approach on a linear regulation problem, and on a nonlinear target tracking problem. This approach is broadly applicable to a wide variety of optimal control problems, and provides a means of working with stochastic systems in a data-driven setting.**

I. Introduction

Stochastic systems are ubiquitous, however most methods for control of stochastic systems are reliant upon accurate modeling not only of the dynamics, but also of the stochastic processes of the system. As autonomous systems become commonplace, and direct human interaction with autonomy become more pervasive, presumptions of linearity and Gaussian stochasticity become questionable, as they could lead to control solutions that are confusing, non-intuitive, or simply incorrect. Robust solutions that work well when uncertainty is bounded by known values may be excessively conservative, and cannot accommodate long-tail phenomena. In contrast, data-driven approaches do not rely upon any prior assumptions on the dynamics or stochasticity of the system. Data-driven approaches have garnered considerable interest recently, due to the capabilities of learning algorithms to handle systems with nonlinear dynamics and unknown disturbances.

This material is based upon work supported by the National Science Foundation under NSF Grant Number CNS-1836900. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The NASA University Leadership initiative (Grant #80NSSC20M0163) provided funds to assist the authors with their research, but this article solely reflects the opinions and conclusions of its authors and not any NASA entity. This research was supported in part by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. The views expressed in this article do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

A. Thorpe and M. Oishi are with Electrical & Computer Eng., University of New Mexico, Abq., NM. Email: {ajthor,oishi}@unm.edu.

We propose a method for data-driven controller synthesis based on *conditional distribution embeddings* [1], a nonparametric learning technique that uses a sample of system observations to construct a model of the stochastic system dynamics as an element in a high-dimensional Hilbert space of functions known as a *reproducing kernel Hilbert space*. These techniques leverage functional analysis and statistical learning theory to empirically estimate the stochastic kernel using data. As a nonparametric technique, kernel methods are inherently data-driven, and do not rely upon prior assumptions placed upon the data or exploit system structure. These techniques have been applied to Markov models [2], partially-observable systems [3, 4], and more recently, to robust optimization approaches [5, 6]. Furthermore, these techniques admit finite sample bounds which show convergence in probability as the number of samples tend to infinity [1]. A Hilbert space framework is particularly well-suited to stochastic optimal control problems [7], primarily because Hilbert spaces are a generalization of inner product spaces to an infinite-dimensional setting, meaning they encompass many optimization problems of interest (note that $\mathbb{R}^n$ is a Hilbert space).

The use of kernel methods for policy synthesis is well-motivated in literature, especially in the area of reinforcement learning (RL) [8]. Methods have been developed to optimize a policy in an RKHS via functional gradient descent [9], [10]. Other approaches rely upon value iteration, approximating the value function as an intermediate step in order to compute an optimal control input [2]. However, most of these approaches face significant computational challenges due to the sampling schemes used by RL, the need for knowledge of a gradient, or reliance upon iterative numerical methods. Some progress has been made to alleviate these issues, for example using stochastic factorization [11].

Our main contribution is a data-driven algorithm for computing approximately optimal policies for arbitrary discrete-time stochastic dynamical systems. The novelty of our approach is the use of conditional distribution embeddings to formulate an optimal control problem as a linear program within a reproducing kernel Hilbert space, which can be solved efficiently via the Lagrangian dual. Our approach is model-free, since it relies only upon data collected from prior observations of the system execution, meaning that it is amenable to systems with arbitrary disturbances and nonlinear dynamics. Because we formulate the optimal control problem as a linear program, we do not rely upon gradient-based algorithms to compute an optimal solution, and thus do not impose a specific structure on the policy for the purpose of computing a functional gradient. The main

difficulty associated with our approach is the dependence of the computational complexity on sample size (generally $\mathcal{O}(M^3)$), as with all kernel based approaches. This arises from the presence of a matrix inverse operation and the large number of observations needed to fully characterize the stochasticity of a system. Fortunately, numerous approaches to reducing the computational burden of kernel methods have been explored, such as [12, 13], which use Fourier transforms and Gaussian matrix approximations to reduce the computational complexity to log-linear time.

The paper is structured as follows: In section II, we define the problem and describe the preliminary theory of embedding distributions in reproducing kernel Hilbert spaces in section III. We then present our method in section IV to compute the optimal policy and present an extension of our proposed approach to solve dynamic programming problems over a finite time horizon. In section V, we demonstrate our proposed approach on a simple stochastic integrator system for the purpose of validation against a known result, and then on a target tracking problem using nonlinear, nonholonomic vehicle dynamics. Concluding remarks are presented in section VI.

II. Preliminaries

We use the following notation throughout: Let E be an arbitrary nonempty space, and denote the σ-algebra on E by $\mathcal{E}$. If E is a topological space [14], the σ-algebra generated by the set of all open subsets of E is called the Borel σ-algebra, denoted by $\mathscr{B}(E)$. Let $(\Omega, \mathcal{F}, \mathbb{P})$ denote a probability space, where $\mathcal{F}$ is the σ-algebra on Ω and $\mathbb{P} : \mathcal{F} \to [0,1]$ is a *probability measure* on the measurable space $(\Omega, \mathcal{F})$. A measurable function $X : \Omega \to E$ is called a *random variable* taking values in $(E, \mathcal{E})$. The image of $\mathbb{P}$ under X, $\mathbb{P}(X^{-1}A)$, $A \in \mathcal{E}$ is called the *distribution* of X. Let T be an arbitrary set, and for each $t \in \mathcal{T}$, let X_t be a random variable. The collection of E-valued random variables $\{X_t : t \in \mathcal{T}\}$ is a *stochastic process*. We define a *stochastic kernel* according to [14].

