Proceedings of Machine Learning Research vol 144:1–12, 2021

Learning Approximate Forward Reachable Sets Using Separating Kernels

Adam J. Thorpe AJTHOR@UNM.EDU
Kendric R. Ortiz KENDRIC@UNM.EDU
Meeko M. K. Oishi OISHI@UNM.EDU
Department of Electrical and Computer Engineering, University of New Mexico

Abstract

We present a data-driven method for computing approximate forward reachable sets using separating kernels in a reproducing kernel Hilbert space. We frame the problem as a support estimation problem, and learn a classifier of the support as an element in a reproducing kernel Hilbert space using a data-driven approach. Kernel methods provide a computationally efficient representation for the classifier that is the solution to a regularized least squares problem. The solution converges almost surely as the sample size increases, and admits known finite sample bounds. This approach is applicable to stochastic systems with arbitrary disturbances and neural network verification problems by treating the network as a dynamical system, or by considering neural network controllers as part of a closed-loop system. We present our technique on several examples, including a spacecraft rendezvous and docking problem, and two nonlinear system benchmarks with neural network controllers.

Keywords: Stochastic reachability, kernel methods, neural network verification

1. Introduction

Reachability analysis is important in many dynamical systems for its ability to provide assurances of desired behavior, such as reaching a desired target set or anticipating potential intersection with a set known to be unsafe. Such analysis has shown utility in a variety of safety-critical applications, including autonomous cars, UAVs, and spacecraft. The forward *stochastic reachable set* describes the set of states that the system will reach with a non-zero likelihood. While most approaches for stochastic reachability are model-based, data-driven approaches are important when mathematical models may not exist or may be too complex for existing numerical approaches. In particular, the growing presence of learning-enabled components in dynamical systems warrants the development of new tools for stochastic reachability that can accommodate neural networks, look-up tables, and other model-resistant elements. In this paper, we propose a technique for data-driven stochastic reachability analysis that provides a convergent approximation to the true stochastic reachable set.

Significant advances have been made in verification of dynamical systems with neural network components. Recent work in Seidman et al. (2020); Weinan (2017) has shown that neural networks can be modeled as a nonlinear dynamical system, making them amenable in some cases to model-based reachability analysis. Typically, these approaches presume that the neural network exhibits a particular structure, such as having a particular activation function, as in Sidrane and Kochenderfer (2019); Tran et al. (2020), and exploit existing tools for forward reachability, such as Althoff (2015); Chen et al. (2013). Other approaches employ a mixed integer linear programming approach, as

in Dutta et al. (2017, 2019); Lomuscio and Maganti (2017). These techniques exploit Lipschitz constants or perform set propagation, however, the latter approach becomes intractable for large-scale systems due to vertex facet enumeration. Further, in practice, knowledge of the network structure or dynamics may not be available, or may be too complex to use with standard reachability methods. Thus, additional tools are needed to efficiently compute stochastic reachable sets.

Data-driven reachability methods provide convergent approximations of reachable sets with high confidence, but are typically unable to provide assured over-approximations. Methods have been developed that use convex hulls in Lew and Pavone (2020) and scenario optimization in Devonport and Arcak (2020). However, these approaches rely upon convexity assumptions which can be limiting in certain cases. Other data-driven approaches leverage a class of machine learning techniques known as kernel methods, including Gaussian processes in Devonport and Arcak (2020) and support vector machines in Allen et al. (2014); Rasmussen et al. (2017). However, the approach in Rasmussen et al. (2017) can suffer from stability issues and does not provide probabilistic guarantees of convergence, and the approach in Allen et al. (2014) requires that we repeatedly solve a nonlinear program offline to generate data for the SVM classifier. The approach in Devonport and Arcak (2020) requires use of a Gaussian process prior. In contrast to these approaches, we propose a method that can accommodate nonlinear dynamical systems with arbitrary stochastic disturbances.

Our approach employs a class of kernel methods known as separating kernels to form a reachable set classifier. By learning a set classifier in Hilbert space, we convert the problem of learning a set boundary into the problem of learning a classifier in a high-dimensional space of functions. Our approach extends the work in De Vito et al. (2014) to the problem of learning reachable sets for stochastic systems that overcomes the stability and convergence issues faced by existing kernel based approaches and allows for arbitrary stochastic disturbances. Our main contribution is an application of the techniques presented in De Vito et al. (2014) to the problem of learning approximate forward reachable sets and neural network verification. Similar to other data-driven approaches, the approximation proposed here does not provide guarantees in the form of assured over- or under-approximations of the forward reachable set. However, although empirically derived, the approximation can be shown to converge in probability almost surely.

The paper is structured as follows: Section 2 formulates the problem. We describe the application of kernel methods to compute approximate forward reachable sets in Section 3, and discuss their convergence properties and finite sample bounds. In Section 4, we demonstrate our approach on a realistic satellite rendezvous and docking problem, as well as two neural network verification benchmarks from Dutta et al. (2019). Concluding remarks are presented in Section 5.

