The Linear Programming Approach to Reach-Avoid Problems for Markov Decision Processes

Nikolaos Kariotoglou * KARIOTO@CONTROL.EE.ETHZ.CH
Automatic Control Laboratory, Information Technology and Electrical Engineering
ETH Zürich, CH-8092, Switzerland

Maryam Kamgarpour MKAMGAR@CONTROL.EE.ETHZ.CH
Automatic Control Laboratory, Information Technology and Electrical Engineering
ETH Zürich, CH-8092, Switzerland

Tyler H. Summers TYLER.SUMMERS@UTDALLAS.EDU
Control, Optimization, and Networks Lab, Mechanical Engineering
The University of Texas at Dallas, Texas, TX 75080, USA

John Lygeros LYGEROS@CONTROL.EE.ETHZ.CH
Automatic Control Laboratory, Information Technology and Electrical Engineering
ETH Zürich, CH-8092, Switzerland

Abstract

One of the most fundamental problems in Markov decision processes is analysis and control synthesis for safety and reachability specifications. We consider the stochastic reach-avoid problem, in which the objective is to synthesize a control policy to maximize the probability of reaching a target set at a given time, while staying in a safe set at all prior times. We characterize the solution to this problem through an infinite dimensional linear program. We then develop a tractable approximation to the infinite dimensional linear program through finite dimensional approximations of the decision space and constraints. For a large class of Markov decision processes modeled by Gaussian mixtures kernels we show that through a proper selection of the finite dimensional space, one can further reduce the computational complexity of the resulting linear program. We validate the proposed method and analyze its potential with a series of numerical case studies.

1. Introduction

A wide range of controlled physical systems can be modeled using the framework of Markov decision processes (MDPs) [1, 2]. Depending on the problem at hand, several objectives can be formulated for an MDP. These include maximization of a reward function or satisfaction of a specification defined by a formal language. Safety and reachability are two of the most fundamental specifications for a dynamical system. In a reach-avoid problem for an MDP, the objective is to maximize the probability of reaching a target set in a given time horizon while staying in a safe set [3]. The stochastic reach-avoid framework has been applied to various problems including aircraft conflict detection [4, 5], camera networks [6] and building evacuation strategies under randomly evolving hazards [7].

*. The work of N. Kariotoglou was supported by the Swiss National Science Foundation under grant number 200021_137876.

The dynamic programming (DP) principle characterizes the solution to the stochastic reach-avoid problem with uncountable state and action spaces [8]. However, it is intractable to find the reach-avoid value function through the DP equations. One can approximate the DP equations on a finite grid defined over the MDP state and action spaces. Gridding techniques are theoretically attractive since they can provide explicit error bounds for the approximation of the value function under general Lipschitz continuity assumptions [9, 10]. In practice, the complexity of gridding based techniques suffers from the infamous Curse of Dimensionality. Thus, such approaches are only applicable to relatively low dimensional problems. For general stochastic reach-avoid problems, the sum of state and control space dimensions that can be addressed with existing tools is at most five. An important problem is therefore to explore alternative approximation techniques to push this limit further.

Several researchers have developed approximate dynamic programming (ADP) techniques for various classes of stochastic control problems [11, 12]. Most of the existing work has focused on problems where the state and control spaces are finite but too large to directly solve DP recursions. Our work is motivated by the technique discussed in [13] where the authors develop an ADP method for optimal control of an MDP with finite state and action spaces and an infinite horizon additive stage cost. In this approach, the value function of the stochastic control problem is characterized as the solution to a linear program (LP). For optimal control of MDPs with uncountable state and action spaces and an additive stage cost, an infinite dimensional linear program has been developed to characterize the value function [14]. The LP approach to stochastic reachability problem for MDPs over uncountable state and action spaces and an infinite horizon was first proposed in [15], however, no computational approach to this problem was provided. In general, LP approaches to ADP are desirable since several commercially available software packages can handle LP problems with large numbers of decision variables and constraints.

We develop a method to approximate the optimal value function and policy of a stochastic reach-avoid problem over uncountable state and action spaces. Our contributions are as follows: First, we derive an infinite dimensional LP formulated over the space of Borel measurable functions and prove its equivalence to the standard DP-based solution approach for the stochastic reach-avoid problem. Second, we prove that through restricting the infinite dimensional decision space to a finite dimensional subspace spanned by a collection of basis functions, we obtain an upper bound on the stochastic reach-avoid value function. Third, we use randomized optimization to obtain a tractable finite dimensional LP with probabilistic feasibility guarantees. Fourth, we focus on numerical validation of the LP approach to stochastic reach-avoid problems. As such, we propose a class of basis functions for reach-avoid problems on MDPs with Gaussian mixture kernels. We then develop several benchmark problems to test the scalability and accuracy of the method.

