Stochastic Reachability for Systems up to a Million Dimensions

Adam J. Thorpe
Electrical & Computer Eng.
University of New Mexico
Albuquerque, New Mexico
ajthor@unm.edu

Vignesh Sivaramakrishnan
Electrical & Computer Eng.
University of New Mexico
Albuquerque, New Mexico
vigsiv@unm.edu

Meeko M. K. Oishi
Electrical & Computer Eng.
University of New Mexico
Albuquerque, New Mexico
oishi@unm.edu

ABSTRACT

We present a solution to the first-hitting time stochastic reachability problem for extremely high-dimensional stochastic dynamical systems. Our approach takes advantage of a non-parametric learning technique known as conditional distribution embeddings to model the stochastic kernel using a data-driven approach. By embedding the dynamics and uncertainty within a reproducing kernel Hilbert space, it becomes possible to compute the safety probabilities for stochastic reachability problems as simple matrix operations and inner products. We employ a convergent approximation technique, random Fourier features, in order to accommodate the large sample sets needed for high-dimensional systems. This technique avoids the curse of dimensionality, and enables the computation of safety probabilities for high-dimensional systems without prior knowledge of the structure of the dynamics or uncertainty. We validate this approach on a double integrator system, and demonstrate its capabilities on a million-dimensional, nonlinear, non-Gaussian, repeated planar quadrotor system.

CCS CONCEPTS

• Computing methodologies → Computational control theory; Kernel methods; • Theory of computation → Stochastic control and optimization.

KEYWORDS

Stochastic Reachability, Machine Learning, Stochastic Optimal Control

ACM Reference Format:
Adam J. Thorpe, Vignesh Sivaramakrishnan, and Meeko M. K. Oishi. 2020. Stochastic Reachability for Systems up to a Million Dimensions. In Proceedings of Hybrid Systems: Computation and Control (HSCC ’20). ACM, New York, NY, USA, 12 pages.

1 INTRODUCTION

Stochastic reachability is an established verification technique which is used to compute the likelihood that a system will reach a desired state without violating a predefined set of safety constraints. The solutions to stochastic reachability problems are broadly framed in terms of a dynamic program [1, 31], which scales poorly with the dimensionality of the system. Methods using approximate dynamic programming [12], particle filtering [14, 17], and abstractions [28] have been posed, but are limited to systems of moderate dimensionality. Optimization-based solutions have garnered modest computational tractability via chance constraints [14, 38], sampling methods [22, 33, 34], and convex optimization with Fourier transforms [36, 37], but are limited to linear dynamical systems and Gaussian or log-concave disturbances.

Recent work in reachability for non-stochastic, linear dynamical systems has accommodated systems with up to a billion dimensions [4–6], an unprecedented size. However, comparably scalable solutions for stochastic systems, even with considerable structure in the dynamics and the uncertainty, remain elusive.

We propose a model-free method for stochastic reachability analysis of extremely high-dimensional systems using a class of machine learning techniques known as kernel methods. Kernel methods [23] employ a data-driven approach to perform functional analysis in a high-dimensional space. Kernel methods are advantageous because they are agnostic to structure in the dynamics or the uncertainty, meaning that they are amenable to generic Markov control processes,
and they avoid numerical integration by approximating integrals via inner products. These techniques scale exponentially with the number of samples, meaning that they can suffer from computational complexity and memory storage requirements. This can be prohibitive for high-dimensional systems, which may require a large number of samples in order to effectively capture the dynamics of the system. The utility of kernel methods for the terminal-hitting time problem has been demonstrated for systems of up to 10,000 dimensions [32], but the jump to a million presents significant computational challenges.

Our approach exploits a nonparametric learning technique known as conditional distribution embeddings [25–27] which has traditionally been applied to POMDPs [11, 15] and more recently to stochastic optimal control [7]. Distribution embeddings extend the data-driven regression techniques afforded by kernel methods to statistical measures and probability distributions. These methods are closely related to Gaussian processes [39], but do not impose the same assumptions about the noise. To facilitate stochastic reachability calculations for extremely high-dimensional systems, we couple kernel distribution embeddings with a technique known as random Fourier features [20], that uses an empirical Fourier approximation to deal with large sample sets. The main contribution of this paper is an application of random Fourier features to compute an efficient model-free approximation of the safety probabilities for high-dimensional systems.

The paper is outlined as follows. Section 2 formulates the problem. Section 3 outlines the use of conditional distribution embeddings to perform safety verification for the first-hitting time problem. Section 4 describes the application of random Fourier features in order to compute the safety probabilities for high-dimensional systems. In section 5, we demonstrate our approach on two examples: a stochastic chain of integrators in order to demonstrate the feasibility of the technique, and a repeated planar quadrotor example to demonstrate the approach for a high-dimensional system up to one million dimensions.

2 PROBLEM FORMULATION

The following notation is used throughout the paper. For any nonempty space $\Omega$, the indicator $1_{\mathcal{A}} : \Omega \rightarrow \{0, 1\}$ of $\mathcal{A} \subset \Omega$ is a function defined such that $1_{\mathcal{A}}(\omega) = 1$ if $\omega \in \mathcal{A}$, and $1_{\mathcal{A}}(\omega) = 0$ if $\omega \notin \mathcal{A}$.

Let $(\Omega, \mathcal{F}(\Omega), \Pr)$ define a probability space, where $\mathcal{F}(\Omega)$ denotes the $\sigma$-algebra relative to $\Omega$, and $\Pr$ is the assigned probability measure. When $\Omega \equiv \mathbb{R}$, let $\mathcal{B}(\Omega)$ denote the Borel $\sigma$-algebra associated with $\Omega$. Given $i \in \mathbb{N}$ random variables $x_i$, which are measurable functions on $(\Omega, \mathcal{F}(\Omega), \Pr)$, let $x = [x_1, \ldots, x_n]^\top$ be a random vector defined on the induced probability space $(\Omega^n, \mathcal{F}(\Omega^n), \Pr_x)$, where $\Pr_x$ is the induced probability measure. A stochastic process is defined as a sequence of random vectors $\{x_k\}_{k=0}^N$, $N \in \mathbb{N}$. For a real, measurable function $x$ on $(\Omega, \mathcal{F}(\Omega), \Pr)$, the Lebesgue integral $\int_\Omega x \Pr$ is denoted by the expectation operator $E_{\Pr_x}[x]$.

2.1 System Model

We consider a Markov control process $\mathcal{H}$, which is defined in [31] as a 3-tuple:

$$\mathcal{H} = (X, \mathcal{U}, Q)$$

where $X \subseteq \mathbb{R}^n$ is a compact Borel space representing the state space; $\mathcal{U} \subseteq \mathbb{R}^m$ is a compact Borel space representing the control space; and $Q: \mathcal{B}(X) \times X \times \mathcal{U} \rightarrow [0, 1]$ is a stochastic kernel, which is a Borel-measurable function that maps a probability measure $Q(\cdot | x, u)$ to each $x \in X$ and $u \in \mathcal{U}$ in $(X, \mathcal{B}(X))$.