Definition 1 (Stochastic Kernel)**.** *Let $(E, \mathcal{E})$ and $(F, \mathcal{F})$ be measurable spaces with σ-algebras $\mathcal{E}$ and $\mathcal{F}$, respectively. A stochastic kernel is a map $\kappa : \mathcal{F} \times E \to [0,1]$, where: 1) $x \mapsto \kappa(B \,|\, x)$ is $\mathcal{E}$-measurable for all $B \in \mathcal{F}$; 2) $B \mapsto \kappa(B \,|\, x)$ is a probability measure on $(F, \mathcal{F})$ for all $x \in E$.*

A. System Model

Consider a Markov control process, which is defined in [15] as a 3-tuple, $(\mathcal{X}, \mathcal{U}, Q)$, consisting of:

- A Borel space $\mathcal{X} \subseteq \mathbb{R}^n$ called the state space;
- A compact Borel space $\mathcal{U} \subset \mathbb{R}^m$ called the control space; and
- A stochastic kernel $Q : \mathscr{B}(\mathcal{X}) \times \mathcal{X} \times \mathcal{U} \to [0,1]$ that assigns a probability measure $Q(\cdot \,|\, x, u)$ to each $(x, u) \in \mathcal{X} \times \mathcal{U}$ on the measurable space $(\mathcal{X}, \mathscr{B}(\mathcal{X}))$.

The system evolves from an initial condition $x_0 \in \mathcal{X}$, which may be chosen from an initial distribution $\mathbb{P}_0$ on $\mathcal{X}$, over a finite time horizon $t = 0, 1, \ldots, N$, $N \in \mathbb{N}_+$. As the system evolves, the control actions $u_0, u_1, \ldots, u_{N-1}$ are chosen from a Markov control policy π.

Definition 2 (Markov Policy, [16, Definition 8.2])**.** *A Markov policy π is a sequence $\pi = \{\pi_0, \pi_1, \ldots \pi_{N-1}\}$ of universally measurable stochastic kernels, where for each $t = 0, 1, \ldots, N-1$, the stochastic kernel $\pi_t : \mathscr{B}(\mathcal{U}) \times \mathcal{X} \to [0,1]$ assigns a probability measure $\pi_t(\cdot \,|\, x)$ to every $x \in \mathcal{X}$ on the measurable space $(\mathcal{U}, \mathscr{B}(\mathcal{U}))$.*

B. Problem Formulation

We assume that the stochastic kernel Q is unknown, but that a sample of observations of the system evolution is available.

Assumption 1. *We assume that Q is unknown, but that a sample $\mathcal{S} = \{(x_i, u_i, x_i')\}_{i=1}^M$ of size $M \in \mathbb{N}_+$ is available, where $x_i' \sim Q(\cdot \,|\, x_i, u_i)$ and u_i is selected randomly from the set of admissible control inputs.*

Consider an arbitrary cost function $c : \mathcal{X} \to \mathbb{R}$, which we assume is a continuous, bounded functional that lies in a Hilbert space of functions $\mathscr{H}$. At any time instant t, we seek to minimize c by selecting the *distribution* π_t on $(\mathcal{U}, \mathscr{B}(\mathcal{U}))$ which minimizes the following unconstrained minimization problem:

$$\min_{\pi_t} \quad J_t(\pi) = \int_{\mathcal{U}} \int_{\mathcal{X}} c(y) Q(\mathrm{d}y \,|\, x, v) \pi_t(\mathrm{d}v \,|\, x) \tag{1}$$

The primary difficulty in solving (1) is that without knowledge of Q, the integral in (1) is intractable. Thus, we seek to form an approximate optimization problem by approximating the integral in (1) using a sample $\mathcal{S}$ taken i.i.d. from Q as an element in a Hilbert space of functions. By optimizing the approximate problem, we obtain an approximate solution. Thus, we additionally seek to ensure that the approximate optimization problem converges in probability to the true optimization problem as the sample size increases.

According to [16], in most cases, the optimal Markov policy for a system can be viewed as nonrandomized, or deterministic, meaning the stochastic kernel assigns a probability measure with mass one at a single element in $\mathcal{U}$ to each $x \in \mathcal{X}$. According to [15, 16], the set of nonrandomized policies is a subset of the set of all randomized policies, meaning we can search among the class of randomized policies in Hilbert space to find an optimal policy which minimizes (1).

III. Embedding Stochastic Kernels in an RKHS

Let $\mathscr{H}$ be a Hilbert space of functions of the form $\mathcal{X} \to \mathbb{R}$ with inner product $\langle \cdot, \cdot \rangle_{\mathscr{H}}$ and the induced norm $\|\cdot\|_{\mathscr{H}}$.

Definition 3 (RKHS, [17])**.** *A Hilbert space $\mathscr{H}$ is a reproducing kernel Hilbert space (RKHS) if there exists a positive definite [18, Definition 4.12] kernel function k that satisfies the following properties:*

$$k(x, \cdot) \in \mathscr{H}, \qquad \forall x \in \mathcal{X} \tag{2}$$

$$f(x) = \langle f, k(x, \cdot) \rangle_{\mathscr{H}}, \qquad \forall f \in \mathscr{H}, x \in \mathcal{X} \tag{3}$$

where (3) *is known as the reproducing property, and for any* $x, x' \in \mathcal{X}$*, we denote* $k(x, \cdot) \in \mathscr{H}$ *as a function on* $\mathcal{X}$ *such that* $x' \mapsto k(x, x')$*.*

Remark 1. *Alternatively, by the Moore-Aronszajn theorem [17], we can define an RKHS by first specifying a kernel* k *and obtain a corresponding RKHS as the closure of the span of kernel functions.*

Given $(x, u) \in \mathcal{X} \times \mathcal{U}$, let $Q(\cdot \,|\, x, u)$ be a conditional probability measure on $\mathcal{X}$. According to [1], if the following sufficient condition holds:

$$\int_{\mathcal{X}} \sqrt{k(y, y)} Q(\mathrm{d}y \,|\, x, u) < \infty \tag{4}$$

then there exists an element $m(x, u) \in \mathscr{H}$ called a *conditional distribution embedding*, where

$$m(x, u) := \int_{\mathcal{X}} k(y, \cdot) Q(\mathrm{d}y \,|\, x, u) \tag{5}$$

By the reproducing property of k in $\mathscr{H}$, for any $f \in \mathscr{H}$, we can evaluate the integral with respect to $Q(\cdot \,|\, x, u)$ as an inner product with the embedding $m(x, u)$:

$$\langle f, m(x, u) \rangle_{\mathscr{H}} = \left\langle f, \int_{\mathcal{X}} k(y, \cdot) Q(\mathrm{d}y \,|\, x, u) \right\rangle_{\mathscr{H}} \tag{6}$$

$$= \int_{\mathcal{X}} \langle f, k(y, \cdot) \rangle_{\mathscr{H}} Q(\mathrm{d}y \,|\, x, u) \tag{7}$$

$$= \int_{\mathcal{X}} f(y) Q(\mathrm{d}y \,|\, x, u) \tag{8}$$

Intuitively, the element $m \in \mathscr{H}$ corresponds to the dynamics of the system at the point (x, u). In other words, if the integral exists, then we can embed the integral operator with respect to the probability measure in $\mathscr{H}$ and evaluate the integral via the reproducing property of k in $\mathscr{H}$.