2. Problem Formulation

We use the following notation throughout: Let E be an arbitrary nonempty space, and denote the σ-algebra on E by $\mathcal{E}$. Let $\wp(E)$ denote the set of all subsets (the power set) of E. If E is a topological space (Çinlar, 2011), the σ-algebra generated by the set of all open subsets of E is called the Borel σ-algebra, denoted by $\mathscr{B}(E)$. If E is countable, with σ-algebra $\wp(E)$, then it is called *discrete*. 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 $t \in T$, let X_t be a random variable. The collection of random variables $\{X_t : t \in T\}$ on $(\Omega, \mathcal{F})$ is a *stochastic process*.

2.1. System Model

Consider a general discrete-time system with dynamics given by:

$$x_{k+1} = f_k(x_k, u_k, \theta_k, w_k) \tag{1}$$

with state $x_k \in \mathcal{X} \subset \mathbb{R}^n$, input $u_k \in \mathcal{U} \subset \mathbb{R}^m$, parameters $\theta_k \in \Theta \subset \mathbb{R}^p$, and stochastic process w, comprised of the random variables w_k, $k \in [0, N]$, defined on the measurable space $(\mathcal{W}, \mathscr{B}(\mathcal{W}))$, $\mathcal{W} \subset \mathbb{R}^q$. The system evolves over a finite time horizon $k \in [0, N]$ from initial condition $x_0 \in \mathcal{X}$, which may be chosen according to initial probability measure $\mathbb{P}_0$. We assume that the dynamics and the structure of the disturbance are unknown, but that a finite sample of $M \in \mathbb{N}$ observations, $\{x_N^i\}_{i=1}^M$, taken i.i.d. from $\mathbb{P}_N$, the measure over terminal states, are available.

A special case of this formulation is a feedforward neural network (Seidman et al., 2020; Weinan, 2017), where k denotes the layer of the network, θ_k are the network parameters, and the dimensionality of the state, input, parameter, and disturbance spaces depends on k and can change at each layer. For example, in each layer, $x_k \in \mathcal{X}_k \subset \mathbb{R}^{n(k)}$, where $n(k)$ is a map to $\mathbb{N}$ and is the dimensionality of the state at each layer.

2.2. Forward Reachable Set

Consider a discrete time stochastic system (1). We consider an initial condition $x_0 \in \mathcal{X}$ and evolve (1) to compute the forward reachable set, $\mathscr{F}(x_0)$. Formally, we define the forward reachable set $\mathscr{F}(x_0)$ as in Lew and Pavone (2020), which is a modification of the definition in Mitchell (2007) or Rosolia and Borrelli (2019) that is amenable to neural network verification problems.

Definition 1 (Lew and Pavone, 2020) *Given an initial condition $x_0 \in \mathcal{X}$, the forward reachable set $\mathscr{F}(x_0)$ is defined as the set of all possible future states x_N at time N for a system following a given control sequence $\{u_0, u_1, \ldots, u_{N-1}\}$.*

$$\mathscr{F}(x_0) := \{x_N = f_{N-1} \circ \cdots \circ f_0(x_0, u_0, \theta_0, w_0) \,|\, x_0 \in \mathcal{X}, u_k \in \mathcal{U}, \theta_k \in \Theta, w_k \in \mathcal{W}\} \tag{2}$$

Note that x_N is a random variable on the probability space $(\mathcal{X}, \mathscr{B}(\mathcal{X}), \mathbb{P}_N)$, and that the forward reachable set can also be viewed as the support of x_N (Vinod et al., 2017), which is the smallest closed set $\mathscr{F}(x_0) \subset \mathcal{X}$ such that $\mathbb{P}_N(x_N \in \mathscr{F}(x_0)) = 1$.

We seek to compute an approximation of the forward reachable set in (2) for a system (1) with unknown dynamics, by using the theory of reproducing kernel Hilbert spaces. We formulate the problem as learning a classifier function F in Hilbert space which describes the geometry of the forward reachable set boundary. We presume that the forward reachable set (2) is implicitly defined by the classifier F.

$$\mathscr{F}(x_0) = \{x \in \mathcal{X} \,|\, F(x) = 1\} \tag{3}$$

Hence, we seek to compute an *empirical estimate* $\tilde{F}$ of F in (3) using observations taken from the system evolution $\{x_N\}_{i=1}^M$.

Forward reachability often focuses on forming an over-approximation that is a superset of $\mathscr{F}(x_0)$. However, because our proposed approach cannot provide such a guarantee, we aim to provide an approximation of the forward reachable set $\tilde{\mathscr{F}}(x_0)$ by computing $\tilde{F}$ that is convergent in probability almost surely. These probabilistic guarantees are important to ensure the consistency of our result and that our estimated classifier (and therefore the approximate reachable set) is close

to the true result with high probability. The main difficulty in this problem arises from the fact that we do not have explicit knowledge of the dynamics (1) or place any prior assumptions on the structure of the uncertainty. This makes the proposed method a useful technique in the context of existing stochastic reachability toolsets, since it can be used on black-box systems with arbitrary disturbances. Note that because we seek to estimate a classifier $\tilde{F}$, in contrast to existing approaches that employ polytopic methods, our approach does not provide a simple geometric representation.

3. Finding Forward Reachable Sets Using Separating Kernels

Let $\mathscr{H}$ denote the Hilbert space of real-valued functions $f : \mathcal{X} \to \mathbb{R}$ on $\mathcal{X}$ equipped with the inner product $\langle \cdot, \cdot \rangle_{\mathscr{H}}$ and the induced norm $\|\cdot\|_{\mathscr{H}}$.