The rest of the paper is organized as follows. In Section 2 we introduce the stochastic reach-avoid problem for MDPs and formulate an infinite dimensional LP that characterizes its solution. In Section 3 we derive an approach to approximate the solution to the infinite LP through restricting the decision space to a finite dimensional subspace using basis functions and reducing the infinite constraints to finite constraints through randomized sampling. Section 4 proposes Gaussian radial basis functions to analytically compute operations arising in the LP for MDPs with Gaussian mixture kernels. In Section 5 we validate the accuracy and scalability of the solution approach with three case studies.

2. Stochastic reach-avoid problem

We consider a discrete-time controlled stochastic process $x_{t+1} \sim Q(dx|x_t, u_t)$, $(x_t, u_t) \in \mathcal{X} \times \mathcal{U}$. Here, $Q : \mathcal{B}(\mathcal{X}) \times \mathcal{X} \times \mathcal{U} \to [0,1]$ is a transition kernel and $\mathcal{B}(\mathcal{X})$ denotes the Borel σ-algebra of $\mathcal{X}$. Given a state control pair $(x_t, u_t) \in \mathcal{X} \times \mathcal{U}$, $Q(A|x_t, u_t)$ measures the probability of x_{t+1} belonging to the set $A \in \mathcal{B}(\mathcal{X})$. The transition kernel Q is a Borel-measurable stochastic kernel, that is, $Q(A|\cdot)$ is a Borel-measurable function on $\mathcal{X} \times \mathcal{U}$ for each $A \in \mathcal{B}(\mathcal{X})$ and $Q(\cdot|x, u)$ is a probability measure on $\mathcal{X}$ for each (x, u). For the rest of the paper all measurability conditions refer to Borel measurability. We allow the state space $\mathcal{X}$ to be any subset of $\mathbb{R}^n$ and assume that the control space $\mathcal{U} \subseteq \mathbb{R}^m$ is compact.

We consider a safe set $K' \in \mathcal{B}(\mathcal{X})$ and a target set $K \subseteq K'$. We define an admissible T-step control policy to be a sequence of measurable functions $\mu = \{\mu_0, \ldots, \mu_{T-1}\}$ where $\mu_i : \mathcal{X} \to \mathcal{U}$ for each $i \in \{0, \ldots, T-1\}$. The reach-avoid problem over a finite time horizon T is to find an admissible T-step control policy that maximizes the probability of x_t reaching the set K at some time $j \leq T$ while staying in K' for all $0 \leq t \leq j$. For any initial state x_0, we denote the reach-avoid probability associated with a given μ as

$$r^{\mu}_{x_0}(K, K') = \mathbb{P}^{\mu}_{x_0}\{\exists j \in [0, T] : x_j \in K \wedge \forall i \in [0, j-1],\ x_i \in K' \setminus K\}.$$

In the above, it is assumed that $[0, -1] = \emptyset$, which implies that the requirement on i is automatically satisfied when $x_0 \in K$.

2.1 Dynamic programming approach

The reach-avoid probability $r^{\mu}_{x_0}(K, K')$ can be equivalently formulated as an expected value objective function. In contrast to an optimal control problem with additive stage cost, $r^{\mu}_{x_0}(K, K')$ is a history dependent sum-multiplicative cost function [16]:

$$r^{\mu}_{x_0}(K, K') = \mathbb{E}^{\mu}_{x_0}\left[\sum_{j=0}^{T}\left(\prod_{i=0}^{j-1} \mathbb{1}_{K' \setminus K}(x_i)\right)\mathbb{1}_K(x_j)\right], \tag{1}$$

where we use the notation of $\prod_{i=k}^{j}(\cdot) = 1$ if $k > j$. Above, $\mathbb{1}_A(x)$ denotes the indicator function of a set $A \in \mathcal{B}(\mathcal{X})$. Our objective is to find $\sup_{\mu} r^{\mu}_{x_0}(K, K')$ and the optimal policy achieving the supremum. The sets K and K' can be time-varying or stochastic [17] but for simplicity we assume here that they are constant. We denote the difference between the safe and target sets by $\bar{\mathcal{X}} := K' \setminus K$ to simplify the presentation of our results.

Similar to the dynamic programming approach to an optimal control problem with additive stage cost, the solution to the reach-avoid problem is characterized by a recursion [16] as follows: Define the value functions $V^*_k : \mathcal{X} \to [0, 1]$ for $k = T-1, \ldots, 0$ as

$$\begin{aligned} V^*_T(x) &= \mathbb{1}_K(x), \\ V^*_k(x) &= \sup_{u \in \mathcal{U}}\left\{\mathbb{1}_K(x) + \mathbb{1}_{\bar{\mathcal{X}}}(x)\int_{\mathcal{X}} V^*_{k+1}(y)Q(dy|x, u)\right\}. \end{aligned} \tag{2}$$

It can be shown that $V^*_0(x_0) = \sup_{\mu} r^{\mu}_{x_0}(K, K')$ [16]. Past work has focused on approximating V^*_k recursively on a discretized grid of $\bar{\mathcal{X}}$ and $\mathcal{U}$ [8, 9, 16]. Next, we will establish the measurability and continuity properties of the reach-avoid value functions to enable the use of a linear program to approximate these functions.