However, in a data-driven setting, the stochastic kernel Q is unknown, which means the embedding $m(x, u)$ is also unknown. Instead, we can empirically estimate the stochastic kernel using a sample of observations taken from Q.

A. Empirical Embeddings Using Observations

Consider a sample $\mathcal{S} = \{(x_i, u_i, {x_i}')\}_{i=1}^{M}$ of size $M \in \mathbb{N}_+$, taken i.i.d. from Q, where x_i and u_i are taken randomly from the state and control spaces $\mathcal{X}$ and $\mathcal{U}$, respectively, and ${x_i}' \sim Q(\cdot \,|\, x, u)$. As shown in [19], we can compute an empirical estimate $\hat{m}$ of m as the solution to a regularized least-squares problem, given by:

$$\min_{\hat{m}} \frac{1}{M} \sum_{i=1}^{M} \| k({x_i}', \cdot) - \hat{m}(x_i, u_i) \|_{\mathscr{H}}^2 + \lambda \| \hat{m} \|_{\mathscr{Q}}^2 \tag{9}$$

where $\lambda > 0$ is the regularization parameter and $\mathscr{Q}$ is a *vector-valued* RKHS [19]. As shown in [19, 20], by the representer theorem, the solution $\hat{m}$ to (9) is unique and has the following form:

$$\hat{m}(x, u) = \sum_{i=1}^{M} \beta_i(x, u) k({x_i}', \cdot) \tag{10}$$

where $\beta(x, u) \in \mathbb{R}^M$ is a vector of real-valued coefficients that depends on the conditioning variables x and u. The problem in (9) admits a closed-form solution, given by:

$$\hat{m}(x, u) = \Phi^\top (\Psi \Psi^\top + \lambda M I)^{-1} \Psi k(x, \cdot) l(u, \cdot) \tag{11}$$

where Φ and Ψ are called *feature vectors*, with elements given by $\Phi_i = k({x_i}', \cdot)$ and $\Psi_i = k(x_i, \cdot) l(u_i, \cdot)$, respectively, where $l : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ is a kernel on $\mathcal{U}$. For simplicity, we denote $W := (\Psi \Psi^\top + \lambda M I)^{-1}$ and let $\beta(x, u) = W \Psi k(x, \cdot) l(u, \cdot)$, such that $\hat{m}(x, u) = \Phi^\top \beta(x, u)$. Using $\hat{m}(x, u)$, we can approximate the expectation with respect to $Q(\cdot \,|\, x, u)$ for any $f \in \mathscr{H}$ as:

$$\langle f, \hat{m}(x, u) \rangle_{\mathscr{H}} \approx \int_{\mathcal{X}} f(y) Q(\mathrm{d}y \,|\, x, u) \tag{12}$$

Further, if the kernel function k is *universal* [21], then the embedding is injective, meaning there exists a unique representation of the distribution in $\mathscr{H}$. In short, a universal kernel k allows us to approximate any arbitrary real-valued function using (10) arbitrarily well as the number of samples tends to infinity. A commonly used kernel function which satisfies this property is the Gaussian kernel $k(x, x') = \exp(-\|x - x'\|_2^2 / 2\sigma^2)$, $\sigma > 0$.

Additionally, the estimate converges in probability to the true embedding as the number of samples M tends to infinity and λ tends to zero (see, e.g. [1, 19, 22]). As shown by [1, Theorem 6], the estimate $\hat{m}(x, u)$ converges in probability to the true embedding $m(x, u)$ in the RKHS norm at a rate $\mathcal{O}_p((M\lambda)^{-1/2} + \lambda^{1/2})$, which means the estimate $\hat{m}$ is a *consistent* estimator of the true embedding, and the integral of a function $f \in \mathscr{H}$ with respect to Q converges in probability to the true result as the sample size increases.

IV. Policy Optimization in Hilbert Space

Consider the problem in (1) where $c \in \mathscr{H}$. As shown in [1], if the sufficient condition in (4) holds, then there exists a conditional distribution embedding $m(x, u)$ such that for any $c \in \mathscr{H}$,

$$\int_{\mathcal{X}} c(y) Q(\mathrm{d}y \,|\, x, u) = \langle c, m(x, u) \rangle_{\mathscr{H}} \tag{13}$$

This allows us to evaluate the expected cost at a particular $(x, u) \in \mathcal{X} \times \mathcal{U}$ as an inner product in Hilbert space. Let π be a Markov policy as in Definition 2. Taking the integral of (13) with respect to the Markov policy π, we obtain the objective function $J_t(\pi)$ in (1). By linearity of the integral and the inner product, we can rewrite the objective using the inner product in (13) to obtain:

$$J_t(\pi) = \left\langle c, \int_{\mathcal{U}} m(x, v) \pi_t(\mathrm{d}v \,|\, x) \right\rangle_{\mathscr{H}} \tag{14}$$

where the integral term on the right hand side of (14) can be interpreted as a representation in $\mathscr{H}$ of the closed-loop dynamics under a policy π.