Definition 2 (Aronszajn, 1950) *A Hilbert space $\mathscr{H}$ is a reproducing kernel Hilbert space if there exists a positive definite kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that satisfies the following properties:*

$$K(x, \cdot) \in \mathscr{H} \qquad \forall x \in \mathcal{X} \tag{4a}$$

$$f(x) = \langle f, K(x, \cdot) \rangle_{\mathscr{H}} \qquad \forall f \in \mathscr{H}, \forall x \in \mathcal{X} \tag{4b}$$

where (4b) is called the reproducing property, and for any $x, x' \in \mathcal{X}$, we denote $K(x, \cdot)$ in the RKHS $\mathscr{H}$ as a function on $\mathcal{X}$ such that $x' \mapsto K(x, x')$.

Alternatively, by the Moore-Aronszajn theorem (Aronszajn, 1950), for any positive definite kernel function K, there exists a unique RKHS $\mathscr{H}$ with K as its reproducing kernel, where $\mathscr{H}$ is the closure of the linear span of functions $\{K(x, \cdot)\}$. In other words, the Moore-Aronszajn theorem allows us to define a kernel function and obtain a corresponding RKHS.

Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be the reproducing kernel function for the RKHS $\mathscr{H}$ on $\mathcal{X}$. We stipulate that the kernel also satisfies the conditions for being a *completely separating* kernel. This property ensures that the kernel function can be used to learn the support of any probability measure on $\mathcal{X}$. We induce a metric d_K on $\mathcal{X}$ via the kernel function K,

$$d_K(x, x') = \sqrt{K(x, x) + K(x', x') - 2K(x, x')} \tag{5}$$

which is known as the kernel-induced metric on $\mathcal{X}$, and is equivalent to the Euclidean metric if K is completely separating and continuous. Following De Vito et al. (2014), we define the notion of $\mathscr{H}$ separating a subset $C \subset \mathcal{X}$.

Theorem 3 (De Vito et al., 2014, Theorem 1) *An RKHS $\mathscr{H}$ separates a subset $C \subset \mathcal{X}$ if for all $x \notin C$, there exists $f \in \mathscr{H}$ such that $f(x) \neq 0$ and $f(x') = 0$ for all $x' \in C$. In this case we also say that the corresponding reproducing kernel separates C.*

According to Theorem 3, we need to determine the existence of a function in $\mathscr{H}$ that acts as a classifier for a subset $C \subset \mathcal{X}$. Since the functions $f \in \mathscr{H}$ have the form $f = \sum_i \alpha_i K(x_i, \cdot)$, this amounts to choosing a kernel function which exhibits the separating property. Note that not all kernel functions can separate every subset $C \subset \mathcal{X}$. In order to ensure that this is possible, the notion of *completely* separating kernels is defined.

Definition 4 (De Vito et al., 2014, Definition 2) *A reproducing kernel Hilbert space $\mathscr{H}$ satisfying the assumption that for all $x, x' \in \mathcal{X}$ with $x \neq x'$ we have $K(x, \cdot) \neq K(x', \cdot)$ is called completely separating if $\mathscr{H}$ separates all the subsets $C \subset \mathcal{X}$ which are closed with respect to the metric d_K. In this case, we also say that the corresponding reproducing kernel is completely separating.*

For example, with $\mathcal{X} = \mathbb{R}^n$, the Abel kernel $K(x, x') = \exp(-\|x - x'\|_2/\sigma)$, where $\sigma > 0$, is a completely separating kernel function (De Vito et al., 2014, Proposition 5). Thus, by properly selecting the kernel function K, we ensure that we can classify any subset $C \subset \mathcal{X}$.

We now turn to a discussion of the classifier F in (3). The following theorems are reproduced with modifications for simplicity from De Vito et al. (2014), and ensure we can define $\mathscr{F}(x_0)$ as the support of x_N, and that (3) holds.

Proposition 5 (De Vito et al., 2014, Proposition 2) *Assume that for all $x, x' \in \mathcal{X}$ with $x \neq x'$, $K(x, \cdot) \neq K(x', \cdot)$, the RKHS $\mathscr{H}$ with kernel K is separable, and K is measurable with respect to the product σ-algebra $\mathscr{B}(\mathcal{X}) \otimes \mathscr{B}(\mathcal{X})$. There exists a unique closed subset $\mathscr{F}(x_0) \subset \mathcal{X}$ with $\mathbb{P}_N(x_N \in \mathscr{F}(x_0)) = 1$ satisfying the following property: if C is a closed subset of $\mathcal{X}$ and $\mathbb{P}_N(x_N \in C) = 1$ then $\mathscr{F}(x_0) \subset C$.*

For any subset $C \subset \mathcal{X}$, let $\mathscr{H}_C$ denote the closure of the linear span of functions $\{K(x, \cdot) \,|\, x \in C\}$, and define $P_C : \mathscr{H} \to \mathscr{H}$ as the orthogonal projection onto the closed subspace $\mathscr{H}_C$. Define the function $F_C : \mathcal{X} \to \mathbb{R}$ such that $F_C(x) := \langle P_C K(x, \cdot), K(x, \cdot) \rangle_{\mathscr{H}}$. Using this definition, we can now define the support $\mathscr{F}(x_0)$ in terms of a function F.