The system evolves over a finite time horizon $k \in [0, N]$, where the inputs are chosen from a Markov control policy $\pi = \{\pi_0, \pi_1, \ldots\}$ [9, 19], which is a sequence of universally measurable maps $\pi_k : X \rightarrow \mathcal{U}$. The set of all Markov policies is denoted as $\mathcal{M}$.

We consider a set $\mathcal{S}$ of $m$ samples $\{(\tilde{x}_i, \tilde{u}_i, \tilde{y}_i)\}_{i=1}^m$ such that $\tilde{y}_i$ is drawn i.i.d. from the stochastic kernel $Q$, and $\tilde{u}_i$ is computed from a deterministic, fixed Markov policy $\pi$.

$$\tilde{y}_i \sim Q(\cdot | \tilde{x}_i, \tilde{u}_i)$$

$$\tilde{u}_i = \pi(\tilde{x}_i)$$

We denote sample vectors with a bar to differentiate them from time-indexed vectors.

2.2 Reproducing kernel Hilbert Spaces

Given a positive definite [30, Definition 4.15] kernel over the domain $(X, \mathcal{B}(X))$, $K_X : X \times X \rightarrow \mathbb{R}$, let $\mathcal{H}_X$ denote the unique reproducing kernel Hilbert space (RKHS) for $X$. A RKHS $\mathcal{H}_X$ is a Hilbert space of real-valued functions on $X$ with inner product $(\cdot, \cdot)_{\mathcal{H}_X}$, and two important properties [3]: 1) For any $x, x' \in X$, $K_X(x, \cdot) : x' \rightarrow \mathbb{R}$ is an element of $\mathcal{H}_X$; and 2) An element $K_X(x, x')$ of $\mathcal{H}_X$ satisfies the reproducing property such that $\forall f \in \mathcal{H}_X$,

$$f(x) = (K_X(x, \cdot), f(\cdot))_{\mathcal{H}_X}$$

From (4), it follows that $K_X(x, x') = (K_X(x, \cdot), K_X(x', \cdot))_{\mathcal{H}_X}$, meaning we can efficiently compute the inner product using the kernel function.

By choosing $K_X$, we effectively choose a basis to represent the functions in $\mathcal{H}_X$. Using the reproducing property, we can then write the function $f$ as $f(x) = w^\top \phi(x)$, a weighted sum of basis functions for some possibly infinite-dimensional weight vector $w$. We wish to solve for the particular $w$ which,
We consider the following problems:

**Problem 1.** Use samples $S$ to compute an efficient approximation of the kernel using random Fourier features that enables an efficient computation of the safety probabilities in (8) for high-dimensional systems.

**Problem 2.** Without direct knowledge of $Q$, use samples $S$ to compute a kernel-based approximation of (8) that converges in probability.

In the following sections, we will utilize conditional distribution embeddings and random Fourier features to compute the safety probabilities for stochastic reachability using a data-driven approach. Because conditional distribution embeddings require that we generate and maintain a matrix that scales with the number of samples, large data sets can have intractable storage requirements. For instance, storing the matrix for a million sample points would require 7450.6 GB. Instead, we show that we can utilize random Fourier features as outlined in [20] to approximate the kernel and enable computation of the safety probabilities in (6) for high-dimensional systems by providing a reduced-complexity model.

In the sections that follow, we first show that we can compute the safety probabilities for the first hitting time problem using a convergent approximation method via conditional distribution embeddings. Then, we show that we can compute an approximation of the kernel using random Fourier features in order to enable first-hitting time analysis for high-dimensional systems.

### 2.3 First-Hitting Time Problem

Let $\mathcal{K}, \mathcal{T} \in B(\mathcal{X})$ denote the safe set and target set, respectively. We define the first-hitting time safety probability $r_{x_0}(\mathcal{K}, \mathcal{T})$ [31] as the probability that a system $\mathcal{H}$ controlled by a Markov policy $\pi \in M$ will reach $\mathcal{T}$ at some time $j \in [0, N]$ while avoiding $\mathcal{X} \setminus \mathcal{K}$ for all $k \in [0, j - 1]$.

$$
 r_{x_0}(\mathcal{K}, \mathcal{T}) \equiv \Pr_{x_0}(\exists j \in [0, N] : x_j \in \mathcal{T} \land \forall i \in [0, j - 1], x_i \in \mathcal{K} \setminus \mathcal{T})
$$

Unlike the terminal-hitting time problem [31], which is concerned with staying safe over the entire time horizon and reaching $\mathcal{T}$ at time $N$, the first-hitting time problem is concerned with the probability that the system will reach $\mathcal{T}$ at some point during the time horizon. Let the value functions $V_k^\pi : \mathcal{X} \to [0, 1], k \in [0, N]$, be defined via backward recursion:

$$
 V_k^\pi(x) = 1_\mathcal{T}(x) \quad (7)
$$

$$
 V_k^\pi(x) = 1_\mathcal{T}(x) + 1_{\mathcal{K}(\mathcal{T})}(x)E_{y \sim Q(\cdot \mid x, u)}[V_{k+1}^\pi(y)] \quad (8)
$$

Then, $V_0^\pi(x) = r_{x_0}(\mathcal{K}, \mathcal{T})$ for every $x_0 \in \mathcal{X}$.

### 2.4 Problem Statement

We consider the following problems:
the reproducing kernel Hilbert space for \(X \times U\) with the positive definite kernel \(K_{X \times U}: (X, U) \times (X, U) \to \mathbb{R}\), such that \(K_{X \times U} = K_X \otimes K_U\), where \(\otimes\) denotes the tensor product. The goal is to find an element \(\tilde{\mu}(x, u) \in \mathcal{H}_X\), which is the conditional distribution embedding of the stochastic kernel \(Q\), such that \(V_f \in \mathcal{H}_X\),

\[
\langle \tilde{\mu}(x, u), f \rangle_{\mathcal{H}_X} = \mathbb{E}_{y \sim Q(|x, u)}[f(y)]
\]

We then compute an estimate \(\tilde{\mu}(x, u)\) of \(\mu(x, u)\) using samples \(S\), such that \(\forall f \in \mathcal{H}_X\),

\[
\langle \tilde{\mu}(x, u), f \rangle_{\mathcal{H}_X} \approx \mathbb{E}_{y \sim Q(|x, u)}[f(y)]
\]  

According to the Riesz representation theorem \([18]\), the estimate \(\tilde{\mu}(x, u)\) is the solution to a regularized least-squares problem \([15]\),

\[
\min_{\tilde{\mu}} \left\{ \frac{1}{M} \sum_{i=1}^{M} \|K_X(y_i, \cdot) - \tilde{\mu}(x_i, u_i)\|_{\mathcal{H}_X}^2 + \lambda \|\tilde{\mu}\|_{\mathcal{H}_X}^2 \right\}
\]  