However, the integral in (14) is intractable, since according to Assumption 1, the stochastic kernel Q (and thus the embedding m) is unknown. Instead, we compute an

empirical estimate $\hat{m}$ of m using a sample $\mathcal{S}$ taken i.i.d. from Q. Recall from (11) that the empirical estimate has the form $\hat{m}(x,u) = \Phi^\top W \Psi k(x,\cdot) l(u,\cdot)$. We then substitute the estimate for the true embedding to approximate the integral in (14).

$$\begin{aligned}\int_{\mathcal{U}} m(x,v)\pi_t(\mathrm{d}v \,|\, x) &\approx \int_{\mathcal{U}} \hat{m}(x,v)\pi_t(\mathrm{d}v \,|\, x) \\ &= \Phi^\top W \Psi k(x,\cdot) \int_{\mathcal{U}} l(v,\cdot)\pi_t(\mathrm{d}v \,|\, x)\end{aligned} \tag{15}$$

Recall that the policy is a collection of stochastic kernels indexed by time, which means that at a given time t, the policy can be represented by a conditional distribution embedding. Thus, it is natural to consider the policy at a particular time as a collection of elements in an RKHS parameterized by $x \in \mathcal{X}$, which admits a representation in terms of finite support. Let $\{\tilde{u}_j\}_{j=1}^P$ be a collection of admissible control inputs. We propose the following representation for the policy π_t:

$$\hat{p}_t(x) = \sum_{j=1}^{P} \alpha_j(x) l(\tilde{u}_j, \cdot) \tag{16}$$

where $\alpha(x) \in \mathbb{R}^P$ is a vector of real-valued coefficients that depends on $x \in \mathcal{X}$. Using (16), we can approximate (15) as:

$$\Phi^\top W \Psi k(x,\cdot) \int_{\mathcal{U}} l(v,\cdot)\pi_t(\mathrm{d}v \,|\, x) \approx \Phi^\top W \Psi k(x,\cdot) \Upsilon^\top \alpha(x) \tag{17}$$

where Υ is a feature vector with elements $\Upsilon_j = l(\tilde{u}_j, \cdot)$. Thus, we can approximate the objective function $J_t(\pi)$ by:

$$\int_{\mathcal{U}} \int_{\mathcal{X}} c(y) Q(\mathrm{d}y \,|\, x, v)\pi_t(\mathrm{d}v \,|\, x) \approx \boldsymbol{c}^\top W \Psi k(x,\cdot) \Upsilon^\top \alpha(x) \tag{18}$$

where $\boldsymbol{c}$ is a vector with elements $\boldsymbol{c}_i = c(x_i{}')$. Thus, we form an approximation of the objective in (1), which converges in probability to the true optimization problem as the sample sizes M and P increase [1]. Now, instead of minimizing over the distribution π_t, we can view the approximate optimization problem as finding $\alpha(x) \in \mathbb{R}^P$ which minimizes (18). However, minimizing $\alpha(x)$ in (18) is unbounded below, which makes the problem unsolvable. As such, additional constraints are required to ensure that the problem admits a feasible solution. Note that intuitively, $\hat{p}_t(x)$ is an approximation of the distribution $\pi_t(\cdot \,|\, x)$ at time t. Because of this, we can view the coefficients $\alpha(x)$ as a vector of probabilities which weight the nonlinear transformations of the control inputs. Thus, we place additional constraints on the coefficients $\alpha(x)$, and form the approximate optimization problem, constraining the values of $\alpha(x)$ such that they are non-negative and sum to one:

$$\min_{\alpha(x) \in \mathbb{R}^P} \quad \boldsymbol{c}^\top W \Psi k(x,\cdot) \Upsilon^\top \alpha(x) \tag{19a}$$

$$\text{s.t.} \quad \sum_{j=1}^{P} \alpha_j(x) = 1 \tag{19b}$$

$$0 \preceq \alpha(x) \tag{19c}$$

Since $\alpha(x)$ is an indirect weighting on the inputs that depends on the state x, we interpret $\alpha(x)$ as a probability weighting of the control inputs $\tilde{u}_j$. Note that (19) is a linear program in standard form [23], and that we can solve (19) via the Lagrangian dual. Let $\nu \in \mathbb{R}$ be a dual variable, and for simplicity, let $C(x) = \boldsymbol{c}^\top W \Psi k(x,\cdot) \Upsilon^\top$. The dual problem is given by:

$$\max \quad -\nu \tag{20a}$$

$$\text{s.t.} \quad -\mathbf{1}\nu \preceq C(x)^\top \tag{20b}$$

where $\mathbf{1}$ is a vector of all ones. From [23, §4], (20) has an optimal solution, given by $\min_i \{C_i(x)^\top\}$, which means the optimal solution $\alpha(x)^*$ to (19) is a vector of all zeros, except $\alpha_i(x)^* = 1$. In other words, we choose the control input that corresponds to the minimal value of $C(x)$.

A. Application to Approximate Dynamic Programming

Many optimal control problems can be formulated as dynamic programs. Consider the following problem with an additive cost, in which we seek a policy π that minimizes the following optimization problem [16]:

$$\min_{\pi} \quad J_N(\pi) = \mathbb{E}_\pi \left[g_N(x_N) + \sum_{t=0}^{N-1} g_t(x_t, u_t) \right] \tag{21}$$

where $N \in \mathbb{N}_+$ is the time horizon, π is the control policy, g_N is the terminal cost for ending in state x_N, g_t is the cost at time t of taking action $u_t \sim \pi_t(\cdot \,|\, x_t)$ while in state x_t, and the expectation is uniquely determined by the initial distribution $\mathbb{P}_0$ and the Markov policy π (see [16, Definition 8.3] for more details). The problem in (21) can be rewritten via the Chapman-Kolmogorov identity and the Markov property as a sequence of sub-problems, where the problem is solved backward in time via *backward recursion* [16]. We define the *value functions* $V_t : \mathcal{X} \to \mathbb{R}$ for all $t = 0, 1, \ldots, N-1$ as:

$$V_t(x) = \min_{\pi_t} \int_{\mathcal{U}} \int_{\mathcal{X}} g(x,v) + V_{t+1}(y) Q(\mathrm{d}y \,|\, x, v)\pi_t(\mathrm{d}v \,|\, x) \tag{22}$$

initialized with $V_N(x_N) = g_N(x_N)$. Then the solution to (21) is equivalent to solving a sequence of sub-problems given by (22), and iteratively substituting the solutions into the subsequent value function.

We can apply (19) in this context to solve for the optimal control policy when the dynamics and stochasticity are not known, but a sample $\mathcal{S}$ is available. In this case, we solve (22) at each time step t using (19). By approximating and recursively substituting the solution to (22) into the subsequent value function, we obtain an approximately optimal control policy $\pi^* \approx \arg\min_\pi J_N(\pi)$ which approximately minimizes the cost in (21).