Theorem 6 (De Vito et al., 2014, Theorem 3) *Under the assumptions of Proposition 5 and the assumption that $K(x, x) = 1$ for all $x \in \mathcal{X}$, if $\mathscr{H}$ separates the support $\mathscr{F}(x_0)$ of the measure $\mathbb{P}_N$, then $\mathscr{F}(x_0) = \{x \in \mathcal{X} \,|\, F(x) = 1\}$.*

Thus, we aim to learn the forward reachable set (3) by identifying a function $F \in \mathscr{H}$ that separates the support in $\mathcal{X}$.

3.1. Estimating Forward Reachable Sets

The classifier F is an element of the RKHS $\mathscr{H}$, meaning it has the form $F = \sum_i \alpha_i K(x_i, \cdot)$ and admits a representation in terms of the finite support $\{x_i\}_{i=1}^M$, $M \in \mathbb{N}$, given by:

$$\tilde{F}(x) = \sum_i \alpha_i K(x_i, x) \tag{6}$$

where $\alpha_i \in \mathbb{R}$ are coefficients that depend on x. Let $\mathcal{D} = \{x_N^i\}_{i=1}^M$ be a sample of terminal states at time N taken i.i.d. from the system evolution. We seek an approximation of the forward reachable set $\mathscr{F}(x_0)$ in (3) using $\mathcal{D}$. Thus, we form an estimate $\tilde{F} \in \mathscr{H}$ of F using the form in (6) with data $\mathcal{D}$. We can view the estimate $\tilde{F}$ as the solution to a regularized least-squares problem.

$$\min_{\tilde{F}} \frac{1}{M} \sum_{i=1}^{M} |K(x_N^i, \cdot) - \tilde{F}(x_N^i)|^2 + \lambda \|\tilde{F}\|_{\mathscr{H}}^2 \tag{7}$$

where $\lambda > 0$ is the regularization parameter. The solution to (7) is unique and admits a closed-form solution, given by:

$$\tilde{F}(x) = \Phi^\top (G + M\lambda I)^{-1} \Phi \tag{8}$$

where Φ is called a feature vector, with elements $\Phi_i = K(x_N^i, x)$, and $G = \Phi\Phi^\top \in \mathbb{R}^{M \times M}$ is known as the Gram matrix, with elements $g_{ij} = K(x_N^i, x_N^j)$. A point $x \in \mathcal{X}$ is estimated to belong

to the support of x_N if $\tilde{F}(x) \geq 1 - \tau$, where τ is a threshold parameter that depends on the sample size M. Thus, we form an approximation $\tilde{\mathscr{F}}(x_0)$ of the forward reachable set $\mathscr{F}(x_0)$ in (3) as:

$$\tilde{\mathscr{F}}(x_0) = \{x \in \mathcal{X} \,|\, \tilde{F}(x) \geq 1 - \tau\} \tag{9}$$

Using this representation, we obtain an estimator $\tilde{F}(x)$ which can be used to determine if a point $x \in \mathcal{X}$ is in the approximate forward reachable set $\tilde{\mathscr{F}}(x_0)$.

3.2. Convergence

We characterize the conditions for the convergence of the empirical forward reachable set $\tilde{\mathscr{F}}(x_0)$ to $\mathscr{F}(x_0)$ via the Hausdorff distance. We assume that $\mathcal{X}$ is a topological metric space with metric d_K.

Definition 7 (Hausdorff Distance) *Let A, B be nonempty subsets of the metric space $(\mathcal{X}, d_K)$. The Hausdorff distance d_H between sets A and B is defined as*

$$d_H(A, B) := \inf\{\varepsilon > 0 \,|\, A \subseteq B_\varepsilon, B \subseteq A_\varepsilon\} \tag{10}$$

where $A_\varepsilon := \bigcup_{y \in A}\{x \in \mathcal{X} \,|\, d_K(x, y) \leq \varepsilon\}$ denotes the set of all points in $\mathcal{X}$ within a ball of radius ε around A.

The Hausdorff distance gives us a method to measure convergence of the estimate $\tilde{\mathscr{F}}(x_0)$ to the true reachable set $\mathscr{F}(x_0)$. In fact, $\lim_{M\to\infty} d_H(\tilde{\mathscr{F}}(x_0), \mathscr{F}(x_0)) = 0$ almost surely under mild conditions on the regularization and threshold parameters, λ and τ (De Vito et al., 2014, Theorem 6). As the sample size M increases, if τ is chosen according to:

$$\tau = 1 - \min_{1 \leq i \leq M} \tilde{F}(x_N^i) \tag{11}$$

and under the condition that $\lim_{M\to\infty} \lambda = 0$, the empirical forward reachable set $\tilde{\mathscr{F}}(x_0)$ converges in probability almost surely to the true forward reachable set $\mathscr{F}(x_0)$.