(13)

where \(\mathcal{H}_Y\) is a vector-valued RKHS \([18]\). We detail the connection to \(\mathcal{H}_X\) in Appendix A. According to \([18, Theorem 4.1]\), the solution to (13) is unique and has the form

\[
\tilde{\mu}(x, u) = \eta \Psi^T (G + \lambda M I)^{-1} \Psi K_{X \times Q}((x, u), \cdot)
\]

(14)

where \(\lambda\) is a regularization parameter to avoid overfitting. \(\eta\) is a normalization constant, and \(\Phi, \Psi, G\) are defined as

\[
\Phi = [K_X(y_1, \cdot), \ldots, K_X(y_M, \cdot)]^T
\]

(15)

\[
\Psi = [K_{X \times Q}((x, u_1), \cdot), \ldots, K_{X \times Q}((x, u_M), \cdot)]^T
\]

(16)

\[
G = \Psi \Psi^T
\]

(17)

By the reproducing property of \(K_X\) in \(\mathcal{H}_X\), \(\forall f \in \mathcal{H}_X\), we can rewrite (12) as

\[
\langle \tilde{\mu}(x, u), f \rangle_{\mathcal{H}_X} = f^T \tilde{\beta}(x, u)
\]  

(18)

where \(f = [f(y_1), \ldots, f(y_M)]^T\), and \(\beta(x, u) \in \mathbb{R}^M\) is a vector of coefficient values,

\[
\tilde{\beta}(x, u) = \eta (G + \lambda M I)^{-1} \Psi K_{X \times Q}((x, u), \cdot)
\]

(19)

This means we can approximate the value function expectation \(\mathbb{E}_{y \sim Q(|x, u)}[V_{k+1}^\pi(y)]\) in (8) as an inner product with the conditional distribution embedding estimate.

**Remark 1.** The normalization constant \(\eta\) is called feature scaling \([23, Section 2.1]\), and ensures that the result of (18) is within the range \([0, 1]\).

### 3.1 First-Hitting Time Problem

We now show that we can use (14) to construct an approximation of the safety probability in (6). We rewrite the value function in (8) as

\[
V_k^\pi(x) = 1_T(x) + 1_{K \setminus T}(x)(\tilde{\mu}(x, u), V_{k+1}^\pi)_{\mathcal{H}_X}
\]

(20)

With the estimate \(\tilde{\mu}(x, u)\), we obtain the approximation

\[
V_k^\pi(x) \approx 1_T(x) + 1_{K \setminus T}(x)(\tilde{\mu}(x, u), V_{k+1}^\pi)_{\mathcal{H}_X}
\]

(21)

We define the approximate value functions \(\overline{V}_k^\pi: X \to [0, 1]\), \(k \in [0, N - 1]\), given by

\[
\overline{V}_k^\pi \triangleq 1_T(x) + 1_{K \setminus T}(x)(\tilde{\mu}(x, u), V_{k+1}^\pi)_{\mathcal{H}_X}
\]

(22)

### 3.2 Convergence

We now seek to characterize the convergence of the approximation. This follows the results presented in \([32]\), where convergence is shown for the terminal-hitting time problem. Here, we extend the results presented in \([32]\) to show convergence for the first-hitting time problem. As shown in \([11, Lemma 2.2]\), for any \(\epsilon > 0\), if the regularization parameter \(\lambda\) in (13) is chosen such that \(\lambda \to 0\) and \(\lambda M \to \infty\), and if \(X\) is bounded and \(K_X\) is strictly positive definite, then the distance pseudometric \(\|\tilde{\mu}(x, u) - \mu(x, u)\|_{\mathcal{H}_X}\) converges in probability. Thus, we can show that as the number of samples \(M\) increases, the value function estimate also converges in probability.

**Proposition 1 (Value Function Convergence).** For any \(\epsilon > 0\), if the regularization parameter \(\lambda\) in (13) is chosen such that \(\lambda \to 0\) and \(\lambda M \to \infty\), and if \(X\) is bounded and \(K_X\) is strictly positive definite, \(|\overline{V}_k^\pi(x) - V_k^\pi(x)|\) converges in probability.
Proof. The proof is by induction and follows [32, Proposition 2]. By subtracting (20) from (21), we define the absolute value function error \( \mathcal{E}_k(x) \) at time \( k \),

\[
\mathcal{E}_k(x) \triangleq |V_k^\tau \langle x \rangle - \overline{V}_k^\tau \langle x \rangle | \tag{24}
\]

\[
= |1_\tau \langle x \rangle + 1_\mathcal{K}(x)\langle \mu(x, \pi_{k}(x)), V_{k+1}^\tau \rangle|_\mathcal{K} - \nonumber
\]

\[
1_\tau \langle x \rangle - 1_\mathcal{K}(x)\langle \hat{\mu}(x, \pi_{k}(x)), V_{k+1}^\tau \rangle|_\mathcal{K} \tag{25}
\]

Using the parallelogram law and Cauchy-Schwarz, we can rewrite (25) to obtain

\[
\mathcal{E}_k(x) = 1_\mathcal{K}(x)|\langle \mu(x, \pi_{k}(x)) - \hat{\mu}(x, \pi_{k}(x)), V_{k+1}^\tau \rangle|_\mathcal{K} \tag{26}
\]

\[
\leq 1_\mathcal{K}(x)|V_{k+1}^\tau |_\mathcal{K} \| \mu(x, \pi_{k}(x)) - \hat{\mu}(x, \pi_{k}(x)) \|_\mathcal{K} \tag{27}
\]

Since \( \| \mu(x, \pi_{k}(x)) - \hat{\mu}(x, \pi_{k}(x)) \|_\mathcal{K} \) converges in probability according to [11, Lemma 2.2], \( V_k^\tau \langle x \rangle - \overline{V}_k^\tau \langle x \rangle \) also converges in probability with the probabilistic error bound \( \epsilon \). \( \square \)

Following the results from [32], we can show that as the value function estimate converges in probability, the approximation obtained using Algorithm 1 converges in probability to \( \pi^\omega_\mu(k, T) \). Because the error converges in probability to \( \epsilon \) for all \( k < N \), by approximating and recursively substituting \( \overline{V}_k^\tau \langle x \rangle \) in Algorithm 1, the error obtained by Algorithm 1 converges in probability to at most \( N \epsilon \) [32].

4 STOCHASTIC REACHABILITY FOR HIGH-DIMENSIONAL SYSTEMS

We now consider computing the safety probabilities for high dimensional systems. In general, the most significant computational bottleneck of kernel methods is computing and storing the matrix \( Q \) (17), which is at least \( O(M^2) \). This means that the complexity of Algorithm 1 is exponential in the number of samples. In the case of high-dimensional systems, a large number of samples may be needed to fully characterize the stochastic kernel \( Q \), which can make computing and storing \( Q \) prohibitive. However, significant progress has been made to reduce the computational complexity of kernel methods [13, 15, 20] to make them scalable for large sample sets. We focus on random Fourier features (RFF), which utilizes an empirical approximation of the Fourier integral to efficiently compute an approximation of the kernel. This reduces the computational complexity of Algorithm 1 by approximating the inner product in (12) using a low-dimensional Euclidean inner-product space.