Assumption 1 *For every $x \in \mathcal{X}, A \in \mathcal{B}(X)$ the mapping $u \mapsto Q(A|x,u)$ is continuous.*

Proposition 1 *Under Assumption (1), at every step k, the supremum in (2) is attained by a measurable function $\mu_k^* : \mathcal{X} \to \mathcal{U}$ and the resulting $V_k^* : \mathcal{X} \to [0,1]$ is measurable.*

Proof 1 *By induction. First, note that the indicator function $V_T^*(x) = \mathbb{1}_K(x)$ is measurable. Assuming that V_{k+1}^* is measurable we will show that V_k^* is also measurable. Define $F(x,u) = \int_{\mathcal{X}} V_{k+1}^*(y) Q(dy|x,u)$. Due to continuity of the map $u \mapsto Q(A|x,u)$ by Assumption 1, the map $u \mapsto F(x,u)$ is continuous for every x ([18, Fact 3.9]). Since $\mathcal{U}$ is compact, there exists a measurable function $\mu_k^*(x)$ that achieves the supremum [19, Corollary 1]. Furthermore, by [20, Proposition 7.29], the mapping $(x,u) \mapsto F(x,u)$ is measurable. It follows that $F(x, \mu_k^*(x))$, and hence V_k^*, is measurable as it is composition of measurable functions.*

Proposition 1 allows one to compute an optimal feedback policy at each stage k through

$$\begin{aligned}\mu_k^*(x) &= \arg\max_{u \in \mathcal{U}} \left\{ \mathbb{1}_K(x) + \mathbb{1}_{\bar{\mathcal{X}}}(x) \int_{\mathcal{X}} V_{k+1}^*(y) Q(dy|x,u) \right\} \\ &= \arg\max_{u \in \mathcal{U}} \left\{ \int_{\mathcal{X}} V_{k+1}^*(y) Q(dy|x,u) \right\}. \end{aligned} \tag{3}$$

For functions $f, g : X \to \mathbb{R}$, we use $f \leq g$ to denote $f(x) \leq g(x)$, $\forall x \in X$. It is easy to verify by induction that $0 \leq V_k^* \leq 1$, for $k = T, T-1, \ldots, 0$. Furthermore, due to the indicator functions in (2), $V_k^*(x)$ are defined on disjoint regions of $\mathcal{X}$ as:

$$V_k^*(x) = \begin{cases} 1, & x \in K \\ \max_{u \in \mathcal{U}} \int_{\mathcal{X}} V_{k+1}^*(y) Q(dy|x,u), & x \in \bar{\mathcal{X}} \\ 0, & x \in \mathcal{X} \setminus K' \end{cases} \tag{4}$$

Hence, it suffices to compute V_k^* and the optimizing policy on $\bar{\mathcal{X}}$. We show that with an additional assumption on kernel Q, V_k^* is continuous on $\bar{\mathcal{X}}$. The continuity is a desired property for approximating V_k^* on $\bar{\mathcal{X}}$ using basis functions.

Assumption 2 *For every $A \in \mathcal{B}(\mathcal{X})$ the mapping $(x,u) \mapsto Q(A|x,u)$ is continuous.*

Proposition 2 *Under Assumption (2), $V_k^*(x)$ is piecewise continuous on $\mathcal{X}$.*

Proof 2 *From continuity of $(x,u) \mapsto Q(A|x,u)$ we conclude that the mapping $(x,u) \mapsto F(x,u)$ is continuous ([18, Fact 3.9]). From the Maximum Theorem [21], it follows that $F(x, u^*(x))$ and thus each $V_k^*(x)$, is continuous on $\bar{\mathcal{X}}$. The result follows by (4).*

2.2 Linear programming approach

Let $\mathcal{F} := \{f : \mathcal{X} \to \mathbb{R},\ f \text{ is measurable}\}$. For $V \in \mathcal{F}$ define two operators $T_u, T : \mathcal{F} \to \mathcal{F}$

$$\mathcal{T}_u[V](x) = \int_{\mathcal{X}} V(y) Q(dy|x,u), \tag{5}$$

$$\mathcal{T}[V](x) = \max_{u \in \mathcal{U}} \mathcal{T}_u[V](x). \tag{6}$$

Let ν be a non-negative measure supported on $\bar{\mathcal{X}}$, referred to as state-relevance measure.