Furthermore, De Vito et al. (2014, Theorem 7) shows that the approximation admits finite sample bounds on the error of the estimated classifier function. Let $T : \mathscr{H} \to \mathscr{H}$ be the integral operator with kernel K. If $\sup_{x\in\mathcal{X}} \|T^{-s/2} T^\dagger T K(x, \cdot)\| \leq \infty$ with $0 < s \leq 1$, where $T^\dagger$ denotes the pseudo-inverse of T, and the eigenvalues of T satisfy $\nu_j = \mathcal{O}(j^{-1/b})$ for some $0 < b \leq 1$ (see Caponnetto and De Vito, 2007, Proposition 3), then for $M \geq 1$ and $\delta > 0$, if $\lambda = M^{-1/(2s+b+1)}$ and D is a suitable constant, then

$$\sup_{x\in\mathcal{X}} |F(x) - \tilde{F}(x)| \leq D\left(\frac{1}{M}\right)^{\frac{s}{2s+b+1}} \tag{12}$$

with probability $1 - 2e^{-\delta}$. These guarantees ensure the accuracy of our results in a probabilistic sense, meaning that by increasing the sample size M, the estimate will approach the true result, and that for a finite sample size M, our result is close to the true result with high probability.

As a final remark, we note that although the formulation in (3) is extensible to level set estimation, which involves learning a set $\mathscr{F}(x_0; \alpha) := \{x \in \mathcal{X} \,|\, F(x) \geq \alpha\}$ with $\alpha \in [0, 1]$, the convergence of $\tilde{F}$ to F does not imply that the approximate level sets converge to the true level sets. This requires further analysis that is beyond the scope of the current work.

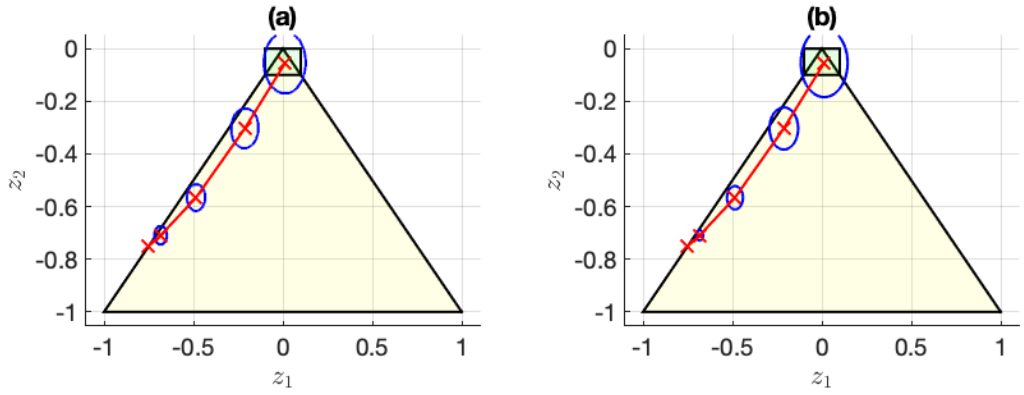

Figure 1: (a) The cross-section of the approximate forward reachable set $\tilde{\mathscr{F}}(x_0)$ computed using $M = 100$ trajectories over a time horizon of $N = 5$ from the initial condition $z_0 = [-0.75, -0.75, 0, 0]^\top$ with $\dot{x} = \dot{y} = 0$ is a good approximation of (b) the cross-section of the forward reachable set computed using the mean and variance of the Gaussian disturbance. The line-of-sight cone of the spacecraft is shown in yellow, the target set is in green, and the unperturbed trajectory and initial condition is shown in red.

4. Numerical Results

For all examples, we choose an Abel kernel $K(x, x') = \exp(-\|x - x'\|_2/\sigma)$, $\sigma > 0$. We chose the parameters to be $\sigma = 0.1$, τ according to (11), and $\lambda = 1/M$, where M is the sample size used to construct the classifier. As noted in De Vito et al. (2014), σ can be chosen via cross-validation and we require that λ be chosen such that $\lim_{M\to\infty} \lambda = 0$. 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 available at: https://github.com/unm-hscl/ajthor-ortiz-L4DC2021

4.1. Clohessy-Wiltshire-Hill System

We first consider a realistic example of spacecraft rendezvous and docking to compare our technique against a known result. The dynamics of a CWH system are defined in Lesser et al. (2013) as:

$$\ddot{x} - 3\omega^2 x - 2\omega\dot{y} = F_x/m_d \qquad \ddot{y} + 2\omega\dot{x} = F_y/m_d \tag{13}$$

with state $z = [x, y, \dot{x}, \dot{y}]^\top \in \mathcal{X} \subseteq \mathbb{R}^4$, input $u = [F_x, F_y]^\top \in \mathcal{U} \subseteq \mathbb{R}^2$, where $\mathcal{U} = [-0.1, 0.1] \times [-0.1, 0.1]$, and parameters ω, m_d. We first discretize the dynamics in time and then apply an affine Gaussian disturbance w, which is a stochastic process defined on $(\mathcal{W}, \mathscr{B}(\mathcal{W}))$ with variance $\Sigma = \text{diag}(1 \times 10^{-4}, 1 \times 10^{-4}, 5 \times 10^{-8}, 5 \times 10^{-8})$ such that $w_k \sim \mathcal{N}(0, \Sigma)$.