This means we can compute the approximately optimal policy for a problem without exploiting knowledge of the system dynamics or the structure of the disturbance. By solving for the approximately optimal policy using (19), we avoid intractable integrals in the stochastic optimal control problem and can compute the policy as a linear operation in

a Hilbert space of functions. Additionally, this approach is largely agnostic to the dimensionality of the system, since the system dimensionality only directly affects the computation of the kernel function. For example, the Gaussian kernel scales linearly as the system dimensionality is increased. However, higher-dimensional systems typically require a larger sample size in order to fully characterize the dynamics of the system, which can be computationally prohibitive if the sample size is large.

V. Numerical Results

We demonstrate our approach on a 2-D discrete-time stochastic integrator system for the purpose of verification, and on a target tracking problem with nonholonomic vehicle dynamics to demonstrate the utility of the approach. For all problems, we used a Gaussian kernel $k(x, x') = \exp(-\|x - x'\|_2^2/2\sigma^2)$. Following [1], we chose the regularization parameter to be $\lambda = 1/M^2$, where M is the sample size, as the default parameter for our calculations. In practice, the parameters σ and λ are chosen via cross-validation, where σ is selected according to the relative "spacing" of the observations, and λ is a "smoothness" parameter chosen such that $\lambda \to 0$ as $M \to \infty$. A more detailed discussion of parameter selection is outside the scope of the current work (see [1, 24] for more information). Numerical experiments were performed in Matlab on an AWS cloud computing instance, and computation times were obtained using Matlab's Performance Testing Framework.

Code to reproduce the analysis and all figures is provided at: `github.com/unm-hscl/ajthor-CDC2021`.

A. Double Integrator System

We consider the problem of regulation for a system whose dynamics are governed by a stochastic 2-D discrete time stochastic integrator system, without any knowledge of the dynamics or the stochastic processes. That is, we seek a distribution π which minimizes the following optimization problem:

$$\min_{\pi} \quad \int_{\mathcal{U}} \int_{\mathcal{X}} c(y) Q(\mathrm{d}y \,|\, x, v) \pi(\mathrm{d}v \,|\, x) \tag{23}$$

where Q is a representation of the *unknown* system dynamics as a stochastic kernel. For the purpose of comparison, we chose the cost function $c : \mathcal{X} \to \mathbb{R}$ to be the norm function:

$$c(x) = \|x\|_2 \tag{24}$$

which serves to drive the system to the origin. The dynamics for a 2-D discrete time stochastic integrator system with sampling time T_s are given by:

$$x_{t+1} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} x_t + \begin{bmatrix} T_s^2/2 \\ T_s \end{bmatrix} u_t + w_t \tag{25}$$

where $x_t \in \mathcal{X}$ is the state, $u_t \in \mathcal{U}$ is the control input, which we specify to lie within the bounds $u_t \in [-1, 1]$, and w is a stochastic process, comprised of the random variables w_t on the measurable space $(\mathbb{R}^p, \mathscr{B}(\mathbb{R}^p))$. We consider three distributions for the disturbance: 1) A Gaussian distribution $w_t \sim \mathcal{N}(0, \Sigma)$, $\Sigma = 0.01I$; 2) A beta distribution $w_t \sim 0.1\mathrm{Beta}(\alpha, \beta)$, with a probability density function (PDF) given by:

$$f(x \,|\, \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha - 1}(1 - x)^{\beta - 1} \tag{26}$$

where Γ is the Gamma function and shape parameters $\alpha = 2$, $\beta = 0.5$; and 3) An exponential distribution $w_t \sim 0.01\mathrm{Exp}(\alpha)$, with $\alpha = 3$ and PDF $f(x \,|\, \alpha) = \alpha \exp(-\alpha x)$.

We consider a sample $\mathcal{S} = \{(x_i, u_i, x_i')\}_{i=1}^M$ of observations of size $M = 1600$ taken i.i.d. from Q, a representation of (25) as a Markov control process. The states $x_i \in \mathcal{X}$ were selected uniformly in the range $x_i \in [-1, 1] \times [-1, 1]$, the control inputs $u_i \in \mathcal{U}$ were chosen in the range $u_i \in [-1.1, 1.1]$, and the resulting states were generated according

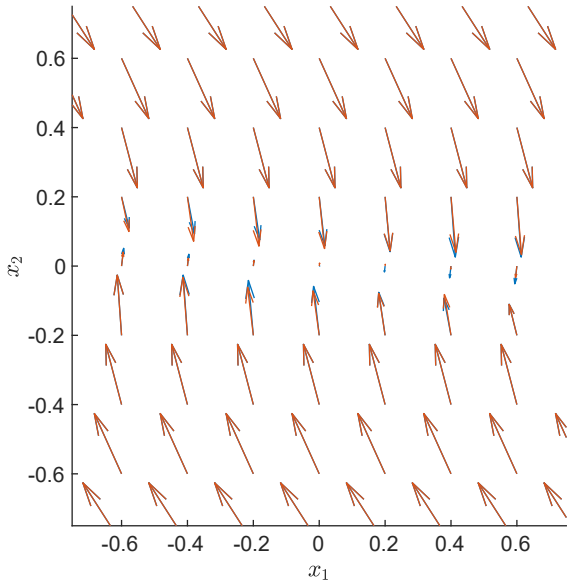

Fig. 1. Vector field showing the optimal closed-loop dynamics of a 2-D discrete-time integrator system under an optimal control strategy computed via CVX (blue). The vector field of the closed-loop dynamics of a stochastic integrator system with a Gaussian disturbance computed using our proposed algorithm (orange).