Theorem 1 *Suppose Assumption 1 holds. For $k \in \{0, \dots, T-1\}$, let V_{k+1}^* be the value function at step $k+1$ defined in* (2)*. Consider the infinite dimensional linear program:*

$$\inf_{V(\cdot)\in\mathcal{F}} \quad \int_{\bar{\mathcal{X}}} V(x)\nu(\mathrm{dx}) \tag{Inf-LP}$$

$$\text{subject to} \quad V(x) \geq \mathcal{T}_u[V_{k+1}^*](x), \quad \forall (x,u) \in \bar{\mathcal{X}} \times \mathcal{U}. \tag{7}$$

(a) Any feasible solution of (Inf-LP) *is an upper bound on the optimal reach-avoid value function V_k^*; (b) V_k^* is a solution to this optimization problem and any other solution to* (Inf-LP) *is equal to V_k^*, ν-almost everywhere on $\bar{\mathcal{X}}$.*

First, note that the decision variable above lives in $\mathcal{F}$, an infinite dimensional space. The objectives and constraints are linear in the decision variable. There are infinitely many constraints since $\bar{\mathcal{X}}$ and $\mathcal{U}$ are continuous spaces. This class of problems is referred to in literature as an infinite dimensional linear program [22, 23].

Proof 3 *Let $J^* \in \mathbb{R}$ denote the optimal value of the objective function in* (Inf-LP)*. From Proposition 1, $V_k^* \in \mathcal{F}$ and is equal to the supremum over $u \in \mathcal{U}$ of the right hand side of the constraint* (7)*. Hence, for any feasible $V \in \mathcal{F}$, we have $V(x) \geq V_k^*(x)$ for all $x \in \bar{\mathcal{X}}$ and part (a) is shown. By non-negativity of ν it follows that for any feasible V, $\int_{\bar{\mathcal{X}}} V(x)\nu(\mathrm{dx}) \geq \int_{\bar{\mathcal{X}}} V_k^*(x)\nu(\mathrm{dx})$, which implies $J^* \geq \int_{\bar{\mathcal{X}}} V_k^*(x)\nu(\mathrm{dx})$. On the other hand, $J^* \leq \int_{\bar{\mathcal{X}}} V_k^*(x)\nu(\mathrm{dx})$ since it is the least cost among the set of feasible functions. Hence, $J^* = \int_{\bar{\mathcal{X}}} V_k^*(x)\nu(\mathrm{dx})$ and V_k^* is an optimal solution. To show that any other solution to* (Inf-LP) *is equal to V_k^* ν-almost everywhere on $\bar{\mathcal{X}}$, assume there exists a function V^*, optimal for* (Inf-LP) *that is strictly greater than V_k^* on a set $A_m \in \mathcal{B}(\mathcal{X})$ of non-zero ν-measure. Since V^* and V_k^* are both optimal, we have that $\int_{\bar{\mathcal{X}}} V^*(x)\nu(\mathrm{dx}) = \int_{\bar{\mathcal{X}}} V_k^*(x)\nu(\mathrm{dx}) = J^*$. We can then reduce V^* to the value of V_k^* on A_m, while ensuring feasibility of V^*. This reduces the value of $\int_{\bar{\mathcal{X}}} V^*(x)\nu(\mathrm{dx})$ below J^*, contradicting that V^* is optimal and part (b) is shown.*

As shown in Theorem 1, the sequence of value functions of the stochastic reach-avoid problem derived in (2) are equivalently characterized as solutions of a sequence of infinite dimensional linear programs. Thus, instead of the classical space gridding approaches to solve (2), we focus on approximating V_k^* by approximating the solutions to (Inf-LP).

3. Approximation with a finite linear program

An infinite dimensional LP is in general NP-hard [22, 23]. We approximate the solution to (Inf-LP) by deriving a finite LP through two steps. First, we restrict the decision space to an finite dimensional subspace $\mathcal{F}^M \subset \mathcal{F}$. Second, we replace the infinite constraints in (7) with a sufficiently large finite number of randomly sampled constraints.

3.1 Restriction to a finite dimensional function class

Let $\mathcal{F}^M$ be a finite dimensional subspace of $\mathcal{F}$ spanned by M basis elements denoted by $\{\phi_i\}_{i=1}^M$. For a fixed function $f \in \mathcal{F}$, consider the following semi-infinite linear program defined over functions $\sum_{i=1}^M w_i\phi_i(x) \in \mathcal{F}^M$ with decision variable $w \in \mathbb{R}^M$:

$$\min_{w_1,\dots,w_M} \quad \sum_{i=1}^M w_i \int_{\bar{\mathcal{X}}} \phi_i(x)\nu(\mathrm{dx}) \qquad \text{(Semi-LP)}$$

$$\text{subject to} \quad \sum_{i=1}^M w_i\phi_i(x) \geq \mathcal{T}_u[f](x),\ \forall (x,u) \in \bar{\mathcal{X}} \times \mathcal{U}\,. \qquad (8)$$

The above linear program has finitely many decision variables and infinitely many constraints. It is referred to as a semi-infinite linear program.