With initial condition $z_0 = [-0.75, -0.75, 0, 0]^\top$, we compute an open-loop control policy π using a chance-constrained algorithm from Vinod and Oishi (2019) implemented in SReachTools (Vinod et al., 2019). The control policy is designed to dock with another spacecraft while remaining within a line of sight cone. We then simulated $M = 100$ trajectories to collect a sample $\mathcal{D}$ consisting of the resulting terminal states $\{z_N^i\}_{i=1}^M$. A classifier was then computed using (8). In order to depict

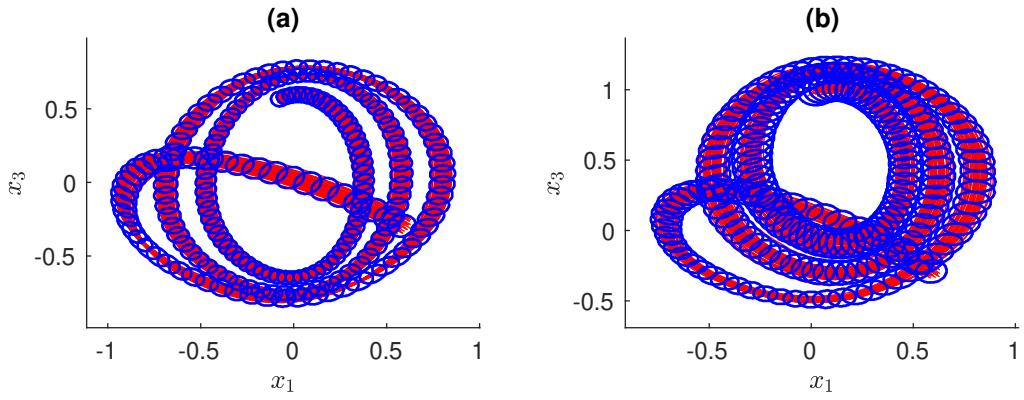

Figure 2: (a) Cross-section of the approximate forward reachable set $\tilde{\mathscr{F}}(x_0)$ for the TORA model over a time horizon of $N = 200$ using $M = 50$ trajectories, validated against simulated trajectories. (b) Cross-section of the approximate forward reachable sets $\tilde{\mathscr{F}}(x_0)$ for the TORA model with an affine beta distribution disturbance $w_k \sim 0.01\ \mathrm{Beta}(2, 0.5)$. The observed trajectories are shown in red.

the sets graphically, we then chose a small region around the mean of the observed state values and computed a visual representation of the forward reachable set by sampling 10,000 points uniformly over the region and connecting evaluations where $\tilde{F}(\cdot) = 1$ along the set boundary.

Figure 1 compares the computed approximate forward reachable sets to forward reachable sets computed using the known Gaussian disturbance properties. We can see that the approximate forward reachable sets $\tilde{\mathscr{F}}(x_0)$ in Figure 1(a) are close to the variance ellipses produced from the known disturbance properties in 1(b), demonstrating that our approach provides a good estimate of the support of the underlying distribution. Computation time for the approximate forward reachable sets was 1.582 seconds using the kernel based approach. This technique could be accelerated with approximative speedup techniques that are common to kernel methods, such as random Fourier features (Rahimi and Recht, 2008) or FastFood (Le et al., 2013). Additionally, incorporating active or adversarial sampling, as in Lew and Pavone (2020), could reduce computation time by reducing the number of observations needed to compute the classifier.

4.2. Neural Network Verification

We now demonstrate our approach on a set of dynamical system benchmarks with neural network controllers as described in Dutta et al. (2019). We define a feed-forward neural network controller $\pi : \mathcal{X} \to \mathcal{U}$ as $\pi(z) = g_L \circ \cdots \circ g_0(z, \theta_0)$, with activation functions $g_i(z, \theta_i)$, $i = 1, \ldots, L-1$ that depend on the parameters θ_i. The function g_0 maps the states into the first layer, and g_L is a function that maps the output of the last layer into the control space. After generating observations of the states, we then assume no knowledge of the structure of π or the dynamics.

4.2.1. TORA MODEL

Consider a translational oscillations by a rotational actuator (TORA) model from Jankovic et al. (1996) with a neural network controller (Dutta et al., 2019, Benchmark 9), with dynamics given by:

$$\begin{aligned} \dot{x}_1 &= x_2 & \dot{x}_3 &= x_4 \\ \dot{x}_2 &= -x_1 + 0.1\sin(x_3) & \dot{x}_4 &= u \end{aligned} \tag{14}$$

where u is the control input, chosen by the neural network controller. The neural network is trained via an MPC algorithm to keep all state variables within the range $[-2, 2]$. We presume an initial distribution that is uniform over the range $[0.6, 0.7] \times [-0.7, -0.6] \times [-0.4, -0.3] \times [0.5, 0.6]$ and collected a sample from $M = 50$ simulated trajectories over a time horizon of $N = 200$. Using (8), we then computed a classifier to indicate whether a given point is within the approximate forward reachable set $\tilde{\mathscr{F}}(x_0)$. As before, we chose a small region around the mean of the observed trajectories and created a visual representation of the approximate forward reachable set by connecting evaluations where $\tilde{F}(\cdot) = 1$ along the set boundary. We validate our approach via Monte Carlo simulation. Figure 2(a) shows a cross-section of the approximate forward reachable set $\tilde{\mathscr{F}}(x_0)$ (9). As expected, we can see the forward reachable set encompasses the simulated trajectories. The computation time was 52.907 seconds for computing the approximate forward reachable sets $\tilde{\mathscr{F}}(x_0)$ over the time horizon $N = 200$.