4.1 Random Fourier Features

Instead of computing a high-dimensional approximation using the kernel trick [24], we consider a low-dimensional approximation of the kernel as the inner product of randomized feature maps \( \tilde{z} : \mathbb{R}^d \rightarrow \mathbb{R}^D \). Using this, we can approximate \( K_X \) and \( K_{X \times U} \) in order to efficiently estimate the kernel.

In order to do this, [20] shows that by exploiting Bochner’s theorem [21], we can approximate the inner product in (18) by computing the Fourier transform \( \Lambda(\omega) \) of the kernel function and approximating the Fourier integral in feature space using samples of the frequency variable.

Bochner’s Theorem. [21] A continuous kernel \( k(x, x') = \varphi(x - x') \) on \( \mathbb{R}^d \) is positive definite if and only if \( \varphi(x - x') \) is the Fourier transform of a non-negative Borel measure \( \Lambda \).

Let \( k(x, x') \) be a real-valued, positive definite, translation-invariant kernel, such that there exists a positive definite function \( \varphi : \mathbb{R}^d \rightarrow \mathbb{R} \) such that \( k(x, x') = \varphi(x - x') \). As shown in [20, 29], by Bochner’s theorem, \( \varphi \) is the Fourier transform of a Borel measure \( \Lambda(\omega) \).

\[
\varphi(x - x') = \int_{\mathbb{R}^d} \exp(i\omega^\top (x - x'))\Lambda(\omega) \tag{28}
\]

\[
= \int_{\mathbb{R}^d} \cos(\omega^\top (x - x'))\Lambda(\omega) \tag{29}
\]

where (29) follows from the real-valued property of \( \varphi \). Using the results from [20], we construct an estimate of the integral using a set \( \Omega \) of \( D \) samples \( \{\omega_i\}_{i=1}^D \), such that \( \omega_i \) is drawn i.i.d. from the Borel measure \( \Lambda \) according to \( \omega_i \sim \Lambda(\cdot) \).

\[
k(x, x') \approx \frac{1}{D} \sum_{i=1}^D \cos(\omega_i^\top (x - x')) \tag{30}
\]

By using the trigonometric identity \( \cos(\alpha - \beta) = \cos(\alpha) \cos(\beta) + \sin(\alpha) \sin(\beta) \), (29) can be written as

\[
\varphi(x - x') = \int_{\mathbb{R}^d} \left[ \cos(\omega^\top x) \cos(\omega^\top x') + \sin(\omega^\top x) \sin(\omega^\top x') \right] \Lambda(\omega) \tag{31}
\]

Then, for the randomized feature map \( z \),

\[
z(x) = \frac{1}{\sqrt{D}} \left[ \cos(\omega_1^\top x), \ldots, \cos(\omega_D^\top x), \sin(\omega_1^\top x), \ldots, \sin(\omega_D^\top x) \right]^\top \tag{32}
\]

we can write (30) as:

\[
k(x, x') \approx \langle z(x), z(x') \rangle \tag{33}
\]

By correlating \( K_X \) and \( K_{U} \) with \( k \) in (33), we obtain approximations of the kernels as inner products of randomized feature maps \( z \). Using the tensor product of kernels, since \( K_{X \times U} = K_X \otimes K_{U} \), an approximation of \( K_{X \times U} \) can be obtained as

\[
k_{X \times U}((u, x), (u', x')) = K_X(x, x')K_{U}(u, u') \tag{34}
\]

\[
\approx \langle z(x), z(x') \rangle \langle z(u), z(u') \rangle \tag{35}
\]

We define the tensor product of random feature maps according to the algebraic tensor product, such that

\[
\langle z(x), z(x') \rangle \langle z(u), z(u') \rangle = \langle z(x) \otimes z(u), z(x') \otimes z(u') \rangle \tag{36}
\]
We now seek to characterize the convergence of the approximation computed using Algorithm 2. The results in [20, Claim 1] show that the approximation in (33) converges in probability for some probabilistic error bound $\rho > 0$ as the number of frequency samples increases,

$$\Pr_{\Omega - \Lambda}\left\{ \sup_{(x,x') \in X} |(z(x), z(x')) - K(x', x')| \geq \rho \right\} \leq 2^k \left( \frac{\sigma_\Lambda \text{diam}(X)}{\rho} \right)^2 \exp\left( -\frac{D\rho^2}{4(d+2)} \right)$$

where $\sigma_\Lambda^2 = \mathbb{E}_\Lambda[\omega^T \omega]$ is the second moment of $\Lambda$. Further, [20, Claim 1] also shows that given any probabilistic error bound $\rho$, the supremum in (43) is bounded by $\rho$ with any constant probability if we choose $D$ such that:

$$D = O(d\rho^{-2}\log(\sigma_\Lambda \text{diam}(X)\rho^{-1}))$$

Because the kernel approximation in (33) converges in probability to $\rho$ as the number of frequency samples increases, the approximation computed using Algorithm 2 also converges in probability to $r^\pi_x(\mathcal{K}, \mathcal{T})$ as $D \to \infty$ for any probabilistic error bound $\rho$. This means that as the kernel approximation converges in probability, the value function approximation $|V^\pi_x(x) - V^\pi_x(x)|$ also converges in probability with the probabilistic error bound $\rho$ where $\epsilon$ is the probabilistic error bound outlined in Proposition 1.

$D$ vs. $\rho$. Because the probabilistic error bound $\rho$ appears in (44), we can choose $\rho$ based upon the desired quality of the approximation. This provides a tradeoff between numerical accuracy and computational efficiency by the choice of $D$. Lower values of $D$ will provide a faster approximation, while higher values will reduce the error of the value function approximation.

## 5 NUMERICAL RESULTS

We implemented Algorithm 1 on a stochastic chain of integrators for the purposes of validation. Then, we implemented Algorithm 2 on a million-dimensional repeated planar quadrotor example in order to demonstrate the method for high-dimensional systems. We first generated $M$ samples via simulation, and then presumed no knowledge of the dynamics or the structure of the uncertainty for the purposes of computing the safety probability $r^\pi_x(\mathcal{K}, \mathcal{T})$ in (6) using Algorithms 1 & 2. For all problems, we used a Gaussian kernel

$$K(x, x') = \exp(-\|x - x'\|^2 / 2\sigma^2)$$

with $\sigma = 0.1$, and chose $\lambda = 1$ as the default regularization parameter for the evaluation. The Borel measure $\Lambda$ that corresponds to the Fourier transform of the Gaussian kernel is a Gaussian distribution of the form

$$\Lambda(\omega) = \frac{1}{\sigma} \exp\left( -\frac{\omega^2}{2\sigma^2} \right)$$