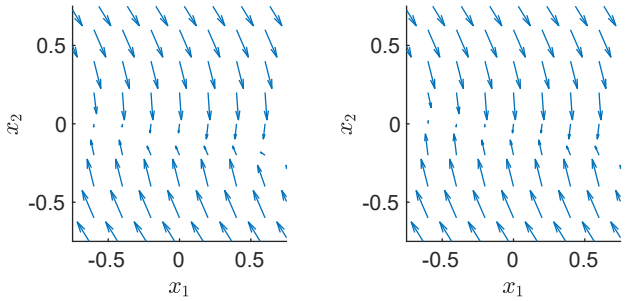

Fig. 2. (Left) Vector field of the approximately optimal closed-loop dynamics of a 2-D stochastic integrator system with a Beta disturbance, where the approximately optimal policy is computed using our proposed approach. (Right) Vector field of the approximately optimal closed-loop system with an exponential disturbance.

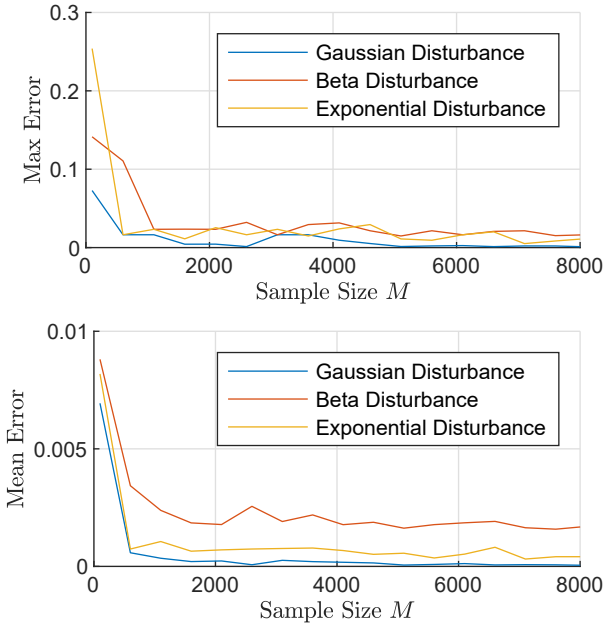

Fig. 3. (Top) Maximum error of the control inputs computed via our proposed method vs. the optimal control inputs computed via CVX. (Bottom) Mean error of the control inputs. The mean error is less than 0.005 when $M > 1000$.

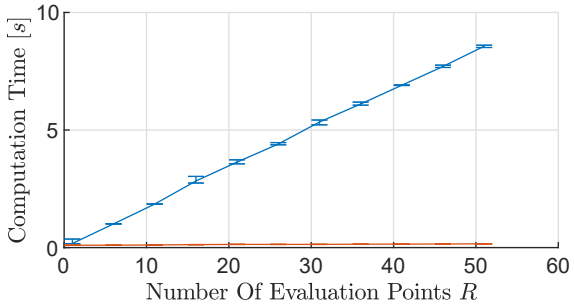

Fig. 4. Mean computation time of the optimal control solution, computed using CVX [25] (blue) vs. the mean computation time of the kernel based algorithm (orange) as a function of the number of evaluation points R. The sample size used to construct the estimate $\hat{m}$ is $M = 1600$, and the number of admissible control inputs is $P = 100$.

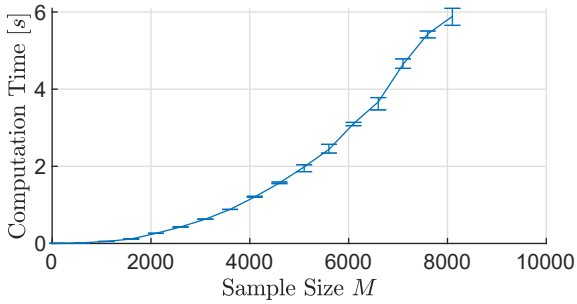

Fig. 5. Mean computation time of the kernel based algorithm (blue) as a function of the sample size M used to construct the approximation. The number of evaluation points is $R = 10$, and the number of admissible control inputs is $P = 100$. The computation time increases exponentially as the sample size increases.

to $x_i' \sim Q(\cdot \,|\, x_i, u_i)$. We then presumed no knowledge of the system dynamics or the structure of the disturbance for the purpose of computing the approximately optimal control inputs using our proposed method.

Using $\mathcal{S}$, we then computed an estimate $\hat{m}$ according to (11), which can be viewed as an empirical estimate of the system dynamics. We used a bandwidth parameter $\sigma = 1$ for our calculations, which was determined by cross-validation. We then chose a collection of admissible control inputs $\{\tilde{u}_\ell\}_{\ell=1}^P$, $P = 100$, in the range $\tilde{u}_\ell \in [-1, 1]$ to compute the estimator $\hat{p}$ in (16). We then selected $R = 25$ *evaluation points* $\{x_j\}_{j=1}^R$, chosen uniformly in the region $[-1, 1] \times [-1, 1]$ from which to compute the approximately optimal control inputs.

In order to demonstrate the effectiveness of the method, we computed the optimal control inputs using CVX [25] from the evaluation points $\{x_j\}_{j=1}^R$ using the *deterministic* dynamics. Once we computed the optimal inputs, we propagate the dynamics forward in time using the optimal inputs to obtain the state at the next time instant. We then plotted the vector field of the closed-loop dynamics under the optimal control input in Figure 1 (blue). We then computed the approximately optimal control inputs using (19) using the sample $\mathcal{S}$ taken from the *stochastic* dynamics to minimize the cost c at each point x_j over a single time step. Using the computed inputs, we then plotted the vector field of the closed-loop dynamics in Figure 1 (orange) to compare against the optimal control inputs computed via CVX.

We can see in Figure 1 that the control inputs selected by the algorithm are close to the optimal control inputs obtained by CVX, especially further away from the origin, where the computed control inputs coincide almost exactly with the optimal control inputs. Closer to the origin, we see that the algorithm deviates slightly from the optimal control inputs, which we anticipate is due largely to the randomness of the state observations.

We then generated a new sample $\mathcal{S}$ of the system in (25) affected by a disturbance with a beta distribution, and then from (25) affected by a disturbance with an exponential distribution and computed the optimal control inputs using our proposed method. The vector fields of the closed-loop dynamics for these cases are shown in Figure 2. We can see that the algorithm computes an approximately optimal controller, despite the state observations being affected by a non-Gaussian disturbance.