We then add an arbitrarily chosen affine disturbance w to the dynamics in (14), which is a stochastic process consisting of the random variables with a beta distribution $w_k \sim 0.01\ \mathrm{Beta}(\alpha, \beta)$, with PDF $f(x \,|\, \alpha, \beta) = x^{\alpha-1}(1-x)^{\beta-1}/B(\alpha, \beta)$ where $B(\alpha, \beta) = \Gamma(\alpha)\Gamma(\beta)/\Gamma(\alpha+\beta)$ and Γ is the Gamma function, with shape parameters $\alpha = 2$, $\beta = 0.5$. We then implemented our method on the stochastic system and computed the approximate reachable sets. As shown in Figure 2(b), the cross-section shows a larger variation in the trajectories due to the added disturbance, resulting in larger approximate forward reachable sets.

4.2.2. DRONE MODEL

Lastly, consider a 12-DOF quadrotor system with nonlinear dynamics defined in Bansal et al. (2016) with a neural network controller described in Dutta et al. (2019). This system has proven challenging for existing reachability tools, since the trajectories diverge locally before converging, meaning the forward reachable set is difficult to compute using over-approximative interval based methods. The system is controlled via four inputs $u \in \mathbb{R}^4$ which are determined by a neural network controller. Following Dutta et al. (2019), we chose an arbitrary state close to the origin $z \in \mathbb{R}^{12}$ with initial distribution uniform over the range $x_0 \sim z + 0.05\ U(0, 1)$ and collected a sample from $M = 50$ simulated trajectories over a time horizon of $N = 50$. We calculated the approximate forward reachable sets $\tilde{\mathscr{F}}(x_0)$ as before. Figure 3(a) shows a cross-section of the approximate forward reachable sets for the problem. As expected, we can see the approximate forward reachable sets $\tilde{\mathscr{F}}(x_0)$ (9) are centered around the trajectories (in red), indicating that the approximate forward reachable sets are a good approximation of the support. The computation time for calculating the forward reachable sets was 12.830 seconds using the kernel method based approach.

We further demonstrate the capability of our algorithm by applying an arbitrarily chosen affine disturbance w to the dynamics, which as before is a stochastic process consisting of the random variables $w_k \sim \mathcal{N}(0, \Sigma)$, with $\Sigma = 0.0025I$. Figure 3(b) depicts the approximate forward reachable sets $\tilde{\mathscr{F}}(x_0)$ for the stochastic nonlinear system. This shows that our method can handle high-

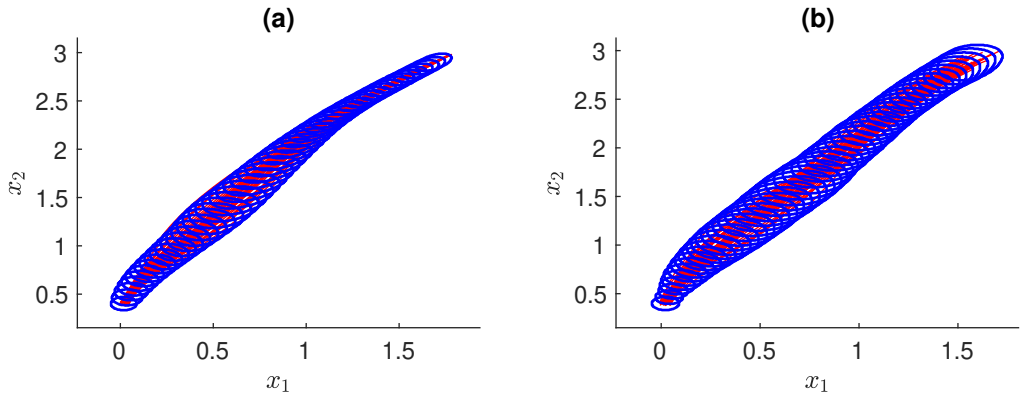

Figure 3: (a) Cross-section of the approximate forward reachable sets $\tilde{\mathscr{F}}(x_0)$ for the drone model over a time horizon $N = 50$ using $M = 50$ trajectories. (b) Cross-section of the approximate forward reachable sets $\tilde{\mathscr{F}}(x_0)$ for the drone model with an affine Gaussian disturbance $w_k \sim \mathcal{N}(0, \Sigma)$, $\Sigma = 0.0025I$. The observed trajectories are shown in red.

dimensional, stochastic nonlinear systems controlled by neural networks, and compute the approximate forward reachable sets with efficient computational time.

5. Conclusion & Future Work

We presented a method for computing forward reachable sets using reproducing kernel Hilbert spaces with separating kernel functions. We demonstrated the effectiveness of the technique at performing neural network verification, and validated its accuracy on a satellite rendezvous and docking problem with Clohessy-Wiltshire-Hill dynamics. This technique is scalable, computationally efficient, and model-free, making it well-suited for systems with learning-enabled components. We plan to investigate extension of this to other reachability problems, such as safety problems that require calculation of the backward reachable set.