All computations were done in Matlab on a 3.8GHz Intel Xeon CPU with 32 GB RAM. Computation times were
Table 1: Computation Time

| System               | Disturbance | Dim. \([n]\) | Number of Sample Points \([M]\) | Number of Evaluation Points | Number of Frequency Samples \([D]\) | Algorithm 1 | Algorithm 2 | Prog. |
|----------------------|-------------|--------------|-------------------------------|-----------------------------|--------------------------------|-------------|-------------|-------|
| Chain of Integrators | Gaussian    | 2            | 2,500                         | 10,000                      | 15,000                         | 2.30 s      | 22.94 s     | 65.78 s |
| Chain of Integrators | Beta        | 2            | 2,500                         | 10,000                      | 15,000                         | 2.30 s      | 22.75 s     | 69.01 s |
| Chain of Integrators | Exponential | 2            | 2,500                         | 10,000                      | 15,000                         | 2.32 s      | 22.81 s     | 65.35 s |
| Planar Quadrotor    | Gaussian    | 6            | 1,000                         | 10,000                      | 15,000                         | 0.62 s      | 15.24 s     | -      |
| Planar Quadrotor    | Beta        | 6            | 1,000                         | 10,000                      | 15,000                         | 0.61 s      | 15.25 s     | -      |
| Planar Quadrotor    | Gaussian    | 1,000,000    | 1,000                         | 1                           | 1000                           | 1.32 h      | 51.83 s     | -      |
| Planar Quadrotor    | Beta        | 1,000,000    | 1,000                         | 1                           | 500                            | 1.23 h      | 44.59 s     | -      |

Figure 2: (a) Dynamic-programming-based solution for a double integrator system with a Gaussian disturbance \(w_k \sim \mathcal{N}(0, \Sigma)\) over the horizon \(N = 5\). (b) First-hitting time safety probabilities for a double integrator system using Algorithm 1. (c) Absolute error between (a) and (b). (d) First-hitting time safety probabilities for a double integrator using Algorithm 2, where \(D = 15000\). (e) Absolute error between (a) and (d).

obtained using Matlab’s Performance Testing Framework. Code to generate all figures is available at https://github.com/unm-hscl/ajthor-vigsiv-HSCC2020.

5.1 Stochastic Chain of Integrators

We consider a 2-D stochastic chain of integrators [36], in which the input appears at the 2\(^{nd}\) derivative and each element of the state vector is the discretized integral of the element that follows it. The dynamics with sampling time \(T\) are given by:

\[
x_{k+1} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} u_k + w_k
\]

(47)

where \(w_k\) is an i.i.d. disturbance defined on the probability space \((\mathcal{W}, \mathcal{B}(\mathcal{W}), \Pr_{w_k})\). We consider three distributions for the disturbance: 1) A Gaussian distribution with variance \(\Sigma = 0.01I\) such that \(w_k \sim \mathcal{N}(0, \Sigma)\); 2) A beta distribution \(w_k \sim 0.1\text{Beta}(\alpha, \beta)\), with PDF

\[
f(x \mid \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha) \Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1}
\]

(48)

described in terms of the Gamma function \(\Gamma\) and positive shape parameters \(\alpha = 2, \beta = 0.5\); and 3) An exponential distribution \(w_k \sim 0.01\text{Exp}(\lambda)\), with the distribution parameter \(\lambda = 3\) and PDF

\[
f(x \mid \lambda) = \lambda \exp(-\lambda x)
\]

(49)

For the purpose of validation against a known model, the control policy \(\pi\) was chosen to be \(\pi_0(x) = \pi_1(x) = \ldots = 0\). The target set is defined as \(T = [-0.5, 0.5]^2\) and the safe set is defined as \(K = [-1, 1]^2\).
Figure 3: First-hitting time safety probabilities for a double integrator system with a Gaussian disturbance over the horizon $N = 5$ using Algorithm 2 with $D = 10000$ (left) and $D = 12500$ (right).

For a 2-D chain of integrators with a Gaussian disturbance $w_k \sim N(0, \Sigma)$, we computed the safety probabilities using Algorithm 1 for a time horizon of $N = 5$ for the first hitting time problem using $M = 2500$ samples. The results are shown in Fig. 2(b). We then computed the safety probabilities for the system using Algorithm 2 (RFF) with $D = 15000$ (Fig. 2(d)). This corresponds to an expected error in the kernel approximation of less than 0.05. We compared the results against a dynamic programming solution implemented in [35] in order to compare against a known “truth” model. The maximum absolute error between Algorithm 1 and the dynamic programming solution was 0.0748, and the maximum absolute error computed using Algorithm 2 was 0.0907. The absolute error between the dynamic programming solution (Fig. 2(a)) and Algorithms 1 and 2 are shown in Fig. 2(c, e). As expected, Algorithm 2 produced a higher error estimate of the safety probabilities than Algorithm 1 due to the kernel approximation (33), and the decrease in accuracy was close to what was expected for the choice of $D$ (44). Also as expected, we obtain a better approximation of the safety probabilities as $D$ is increased (Fig. 4). Fig. 3 demonstrates the effect of $D$ on the accuracy for the 2-D stochastic chain of integrators. Computation times for both algorithms are shown in Table 1.

As shown in Table 1, the first-hitting time problem computations took 2.30 seconds with $M = 2500$ samples for a 2-D stochastic chain of integrators using Algorithm 1, a two order of magnitude improvement over dynamic programming. However, Algorithm 2 took 22.94 seconds to compute the safety probabilities with $M = 2500$ and $D = 15000$. This is because the choice of $D$ determines the size of the matrix $H$ in (41), and for $D \gg M$, the algorithm does not produce a more efficient approximation of the safety probabilities.

We consider the same system with a beta distribution disturbance $w_k \sim 0.1 \text{Beta}(2, 0.5)$ and an exponential distribution disturbance $w_k \sim 0.01 \text{Exp}(3)$. We computed the safety probabilities for a time horizon of $N = 50$ as shown in Fig. 5. Algorithm 1 is agnostic to the complexities of the disturbance, which means handling arbitrary disturbances is straightforward.

Figure 4: Mean absolute error of the value function approximation $|V_0 - \hat{V}_0|$ vs. $D$ for the 2-D stochastic chain of integrators with a Gaussian disturbance.

Figure 5: First-hitting time safety probabilities for a double integrator system with a beta distribution disturbance $w_k \sim 0.1 \text{Beta}(2, 0.5)$ (left) and an exponential distribution disturbance $w_k \sim 0.01 \text{Exp}(3)$ (right) over the horizon $N = 50$ using Algorithm 1.
5.2 Planar Quadrotor

We implemented Algorithms 1 and 2 on a million-dimensional nonlinear system with a non-Gaussian disturbance. We chose a repeated planar quadrotor system, in which 170,000 six-dimensional planar quadrotors must all take off and reach a desired altitude. This can be interpreted as a simplification of formation control for large swarms of quadrotors, where we compute the safety probabilities for the entire swarm as the quadrotors are controlled to reach a particular configuration. The nonlinear dynamics of a single quadrotor are given by

\[
\begin{align*}
    m\ddot{x} &= -(u_1 + u_2) \sin(\theta) \\
    m\ddot{y} &= (u_1 + u_2) \cos(\theta) - mg \\
    I\ddot{\theta} &= r(u_1 - u_2)
\end{align*}
\]

where \(x\) is the lateral position, \(y\) is the vertical position, \(\theta\) is the pitch, and we have the constants intertia \(I = 2\), length \(r = 2\), mass \(m = 5\), and \(g = 9.8\) is the gravitational constant.