Note that the quality of the approximation obtained from our method depends on the sample size M. As the number of observations in the sample increase, the approximately optimal control inputs converge to the actual optimal control inputs. In order to demonstrate this, we computed the maximum and mean squared Euclidean error between the control inputs computed via our method and the optimal control inputs computed via CVX for varying sample sizes $M \in [100, 8000]$ in order to characterize the performance of the algorithm. The results are shown in Figure 3. We can see that the error of the approximately optimal control

inputs decreases quickly as the sample size increases, and that the mean error is approximately less than 0.005 when the sample size is greater than $M > 2000$. However, we can also see that the quality of the approximation does not improve significantly as the sample size increases, which is due to the asymptotic convergence of the estimate $\hat{m}$ to the true embedding m [1]. This presents a tradeoff between computation time and numerical accuracy, especially since the complexity scales exponentially as the sample size increases.

The computation times for both approaches are shown in Figure 4 as a function of the evaluation points up to $R = 51$. We can see from Figure 4 that the computation times for the CVX optimization method increases roughly linearly as the number of evaluation points increases, since the optimization problem needs to solve for each point independently. The computation time for our proposed method is also roughly linear in the number of evaluation points, but is dominated primarily by the computation time required for the matrix inversion, which increases exponentially with the sample size M, and is generally $\mathcal{O}(M^3)$ [1].

This is demonstrated empirically in Figure 5, where we compute the mean computation time as a function of the sample size M. We can see that as the sample size increases, the computation time increases exponentially. However, as mentioned earlier, the computation time of the kernel based approach can also be improved using existing speedup techniques [12, 13].

This also illustrates the computational advantage of our proposed method, since the quality of the approximation obtained via kernel methods has a mean error of roughly 0.005 with a sample size of $M = 1600$, but is able to compute the optimal control inputs an order of magnitude faster than the optimal solution via CVX for multiple evaluation points.

B. Nonholonomic Vehicle

We consider the problem of target tracking for a system with nonholonomic vehicle dynamics as defined in [26], modified such that it has a minimum forward velocity. The dynamics with sampling time T_s are given by:

$$\begin{aligned} \dot{x}_1 &= (u_1 + V_{\min}) \sin(x_3) + w \\ \dot{x}_2 &= (u_1 + V_{\min}) \cos(x_3) + w \\ \dot{x}_3 &= u_2 + w \end{aligned} \tag{27}$$

where $V_{\min} = 0.1$ is the minimum, constant forward velocity, $[x_1, x_2, x_3] \in \mathcal{X} \subseteq \mathbb{R}^3$ are the states, $[u_1, u_2]^\top \in \mathcal{U} \in \mathbb{R}^2$ are the control inputs, and $w \sim \mathcal{N}(0, \Sigma)$ is a random variable on the measurable space $(\mathbb{R}^p, \mathscr{B}(\mathbb{R}^p))$, where $\Sigma = 0.1I$. We define a target trajectory, moving from $x_{\text{init}} = [-1, -1, \pi/4]^\top$ to $x_{\text{final}} = [1, 1, \pi/4]^\top$ (shown in black in Fig. 6 and Fig. 7), and define the cost function such that the goal is to minimize the squared Euclidean distance from the system's position to the target trajectory's position at each time step.

We consider a sample $\mathcal{S} = \{(x_i, u_i, x_i')\}_{i=1}^M$ of size $M = 1600$ taken i.i.d. from Q, a representation of (27) as a Markov control process. The states x_i were drawn uniformly in the range $[-1.1, 1.1] \times [-1.1, 1.1] \times [-6, 6]$, the control inputs u_i were drawn uniformly in the range $[-0.1, 1.2] \times [-10.1, 10.1]$, and then x_i' drawn from $Q(\cdot \mid x_i, u_i)$.

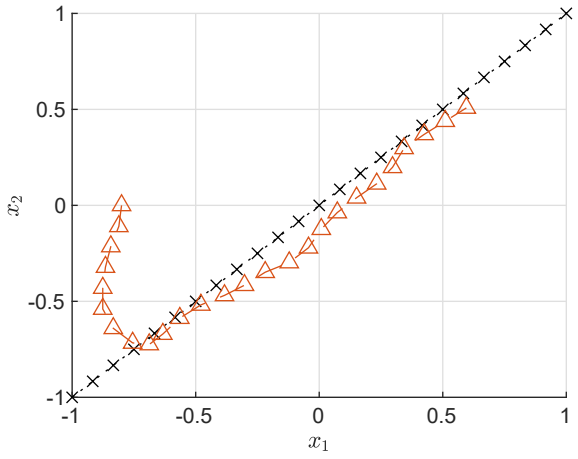

Fig. 6. Trajectory computed using our proposed method (orange) which tracks the target trajectory (black). The control actions are computed forward in time, with the system selecting the approximately optimal control action at each time step.

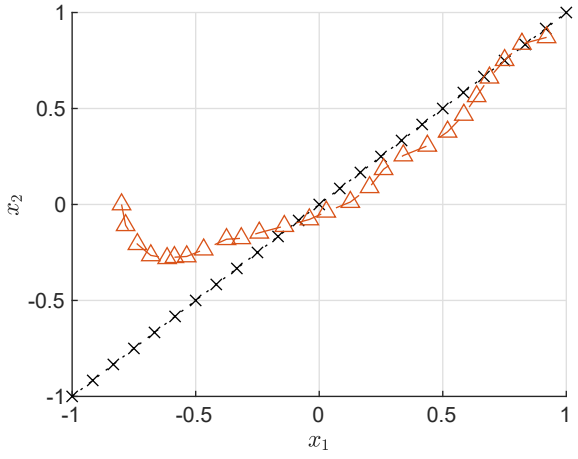

Fig. 7. Trajectory computed using our proposed method (orange) which tracks the target trajectory (black). The control actions are computed backward in time using dynamic programming. Note that the trajectory more closely follows the target using dynamic programming.

We then computed an estimate $\hat{m}$ according to (11) and used a bandwidth parameter of $\sigma = 3$ for the kernel function, determined by cross-validation. We then selected a collection $\{\tilde{u}_\ell\}_{\ell=1}^P$ of $P = 231$ admissible control inputs within the range $[0, 1] \times [-10, 10]$ to construct $\hat{p}$ as in (16). We choose an initial condition $x_0 = [-0.8, 0, \pi]^\top$, and evolve the system forward in time via (27) over a time horizon $N = 20$, computing an approximately optimal control action at each time step using our proposed method. The resulting trajectory is plotted in Figure 6 (orange), and the computation time over the time horizon was approximately 0.538 seconds. As expected, we can see that the control actions selected from our proposed method drive the system to closely follow the target trajectory.