We assume that problem (Semi-LP) is feasible. Note that for a bounded f, this can always be guaranteed by including $\phi(x) = 1$ in the basis functions. Consider the following semi-norm on $\mathcal{F}$ induced by the state-relevance measure ν, $\|V\|_{1,\nu} := \int_{\bar{\mathcal{X}}} |V(x)|\nu(\mathrm{dx})$. In the infinite dimensional linear program (Inf-LP) the choice of ν does not affect the optimal solution, as seen in Theorem (1). For finite dimensional approximations, as will be shown in the next Lemma, ν influences approximation accuracy in different regions of $\bar{\mathcal{X}}$.

Let $\hat{V}_f = \sum_{i=1}^M \hat{w}_i\phi_i$ be a solution to (Semi-LP) and $V_f^* \in \mathcal{F}$ be a solution to the same problem but in the original infinite dimensional decision space $\mathcal{F}$.

Lemma 1 *$\hat{V}_f$ achieves the minimum of $\left\|V - V_f^*\right\|_{1,\nu}$, over the set $\{V \in \mathcal{F}^M,\ V \geq V_f^*\}$.*

Proof 4 *It follows from the proof of Theorem (1) that $V_f^* = \sup_u \mathcal{T}_u[f]$, ν-almost everywhere. Now, a function $\hat{V} \in \mathcal{F}^M$ is an upper bound on $V_f^* = \sup_u \mathcal{T}_u[f]$ if and only if it satisfies constraint (8). To show that $\hat{V}_f$ minimizes the ν-norm distance to V_f^*, notice that for any $V(x) = \sum_{i=1}^M w_i\phi_i(x)$ satisfying (8) we have that*

$$\|V - V_f^*\|_{1,\nu} = \int_{\bar{\mathcal{X}}} |V(x) - V_f^*(x)|\nu(\mathrm{dx}) = \int_{\bar{\mathcal{X}}} V(x)\nu(\mathrm{dx}) - \int_{\bar{\mathcal{X}}} V_f^*(x)\nu(\mathrm{dx}),$$

where the second equality is due to the fact that V is an upper bound of V_f^*. Since V_f^* is a fixed constant in the norm optimization of the lemma above, the result follows.

The semi-infinite problem (Semi-LP) can be used to recursively approximate V_k^* using a weighted sum of basis functions. The next proposition formalizes this result.

Proposition 3 *For every $k \in \{0,\dots,T-1\}$, let $\mathcal{F}^{M_k}$ denote the span of a fixed set of M_k basis elements $\{\phi_i^k\}_{i=1}^{M_k}$. Start with the known value function V_T^* and recursively construct $\hat{V}_k(x) = \sum_{i=1}^{M_k} \hat{w}_i^k\phi_i^k(x)$ where $\hat{w}_i^k$ is the solution to* (Semi-LP) *obtained by substituting $f = \hat{V}_{k+1}$ in* (Semi-LP)*. Then, (a) Each $\hat{V}_k$ is also a solution to the problem:*

$$\min_{V(\cdot)\in\mathcal{F}^{M_k}} \quad \left\|V - V_k^*\right\|_{1,\nu} \qquad (9)$$

$$\text{subject to} \quad V(x) \geq \mathcal{T}_u[\hat{V}_{k+1}](x), \quad \forall (x,u) \in \bar{\mathcal{X}} \times \mathcal{U}. \qquad (10)$$

(b) $\hat{V}_k(x) \geq V_k^*(x)$ for all $x \in \bar{\mathcal{X}}$ and $k = 0,\dots,T-1$.

Proof 5 *By induction. Note that at step $T-1$ the results above hold as a direct consequence of Lemma* (1)*. Now, suppose at time step k, $\hat{V}_k(x) \geq V_k^*(x)$. From monotinicity of the operator $\mathcal{T}_u$ [16], it follows that $\mathcal{T}_u[\hat{V}_k](x) \geq \mathcal{T}_u[V_k^*](x)$. By constraint* (10)*, it follows that $\hat{V}_{k-1}(x) \geq \mathcal{T}_u[\hat{V}_k](x) \geq \mathcal{T}_u[V_k^*](x) = V_{k-1}^*(x)$, where the last equality is by* (4)*. So, part (b) is proven. The fact that $\hat{V}_{k-1}$ solving* (Semi-LP) *also minimizes $\left\|V - V_k^*\right\|_{1,\nu}$ follows from the same argument as in Lemma* (1) *and thus, part (a) is proven.*

The above proposition shows that by restricting the decision space of the infinite dimensional linear program, we obtain an upper bound to the reach-avoid value functions V_k^*, at every step k, which is also the least upper bound in the space spanned by the basis functions subject to constraint (10).

3.2 Restriction to a finite number of constraints

A semi-infinite linear program, such as (Semi-LP) is in general NP-hard [24, 25, 26] due to existence of infinitely many constraints, one for each state-action pair $(x,u) \in \bar{\mathcal{X}} \times \mathcal{U}$. One way to approximate the solution is to select a finite set of points from $\bar{\mathcal{X}} \times \mathcal{U}$ to impose the constraints on. One can then use generalization results from sampled convex programs [27, 28] to quantify the near-feasibility of the solution obtained from constraint sampling.