For a single quadrotor, the state space is \(\mathcal{X} \subseteq \mathbb{R}^6\), with state vector given by \(z = [x, \dot{x}, y, \dot{y}, \theta, \dot{\theta}]^\top\), and the input space is \(\mathcal{U} \subseteq \mathbb{R}^2\), with input vector \(u = [u_1, u_2]^\top\). The input is chosen to be a reference tracking controller, computed using a linearization of the system dynamics about a hover point. The controller is therefore a stationary policy that depends only on the current state of the system. We discretize the nonlinear dynamics in time using an Euler approximation \((T = 0.25)\), and add an affine disturbance. The disturbance \(\omega[\cdot]\) is a Markov process with elements \(\omega_k\) defined on the probability space \((\mathcal{W}, \mathcal{B}(\mathcal{W}), \Pr_\omega)\). We consider two cases: 1) A Gaussian disturbance with variance \(\Sigma = \text{diag}(1 \times 10^{-3}, 1 \times 10^{-2}, 1 \times 10^{-2}, 1 \times 10^{-3}, 1 \times 10^{-3}, 1 \times 10^{-5})\) such that \(\omega_k \sim \mathcal{N}(0, \Sigma)\); and 2) A disturbance with beta distribution, such that \(\omega_k \sim \text{Beta}(\alpha, \beta)\) \((48)\) with shape parameters \(a = 2\), \(\beta = 0.5\). The beta disturbance has a non-zero mean, and can be interpreted as wind, such that the dynamics are biased in a particular direction. For a single planar quadrotor, the safe set is defined as \(\mathcal{K} = \{z \in \mathbb{R}^6 : |z_1| < 1, 0 \leq z_3 < 0.8\}\), and the target set is defined as \(\mathcal{T} = \{z \in \mathbb{R}^6 : |z_1| < 1, 0.8 \leq z_3\}\). For repeated quadrotors, we define the safe sets and target sets as a series of parallel tubes, such that no quadrotor may enter into the safe set of an adjacent quadrotor. This can be interpreted as ensuring the quadrotors do not collide in midair. This means the quadrotors must all reach an altitude of 0.8 while remaining within their respective tube.

We first computed the safety probabilities for a single planar quadrotor with a reference tracking policy in order to demonstrate the feasibility of Algorithms 1 and 2. We used \(M = 1000\) samples over a time horizon of \(N = 5\) and \(D = 15000\) for Algorithm 2 using a Gaussian and a beta distribution disturbance. The results are shown in Fig. 6.

We then computed the safety probabilities for a repeated planar quadrotor system, consisting of 170,000 individual quadrotors. We generally cannot use Algorithm 1 for a system of this size due to the computational requirements of storing and computing the matrix \(G\) \((17)\) for large sample sets. However, due to the repeated structure of the dynamics, we do not need to choose samples that are far outside the safe set for any single quadrotor, allowing us to choose fewer samples overall. In practice, without prior knowledge of the dynamics, the number of samples needed to fully characterize the state space can make computing the safety probabilities using Algorithm 1 intractable.

For this example, we chose \(M = 1000\) and \(D = 1000\) in order to demonstrate the reduced computational complexity of Algorithm 2 for high-dimensional systems. We then computed the safety probabilities for a time horizon of \(N = 1\) and a single test point. The safety probabilities for all 170,000 planar quadrotors were computed in 1.32 hours using Algorithm 1 (Table 1). We then computed the safety probabilities using Algorithm 2, which took 51.83 seconds. As expected, the computation time using Algorithm 2 was almost two orders of magnitude faster than Algorithm 1. We also reduced the number of frequency samples to \(D = 500\) in order to demonstrate the reduced computation time of the approximation as \(D\) is decreased. The computation time with \(D = 500\) was 20.39 seconds.

\(M\) vs. \(D\). The quality of the approximation obtained from Algorithm 2 is dependent on the number of samples \(M\) and on the number of frequency samples \(D\). In some cases, the number of frequency samples from \(A\) that are required to approximate the kernel can mean the approximation using RFF does not provide a lower-dimensional approximation of \(G\) \((17)\). However, when \(M \gg D\), or when the system is high-dimensional, RFF can significantly reduce the computational complexity of computing the safety probabilities. By choosing a lower value of \(D\), we exchange numerical accuracy for lower computation times of the algorithm.

5.3 Algorithm Parameters

The quality of the approximation obtained by Algorithm 1 and Algorithm 2 can be greatly affected by the choice of parameters \(\lambda\) and \(\sigma\). A cross-validation scheme to empirically choose these parameters is presented in [18, Section 6].

Similarly, the parameter \(D\) in Algorithm 2 can also affect the approximation of the safety probabilities. In order to approximate the kernel to within some probabilistic error bound \(\rho\), [20] suggests that \(D\) should be chosen according to \((44)\). An alternative choice of \(D\) can be found in [29], which suggest a more optimal rate. This rate ideally determines the minimal size of \(H\) in \((39)\), which is larger than \(G\) if \(D > M\). However, in practice, lower values of \(D\) still produce quality
We also plan to release an implementation of the algorithm we presented a sample-based method to compute the stochastic reachability safety probabilities for a planar quadrotor system with a Gaussian disturbance and (right) with a beta distribution disturbance over the horizon \( N = 5 \).

![Figure 6: (left) First-hitting time safety probabilities for a planar quadrotor system with a Gaussian disturbance and (right) with a beta distribution disturbance over the horizon \( N = 5 \).](https://sreachtools.github.io/)

6 CONCLUSION & FUTURE WORK

We presented a sample-based method to compute the stochastic reachability safety probabilities for high-dimensional Markov control processes. This approach is applicable to arbitrary disturbances and is model-free, meaning it does not rely upon a known stochastic kernel. We presented an algorithm based on random Fourier features to compute the stochastic reachability safety probabilities for the first-hitting time problem and demonstrated it on a million-dimensional system to showcase the efficiencies of the computation.

We plan to extend this to controller synthesis for stochastic reachability, and explore applications to DTSHS models. We also plan to release an implementation of the algorithm as part of a Matlab toolbox for stochastic reachability, SReachTools [35], available at https://sreachtools.github.io/.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation and the Air Force Office of Scientific Research under NSF Grant Numbers CMMI-1254990, IIS-1528047, and CNS-1836900, and by the Air Force Research Lab under AFRL Cooperative Agreement FA9453-18-2-0022.

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.