Acknowledgments

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.

References

Ross E. Allen, Ashley A. Clark, Joseph A. Starek, and Marco Pavone. A machine learning approach for real-time reachability analysis. In *International Conference on Intelligent Robots and Systems*, pages 2202–2208. IEEE, 2014.

Matthias Althoff. An introduction to CORA 2015. In *Workshop on Applied Verification for Continuous and Hybrid Systems*, 2015.

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

Somil Bansal, Anayo K. Akametalu, Frank J. Jiang, Forrest Laine, and Claire J. Tomlin. Learning quadrotor dynamics using neural network for flight control. In *IEEE Conference on Decision and Control*, pages 4653–4660. IEEE, 2016.

Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

Xin Chen, Erika Ábrahám, and Sriram Sankaranarayanan. Flow*: An analyzer for non-linear hybrid systems. In *International Conference on Computer Aided Verification*, pages 258–263. Springer, 2013.

Erhan Çinlar. *Probability and Stochastics*. Springer, 2011.

Ernesto De Vito, Lorenzo Rosasco, and Alessandro Toigo. Learning sets with separating kernels. *Applied and Computational Harmonic Analysis*, 37(2):185–217, 2014.

Alex Devonport and Murat Arcak. Data-driven reachable set computation using adaptive Gaussian process classification and Monte Carlo methods. In *American Control Conference*, pages 2629–2634, 2020.

Alex Devonport and Murat Arcak. Estimating reachable sets with scenario optimization. In *Proceedings of Machine Learning Research*, volume 120, pages 75–84, 2020.

Souradeep Dutta, Susmit Jha, Sriram Sanakaranarayanan, and Ashish Tiwari. Output range analysis for deep neural networks. *arXiv preprint arXiv:1709.09130*, 2017.

Souradeep Dutta, Xin Chen, Susmit Jha, Sriram Sankaranarayanan, and Ashish Tiwari. Sherlock-a tool for verification of neural network feedback systems. In *International Conference on Hybrid Systems: Computation and Control*, pages 262–263, 2019.

Mrdjan Jankovic, Dan Fontaine, and Petar V. Kokotovic. TORA example: cascade- and passivity-based control designs. *IEEE Transactions on Control Systems Technology*, 4(3):292–297, 1996.

Quoc Le, Tamás Sarlós, and Alex Smola. Fastfood-approximating kernel expansions in loglinear time. In *International Conference on Machine Learning*, volume 85, 2013.

Kendra Lesser, Meeko M. K. Oishi, and R. Scott Erwin. Stochastic reachability for control of spacecraft relative motion. In *IEEE Conference on Decision and Control*, pages 4705–4712, Dec 2013.

Thomas Lew and Marco Pavone. Sampling-based reachability analysis: A random set theory approach with adversarial sampling. *arXiv preprint arXiv:2008.10180*, 2020.

Alessio Lomuscio and Lalit Maganti. An approach to reachability analysis for feed-forward ReLU neural networks. *arXiv preprint arXiv:1706.07351*, 2017.

Ian M. Mitchell. Comparing forward and backward reachability as tools for safety analysis. In *International Workshop on Hybrid Systems: Computation and Control*, pages 428–443. Springer, 2007.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pages 1177–1184, 2008.

Martin Rasmussen, Janosch Rieger, and Kevin N. Webster. Approximation of reachable sets using optimal control and support vector machines. *Journal of Computational and Applied Mathematics*, 311:68–83, 2017.

Ugo Rosolia and Francesco Borrelli. Sample-based learning model predictive control for linear uncertain systems. In *IEEE Conference on Decision and Control*, pages 2702–2707. IEEE, 2019.

Jacob H. Seidman, Mahyar Fazlyab, Victor M. Preciado, and George J. Pappas. Robust deep learning as optimal control: Insights and convergence guarantees. *arXiv preprint arXiv:2005.00616*, 2020.

Chelsea Sidrane and Mykel J. Kochenderfer. OVERT: Verification of nonlinear dynamical systems with neural network controllers via overapproximation. In *Safe Machine Learning Workshop at ICLR*, 2019.

Hoang-Dung Tran, Patrick Musau, Diego Manzanas Lopez, Xiaodong Yang, Luan Viet Nguyen, Weiming Xiang, and Taylor Johnson. NNV: A tool for verification of deep neural networks and learning-enabled autonomous cyber-physical systems. In *International Conference on Computer-Aided Verification*, 2020.

Abraham P. Vinod and Meeko M. K. Oishi. Affine controller synthesis for stochastic reachability via difference of convex programming. In *IEEE Conference on Decision and Control*, pages 7273–7280. IEEE, 2019.

Abraham P. Vinod, Baisravan HomChaudhuri, and Meeko M. K. Oishi. Forward stochastic reachability analysis for uncontrolled linear systems using Fourier transforms. In *International Conference on Hybrid Systems: Computation and Control*, pages 35–44, 2017.

Abraham P. Vinod, Joseph D. Gleason, and Meeko M. K. Oishi. SReachTools: A MATLAB stochastic reachability toolbox, April 16–18 2019. https://sreachtools.github.io.

E. Weinan. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5(1):1–11, 2017.