Let $S := \{(x^i, u^i)\}_{i=1}^N$ denote a set of $N \in \mathbb{N}$ elements in $\bar{\mathcal{X}} \times \mathcal{U}$. For a fixed function $f \in \mathcal{F}$, consider the following finite LP defined over functions $\sum_{i=1}^M w_i \phi_i(x) \in \mathcal{F}^M$:

$$\begin{aligned} \min_{w_1,\dots,w_M} \quad & \sum_{i=1}^M w_i \int_{\bar{\mathcal{X}}} \phi_i(x)\nu(\mathrm{d}x) \\ \text{subject to} \quad & \sum_{i=1}^M w_i \phi_i(x) \geq \mathcal{T}_u[f](x),\ \forall (x,u) \in S \end{aligned} \tag{Fin-LP}$$

Since the objective and constraints are linear in the decision variable, the sampling theorem from [28] applies to obtain the following probabilistic feasibility guarantee.

Lemma 2 *Assume that for any $S \subset \bar{\mathcal{X}} \times U$, the feasible region of* (Fin-LP) *is non-empty and the optimizer is unique. Choose the violation and confidence levels $\varepsilon, \beta \in (0,1)$. Construct a set of samples S by drawing N independent points from $\bar{\mathcal{X}} \times \mathcal{U}$ identically distributed according to a probability measure on $\bar{\mathcal{X}} \times \mathcal{U}$ denoted by $\mathbb{P}_{\bar{\mathcal{X}}\times\mathcal{U}}$. Choose N such that*

$$N \geq \frac{2}{\varepsilon}\left(M + \ln\left(\frac{1}{\beta}\right)\right).$$

Let $\tilde{w}^S$ be the sample dependent optimizer in (Fin-LP)*, and $\tilde{V}^S(x) = \sum_{i=1}^M \tilde{w}_i^S \phi_i(x)$. Then,*

$$\mathbb{P}_{\bar{\mathcal{X}}\times\mathcal{U}}(\tilde{V}^S(x) < \mathcal{T}_u[f](x)) \leq \varepsilon \tag{11}$$

with confidence $1-\beta$, with respect to the product measure $(\mathbb{P}_{\bar{\mathcal{X}}\times\mathcal{U}})^N$.

The probabilistic expression in (11) is referred to as violation of $\tilde{V}^S$ [27, 28]. Note that $\tilde{V}^S$ is a function of the N sample realizations. As such, it can only be bounded to an ϵ-level with a confidence with respect to the product measure $(\mathbb{P}_{\bar{\mathcal{X}}\times\mathcal{U}})^N$.

We can recursively construct $\tilde{V}_k = \sum_{i=1}^{M_k} \tilde{w}_i^k \phi_i^k$ by solving (Fin-LP) using $f = \tilde{V}_{k+1}$ and a number $N_k(\varepsilon_k, \beta_k, M_k)$, of samples. It follows that with probability greater than $1-\beta_k$, the violation of $\tilde{V}_k$ is at most ε_k. Consequently, the approximation functions $\tilde{V}_k$ are probabilistic upper bounds on the value functions V_k^*, in contrast to the guaranteed upper bounds provided in Proposition (3).

To evaluate the accuracy of $\tilde{V}_k$, ideally, we would like to find bounds on $\|\tilde{V}_k - V_k^*\|_{1,\nu}$ as a function of $\|\hat{V}_k - V_k^*\|_{1,\nu}$, where $\hat{V}_k$ is computed according to Proposition (3) and then determine the number of basis functions required to bound $\|\hat{V}_k - V_k^*\|_{1,\nu}$ to a given accuracy. As for the first problem, unfortunately, for a given accuracy in the objective function of a sampled convex program, the number of samples grows exponentially in the number of decision variables [29]. This is reminiscent of the Curse of Dimensionality. As for the second problem, bounding $\|\hat{V}_k - V_k^*\|$ is dependent on the basis function choice. We do not elaborate on these topics further and refer the interested readers to [30] for similar issues for MDPs with additive stage average cost. Our focus in the remainder of the paper will be to evaluate the computational tractability and accuracy of (Fin-LP) in estimating reach-avoid value functions through case studies for a general class of MDPs.

4. Radial basis functions for MDPs with Gaussian mixture kernels

For a general class of MDPs modeled by Gaussian mixture kernels [31] we propose using Gaussian radial basis functions (GRBFs) for approximating the reach-avoid value functions. Through this choice, the constraint in (Fin-LP) involving the integration $\mathcal{T}_u[f]$ can be found in closed form. Moreover, it is known that radial basis functions are a sufficiently rich function class to approximate continuous functions [32, 33, 34, 35].