A VECTOR-VALUED REPRODUCING KERNEL HILBERT SPACES AND RANDOM FOURIER FEATURES

Recall that a stochastic kernel maps a probability measure \( Q(\cdot \mid x, u) \) to each \( x \in \mathcal{X} \) and \( u \in \mathcal{U} \). Because we consider that \( \mathcal{X} \) and \( \mathcal{U} \) are compact Borel spaces, we note that a probability measure \( Q(\cdot \mid x, u) \) assigned to the measurable space \((\mathcal{X}, \mathcal{B}(\mathcal{X}))\) is a continuous, real-valued function. Using this, a probability measure \( Q(\cdot \mid x, u) \) can be represented as an element within an RKHS of vector-valued functions, which is a Hilbert space of functions on \( \mathcal{X} \) of the form \( f : \mathcal{X} \to \mathbb{R}^{d} \).

Let \( \mathcal{V} \) be a real Hilbert space with inner product \( \langle \cdot, \cdot \rangle_{\mathcal{V}} \). Let \( \mathcal{H}_{\mathcal{F}} \) denote the linear space of functions \( f : \mathcal{X} \to \mathcal{V} \), which is a vector-valued reproducing kernel Hilbert space [18] with inner product \( \langle \cdot, \cdot \rangle_{\mathcal{H}_{\mathcal{F}}} \). By the Riesz representation theorem [2, Theorem 16], there exists, for every \( x \in \mathcal{X} \) and \( v \in \mathcal{V} \), an operator from \( \mathcal{V} \) to \( \mathcal{H}_{\mathcal{F}} \), written \( \Gamma(x, \cdot) v \in \mathcal{H}_{\mathcal{F}} \), such that \( \forall f \in \mathcal{H}_{\mathcal{F}} \),

\[
\langle v, f(x) \rangle_{\mathcal{V}} = \langle f, \Gamma(x, \cdot) v \rangle_{\mathcal{H}_{\mathcal{F}}} \tag{51}
\]

Letting \( \mathcal{L}(\mathcal{V}) \) be the space of bounded linear operators from \( \mathcal{V} \) into itself, we define the reproducing kernel \( \Gamma(x, x') \in \mathcal{L}(\mathcal{V}) \) as \( \Gamma(x, x') v \). A function \( \Gamma : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{V}) \) is a kernel according to [18, Proposition 2.1] and [11] if it satisfies the following properties: 1) For every \( v, v' \in \mathcal{V} \), we have

\[
\langle v, \Gamma(x, x') v' \rangle_{\mathcal{H}_{\mathcal{F}}} = \langle \Gamma(x, \cdot) v, \Gamma(x', \cdot) v' \rangle_{\mathcal{H}_{\mathcal{F}}} \tag{52}
\]

2) \( \Gamma(x, x') \in \mathcal{L}(\mathcal{V}) \), \( \Gamma(x, x') = \Gamma(x', x)^\ast \), where \( ^\ast \) denotes the adjoint of a bounded linear operator in \( \mathcal{L}(\mathcal{V}) \), and \( \Gamma(x, x) \) is in \( \mathcal{L}_+(\mathcal{V}) \), the cone of nonnegative bounded linear operators. 3) For all \( n \in \mathbb{N}, x_1, \ldots, x_n \in \mathcal{X} \), and \( v_1, \ldots, v_n \in \mathcal{V}, \)

\[
\sum_{i=1}^{n} \sum_{j=1}^{n} \langle v_i, \Gamma(x_i, x_j) v_j \rangle_{\mathcal{V}} \geq 0 \tag{53}
\]

We correlate the RKHS \( \mathcal{H}_{\mathcal{X}} \) with the general Hilbert space \( \mathcal{V} \), and define the vector-valued reproducing kernel Hilbert spaces \( \mathcal{H}_{\mathcal{V}} \) and \( \mathcal{H}_{\mathcal{0}} \), with kernels \( \vartheta : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{H}_{\mathcal{X}}) \) and \( \varrho : \mathcal{U} \times \mathcal{U} \to \mathcal{L}(\mathcal{H}_{\mathcal{V}}) \). Further, we define the reproducing kernel Hilbert space \( \mathcal{H}_{\mathcal{K}} \) with the kernel \( \Upsilon = \vartheta \otimes \varrho \), such that if we choose \( \vartheta(x, x') \) and \( \varrho(u, u') \) to be

\[
\vartheta(x, x') = K_{\mathcal{X}}(x, x') \operatorname{Id}_{\mathcal{X}} \tag{54}
\]

\[
\varrho(u, u') = K_{\mathcal{U}}(u, u') \operatorname{Id}_{\mathcal{X}} \tag{55}
\]

where \( \operatorname{Id}_{\mathcal{X}} : \mathcal{H}_{\mathcal{X}} \to \mathcal{H}_{\mathcal{X}} \) is the identity map on \( \mathcal{H}_{\mathcal{X}} \), then the kernel \( \Upsilon : (\mathcal{X}, \mathcal{U}) \times (\mathcal{X}, \mathcal{U}) \to \mathcal{L}(\mathcal{H}_{\mathcal{V}}) \) is given by

\[
\Upsilon((x, u), (x', u')) = K_{\mathcal{X}}(x, x') K_{\mathcal{U}}(u, u') \operatorname{Id}_{\mathcal{X}} \tag{56}
\]

using the tensor product of kernels. We can easily verify that this choice of kernel satisfies the requirements in [18,
We can verify that this is a valid kernel according to [18].

\[ \langle \mu(x,u), f \rangle_{\mathcal{H}} = \langle \mu, Y((x,u), \cdot) f \rangle_{\mathcal{Y}} \]  

(57)

and the dependence of \( \mu(x,u) \) on \( x \) and \( u \) can be characterized by the kernel \( Y \).

We compute the approximation of the kernel using random Fourier features in (33) and (35) [10]. Using the tensor product definition in (36), the kernel \( Y \) in (56) can be written as

\[ Y((x,u), (x',u')) \approx \langle z(x), z(x') \rangle \langle z(u), z(u') \rangle \text{Id}_X \]  

(58)

Let the approximate kernel \( \tilde{Y} : (X, \mathcal{U}) \times (X, \mathcal{U}) \rightarrow \mathcal{L}(\mathcal{H}) \) computed via RFF be defined as

\[ \tilde{Y}((x,u), (x',u')) = \langle z(x), z(x') \rangle \langle z(u), z(u') \rangle \text{Id}_X \]  

(59)

We can verify that this is a valid kernel according to [18]. Substituting the kernel \( \tilde{Y} \) in (57) gives an approximation of the inner product \( \langle \cdot, \cdot \rangle_{\mathcal{H}} \). This means an approximation of the kernel functions via RFF can be used to approximate the vector-valued kernel function \( Y \), and thus can be used to approximate the conditional distribution embedding.

REFERENCES

[1] Alessandro Abate, Maria Prandini, John Lygeros, and Shankar Sastry. 2008. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Automatica* 44, 11 (2008), 2724–2734.

[2] Naum Il'ich Akhiezer and Izrail Markovich Glazman. 2013. *Theory of linear operators in Hilbert space*. Courier Corporation.

[3] Nachman Aronszajn. 1950. Theory of reproducing kernels. *Transactions of the American mathematical society* 68, 3 (1950), 337–404.

[4] Stanley Bak and Parasara Sridhar Duggirala. 2017. HyLaa: A tool for computing simulation-equivalent reachability for linear systems. In *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*. ACM, 173–178.