We then computed the optimal controller via dynamic programming in order to compare against the forward in time approach. Unlike the previous approach, in which the control actions are selected in a greedy fashion, the dynamic programming approach computes the optimal control actions backward in time by iteratively optimizing a sequence of value functions (22), and then selecting the control actions at each time step which have the highest *value*. We use the same tracking trajectory as before, as well as the same initial condition in order to compare the performance of the two approaches. The resulting trajectory is shown in Figure 7 (orange). The computation time for the dynamic programming solution was approximately 6.448 seconds.

As expected, we can see that the trajectories obtained from the two approaches both follow the target trajectory, but the dynamic programming solution follows the trajectory better over the entire time horizon. This is because the value functions take into account the future actions of the system in order to minimize the total cost. This shows that our algorithm is able to select the approximately optimal control actions at each time step for a nonlinear system either forward in time or backward in time via dynamic programming, using only sample information taken from observations of the system evolution.

VI. Conclusions & Future Work

In this paper, we have presented a novel method for computing the optimal policy for discrete-time dynamic programming problems using observations taken from a stochastic system under an arbitrary disturbance. Our method is model-free and largely agnostic to the cost function used. We have demonstrated our proposed method on a discrete time stochastic double integrator system and on a nonholonomic vehicle target tracking problem. We plan to explore further theoretical extensions of this method to other classes of stochastic control problems, and to constrained optimal control problems.

References

[1] L. Song, J. Huang, A. Smola, and K. Fukumizu, "Hilbert space embeddings of conditional distributions with applications to dynamical systems," in *Proc. Int. Conf. on Mach. Learn.*, 2009, p. 961–968.

[2] S. Grünewälder, G. Lever, L. Baldassarre, M. Pontil, and A. Gretton, "Modelling transition dynamics in MDPs with RKHS embeddings," in *Proc. Int. Conf. on Mach. Learn.*, 2012, p. 1603–1610.

[3] Y. Nishiyama, A. Boularias, A. Gretton, and K. Fukumizu, "Hilbert space embeddings of POMDPs," in *Proc. Conf. on Uncertainty in Artif. Intell.*, 2012, p. 644–653.

[4] L. Song, B. Boots, S. Siddiqi, G. Gordon, and A. Smola, "Hilbert space embeddings of hidden Markov models," in *Proc. Int. Conf. on Mach. Learn.*, 2010, p. 991–998.

[5] J.-J. Zhu, W. Jitkrittum, M. Diehl, and B. Schölkopf, "Kernel distributionally robust optimization," *ArXiv Preprint ArXiv:2006.06981*, 2020.

[6] J.-J. Zhu, B. Schölkopf, and M. Diehl, "A kernel mean embedding approach to reducing conservativeness in stochastic programming and control," in *Learn. for Dynamics and Ctrl.*, 2020, pp. 915–923.

[7] D. Luenberger, *Optimization by Vector Space Methods*. John Wiley & Sons, 1997.

[8] D. Ormoneit and S. Sen, "Kernel-based reinforcement learning," *Mach. Learn.*, vol. 49, no. 2–3, p. 161–178, 2002.

[9] J. Bagnell and J. Schneider, "Policy search in kernel Hilbert space," 2003.

[10] G. Lever and R. Stafford, "Modelling Policies in MDPs in Reproducing Kernel Hilbert Space," in *Proc. Int. Conf. on Artif. Intell. and Statist.*, vol. 38, 2015, pp. 590–598.

[11] A. M. S. Barreto, D. Precup, and J. Pineau, "Practical kernel-based reinforcement learning," *J. Mach. Learn. Res.*, vol. 17, no. 1, p. 2372–2441, 2016.

[12] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Int. Conf. on Neural Inf. Process. Syst.*, 2007, p. 1177–1184.

[13] Q. Le, T. Sarlós, and A. Smola, "Fastfood: Approximating kernel expansions in loglinear time," in *Proc. Int. Conf. on Mach. Learn. - Volume 28*, 2013, p. III–244–III–252.

[14] E. Çinlar, *Probability and Stochastics*. Springer, 2011.

[15] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2005.

[16] D. Bertsekas and S. Shreve, *Stochastic optimal control: the discrete time case*. Elsevier, 1978.

[17] N. Aronszajn, "Theory of reproducing kernels," *Trans. of the Amer. Math. Soc.*, vol. 68, no. 3, pp. 337–404, 1950.

[18] I. Steinwart and A. Christmann, *Support vector machines*. Springer, 2008.

[19] S. Grünewälder, G. Lever, L. Baldassarre, S. Patterson, A. Gretton, and M. Pontil, "Conditional mean embeddings as regressors," in *Proc. Int. Conf. on Mach. Learn.*, 2012, p. 1803–1810.

[20] C. Micchelli and M. Pontil, "On learning vector-valued functions," *Neural Comput.*, vol. 17, no. 1, p. 177–204, 2005.

[21] C. Micchelli, Y. Xu, and H. Zhang, "Universal kernels," *J. Mach. Learn. Res.*, vol. 7, p. 2651–2667, 2006.

[22] L. Song, A. Gretton, and C. Guestrin, "Nonparametric tree graphical models," in *Proc. Int. Conf. on Artif. Intell. and Statist.*, vol. 9, 2010, pp. 765–772.

[23] S. Boyd, S. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[24] A. Caponnetto and E. De Vito, "Optimal rates for the regularized least-squares algorithm," *Foundations of Computational Mathematics*, vol. 7, no. 3, pp. 331–368, 2007.

[25] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," 2014.

[26] H.-T. Chiang, N. Malone, K. Lesser, M. Oishi, and L. Tapia, "Path-guided artificial potential fields with stochastic reachable sets for motion planning in highly dynamic environments," in *IEEE Int. Conf. on Robot. and Automation*, 2015, pp. 2347–2354.