4.1 Basis function choice

To apply GRBFs in the reach-avoid framework, we consider the following problem data:

1. The kernel Q is a Gaussian mixture kernel $\sum_{j=1}^{J} \alpha_j \mathcal{N}(\mu_j, \Sigma_j)$ with diagonal covariance matrices Σ_j, means μ_j and weights α_j such that $\sum_{j=1}^{J} \alpha_j = 1$ for a finite $J \in \mathbb{N}_+$.
2. The target and safe sets K and K' are unions of disjoint hyper-rectangle sets, i.e. $K = \bigcup_{p=1}^{P} K_p = \bigcup_{p=1}^{P} \left(\prod_{l=1}^{n} [a_l^p, b_l^p]\right)$ and $K' = \bigcup_{m=1}^{M} K'_m = \bigcup_{m=1}^{M} \left(\prod_{l=1}^{n} [c_l^m, d_l^m]\right)$ for finite $P, M \in \mathbb{N}_+$ with $n = \dim(\mathcal{X})$ and $a^p, b^p, c^m, d^m \in \mathbb{R}^n$, $\forall p, m$.

The above restrictions apply to a large class of MDPs. For example, the kernel of general non-linear systems subject to additive Gaussian mixture noise is a Gaussian mixture kernel. Moreover, in several problems, the state and input constraints are decoupled in different dimensions resulting in disjoint hyper-rectangles as constraint sets.

For each time step k, let $\mathcal{F}^{M_k}$ denote the span of a set of GRBFs $\{\phi_i^k\}_{i=1}^{M_k} : \mathbb{R}^n \to \mathbb{R}$:

$$\phi_i^k(x) = \prod_{l=1}^{n} \frac{1}{\sqrt{2\pi s_{i,l}^k}} \exp\left(-\frac{1}{2}\frac{(x_l - c_{i,l}^k)^2}{s_{i,l}^k}\right), \tag{12}$$

where $\{c_{i,l}^k\}_{l=1}^n \in \mathbb{R}, \{s_{i,l}^k\}_{l=1}^n \in \mathbb{R}_+$ are the centers and the variances, respectively, of the GRBF. The class of GRBFs is closed with respect to multiplication [32, Section 2]. In particular, let $f^1 = \sum_{i=1}^{M_k} w_i^1 \phi_i^k$, $f^2 = \sum_{j=1}^{M_k} w_j^2 \phi_j^k$. Then, $f^1 f^2 = \sum_{i=1}^{M_k} \sum_{j=1}^{M_k} w_i^1 w_j^2 \tilde{\phi}_{ij}^k$, where the centers and variances of the bases $\tilde{\phi}_{ij}^k$ are explicit functions of those of ϕ_i^k, ϕ_j^k.

Integrating the proposed GRBFs over a union of hyper-rectangles decomposes into one dimensional integrals of Gaussian functions. In particular, let $\tilde{V}_k(x) = \sum_{i=1}^{M_k} \tilde{w}_i^k \phi_i^k(x)$ denote the approximate value function at time k and $A = \bigcup_{d=1}^{D} \{[a_1^d, b_1^d] \times \cdots \times [a_n^d, b_n^d]\}$, a finite union of hyper-rectangles. The integral of $\tilde{V}_k$ over A after some algebra reduces to

$$\int_A \tilde{V}_k(x)\nu(\mathrm{d}x) = \sum_{d=1}^{D} \sum_{i=1}^{M_k} \tilde{w}_i^k \prod_{l=1}^{n} \frac{1}{2} \operatorname{erf}\left(\frac{b_l^d - c_{i,l}^k}{\sqrt{2 s_{i,l}^k}}\right) - \frac{1}{2} \operatorname{erf}\left(\frac{a_l^d - c_{i,l}^k}{\sqrt{2 s_{i,l}^k}}\right), \tag{13}$$

where ν is assumed to be uniform product measure on each dimension d and erf denotes the error function defined as $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2)\mathrm{d}t$.

Due to the decomposition of the reach-avoid value functions on the sets $K = \cup_{p=1}^{P} K_p$ and $\bar{\mathcal{X}} = (K' = \cup_{m=1}^{M} K'_m) \setminus K$ as stated in (4), $\mathcal{T}_u[\tilde{V}_k]$ in (5) is equivalent to

$$\int_{\mathcal{X}} \tilde{V}_k(y) Q(dy|x,u) = \sum_{m=1}^{M} \sum_{i=1}^{M_k} \tilde{w}_i^k \int_{K'_m} \phi_i^k(y) Q(dy|x,u) + \sum_{p=1}^{P} \int_{K_p} Q(dy|x,u). \tag{14}$$

Since a Gaussian mixture kernel Q can be written as a GRBF, every term inside the integral above is a product of GRBFs. Hence, it is a GRBF with known centers and variances. The integrals over K_p and K'_m can thus be computed using (13) at a sampled point (x^s, u^s).