[5] Stanley Bak and Parasara Sridhar Duggirala. 2018. HyLAA 2.0: A Verification Tool for Linear Hybrid Automaton Models of Cyber-Physical Systems. In *Tool Demonstrations at IEEE Real-Time Systems Symposium (Demo@RTSS)* December.

[6] Stanley Bak, Hoang-Dung Tran, and Taylor T Johnson. 2019. Numerical verification of affine systems with up to a billion dimensions. In *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*. ACM, 23–32.

[7] Amrit Singh Bedi, Alec Koppel, and Ketan Rajawat. 2019. Nonparametric Compositional Stochastic Optimization. arXiv:math.OC/1902.06011

[8] Aliun Berliquet and Christine Thomas-Agnan. 2011. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media.

[9] Dimitri P Bertsekas and Steven Shreve. 2004. *Stochastic optimal control: the discrete-time case*.

[10] Romain Brault, Markus Heinonen, and Florence Buc. 2016. Random fourier features for operator-valued kernels. In *Asian Conference on Machine Learning*, 110–125.

[11] Steffen Grunewalder, Guy Lever, Luca Baldassarre, Massi Pontil, and Arthur Gretton. 2012. Modelling transition dynamics in MDPs with RKHS embeddings. arXiv preprint arXiv:1206.4655 (2012).

[12] Nikolaos Kariotoglou, Sean Summers, Tyler Summers, Maryam Kamgarpour, and John Lygeros. 2013. Approximate dynamic programming for stochastic reachability. In *Euro. Ctrl. Conf.* 584–589.

[13] Quoc Le, Tamás Sarlós, and Alex Smola. 2013. Fastfood approximating kernel expansions in loglinear time. In *Proceedings of the international conference on machine learning*, Vol. 85.

[14] Kendra Lesser, Meeko Oishi, and R Scott Erwin. 2013. Stochastic reachability for control of spacecraft relative motion. In *52nd IEEE Conference on Decision and Control*. IEEE, 4705–4712.

[15] Guy Lever, Luca Baldassarre, Sam Patterson, Arthur Gretton, Massimiliano Pontil, and Steffen Grunewalder. 2012. Conditional mean embeddings as regressors. In *International Conference on Machine Learning (ICML)*, Vol. 5.

[16] Zhu Li, Jean-Francois Ton, Dino Oglic, and Dino Sejdinovic. 2018. Towards A Unified Analysis of Random Fourier Features. arXiv preprint arXiv:1806.09178 (2018).

[17] Giorgio Manganini, Matteo Pirotta, Marcello Restelli, Luigi Pirola, and Maria Prandini. 2015. Policy search for the optimal control of Markov Decision Processes: A novel particle-based iterative scheme. *IEEE Trans. on Cybern.* 46, 11 (2015), 2643–2655.

[18] Charles A Micchelli and Massimiliano Pontil. 2005. On learning vector-valued functions. *Neural computation* 17, 1 (2005), 177–204.

[19] Martin L Puterman. 2014. Markov decision processes: discrete stochastic dynamic programming. *John Wiley & Sons*.

[20] Ali Rahimi and Benjamin Recht. 2008. Random features for large-scale kernel machines. In *Advances in neural information processing systems*. 1177–1184.

[21] Walter Rudin. 1962. *Fourier analysis on groups*. Vol. 121967. Wiley Online Library.

[22] Hossein Sartipizadeh, Abraham Vinod, Behcet Ackimse, and Meeko Oishi. 2018. Voronoi Partition-based Scenario Reduction for Fast Sampling-based Stochastic Reachability Computation of LTI Systems. *Amer. Ctrl. Conf.* (2018), 37–44.

[23] Bernhard Scholkopf and Alexander J Smola. 2001. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

[24] John Shawe-Taylor, Nello Cristianini, et al. 2004. *Kernel methods for pattern analysis*. Cambridge university press.

[25] A. Smola, A. Gretton, L. Song, and B. Schölkopf. 2007. A Hilbert Space Embedding for Distributions. In *Int’l Conf. Algorithmic Learn. Theory*. 13–31.

[26] Le Song, Kenji Fukumizu, and Arthur Gretton. 2013. Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *IEEE Signal Processing Magazine* 30, 4 (2013), 98–111.

[27] Le Song, Jonathan Huang, Alex Smola, and Kenji Fukumizu. 2009. Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 961–968.

[28] Sadegh Esmaeil Zadeh Soudjani, Caspar Gevaerts, and Alessandro Abate. 2015. FAU ST 2: Formal Abstractions of Uncountable-State Stochastic Processes. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 272–286.

[29] Bhrathri Sripirembudur and Zoltán Szabó. 2015. Optimal rates for random fourier features. In *Advances in Neural Information Processing Systems*. 1144–1152.

[30] Ingo Steinwart and Andreas Christmann. 2008. *Support vector machines*. Springer Science & Business Media.
[31] Sean Summers and John Lygeros. 2010. Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem. *Automatica* 46, 12 (2010), 1951–1961.

[32] Adam J. Thorpe and Meeko M. K. Oishi. 2019. Model-Free Stochastic Reachability Using Kernel Distribution Embeddings. *arXiv preprint arXiv:1908.00697* (2019).

[33] Abraham Vinod, Baisravan HomChaudhuri, Christoph Hintz, Anup Parikh, Stephen Buerger, Meeko Oishi, Greg Brunson, Shakeeb Ahmad, and Rafael Fierro. 2018. Multiple Pursuer-Based Intercept via Forward Stochastic Reachability. In *Amer. Ctrl. Conf.* 1559–1566.

[34] Abraham Vinod, Sean Rice, Yuanqi Mao, Meeko Oishi, and Behçet Açıklmeşe. 2018. Stochastic Motion Planning Using Successive Convexification and Probabilistic Occupancy Functions. In *IEEE Conf. Dec. & Ctrl.* 4425–4432.

[35] Abraham P Vinod, Joseph D Gleason, and Meeko MK Oishi. 2019. SReachTools: a MATLAB stochastic reachability toolbox. In *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*. ACM, 33–38.

[36] Abraham P Vinod and Meeko MK Oishi. 2017. Scalable underapproximation for the stochastic reach-avoid problem for high-dimensional LTI systems using Fourier transforms. *IEEE control systems letters* 1, 2 (2017), 316–321.

[37] Abraham P Vinod and Meeko MK Oishi. 2018. Scalable underapproximative verification of stochastic LTI systems using convexity and compactness. In *Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control (part of CPS Week)*. ACM, 1–10.

[38] Abraham P Vinod, Vignesh Sivaramakrishnan, and Meeko MK Oishi. 2019. Piecewise-Affine Approximation-Based Stochastic Optimal Control with Gaussian Joint Chance Constraints. In *2019 American Control Conference (ACC)*. IEEE, 2942–2949.

[39] Christopher KI Williams and Carl Edward Rasmussen. 2006. *Gaussian processes for machine learning*. Vol. 2. MIT Press Cambridge, MA.