4.2 Recursive value function and policy approximation

We summarize the method to approximate the reach-avoid value function in Algorithm 1. The design choices include the number of basis functions, their centers and variances, the sample violation and confidence bounds in Lemma 2 and the state-relevance weights. The number of basis functions is problem dependent and in our case studies, we use trial and error to fix this number. We choose the centers of the GRBFs by sampling them from a uniform probability measure supported on $\bar{\mathcal{X}}$. We sample the variances from a uniform measure supported on a bounded set that depends on problem data. Note that the method is still applicable if centers and variances are not sampled but set in another way, for example using neural network training or trial and error. Typically, ε and β are chosen to be close to 0 to enhance the feasibility guarantees of Lemma 2 at the expense of more constraints in (Fin-LP). Furthermore, we choose the state-relevance measure ν as a uniform product measure on the space $\bar{\mathcal{X}}$ to use the analytic integration in (13). This corresponds to equal weighting on potential errors on different state-space regions.

Given the approximate value functions, we compute the so-called greedy control policy:

$$\tilde{\mu}_k(x) = \arg\max_{u \in \mathcal{U}} \int_{\mathcal{X}} \tilde{V}_{k+1}(y) Q(dy|x,u). \tag{15}$$

The optimization problem in (15) is non-convex. However, the cost function is smooth with respect to u for a fixed $x \in \bar{\mathcal{X}}$, the gradient and Hessian information can be analytically obtained using the erf function and the decision space $\mathcal{U}$ is typically low dimensional (in most mechanical systems for example, $\dim \mathcal{U} \leq \dim \mathcal{X}$). Thus, a locally optimal solution can be obtained efficiently using off-the-shelf optimization solvers.

Algorithm 1 linear programming based reach-avoid value function approximation

Input Data:

- State and control spaces $\bar{\mathcal{X}} \times \mathcal{U}$, reach-avoid time horizon T.
- Target and safe sets K and K', written as unions of disjoint hyper-rectangles.
- Centers and variances of the MDP Gaussian mixture kernel Q.

Design parameters:

- Number of basis functions $\{M_k\}_{k=0}^{T-1}$.
- Violation and confidence levels $\{\varepsilon_i\}_{i=0}^{T-1}$, $\{1-\beta_i\}_{i=0}^{T-1}$, probability measure $\mathbb{P}_{\bar{\mathcal{X}}\times\mathcal{U}}$.
- Probability measure of centers and variances for the basis functions $\{\phi_i^k\}_{i=1}^{M_k}$.
- State-relevance measure ν decomposed as a product measure on the state space.

Initialize $\tilde{V}_T(x) \leftarrow \mathbb{1}_K(x)$.
for $k = T-1$ **to** $k = 0$ **do**
  Construct $\mathcal{F}^{M_k}$ by sampling M_k centers $\{c_i\}_{i=1}^{M_k}$ and variances $\{s_i\}_{i=1}^{M_k}$ according to the chosen probability measures.
  Sample $N(\varepsilon_k, \beta_k, M_k)$ pairs (x^s, u^s) from $\bar{\mathcal{X}} \times \mathcal{U}$ using the measure $\mathbb{P}_{\bar{\mathcal{X}}\times\mathcal{U}}$.
  for all (x^s, u^s) **do**
    Evaluate $\mathcal{T}_{u^s}[\tilde{V}_{k+1}](x^s)$ using (14).
  end for
  Solve the finite LP in (Fin-LP) to obtain $\tilde{w}^k = (\tilde{w}_1^k, \ldots, \tilde{w}_{M_k}^k)$.
  Set the approximated value function on $\bar{\mathcal{X}}$ to $\tilde{V}_k(x) = \sum_{i=1}^{M_k} \tilde{w}_i^k \phi_i^k(x)$.
end for

5. Numerical case studies

We develop and solve a series of benchmark problems and evaluate our approximate solutions in two ways. First, we compute the closed-loop empirical reach-avoid policy by applying the approximated control input obtained from (15). Second, we use scalable alternative approaches to approximate the benchmark reach-avoid problems. To this end, we consider three reach-avoid problems that differ in structure and complexity. The first two examples are academic and illustrate the scalability and accuracy of the approach. The last example is a practical problem, where the approach was also implemented on a miniature race-car testbed. Throughout, we refer to our approach as the ADP approach. All computations were carried out on an Intel Core i7 Q820 CPU clocked at 1.73 GHz with 16GB of RAM memory, using IBM's CPLEX optimization toolkit in its default settings.

5.1 Example 1

We consider linear systems with additive Gaussian noise, $x_{k+1} = x_k + u_k + \omega_k$, where $x_k \in \mathcal{X} = \mathbb{R}^n$, $u_k \in \mathcal{U} = [-0.1, 0.1]^n$ and ω_k is distributed as a Gaussian random variable $\omega_k \sim \mathcal{N}(\mathbf{0}_{n\times 1}, \Sigma)$ with diagonal covariance matrix. We consider a target set $K = [-0.1, 0.1]^n$ centered at the origin and a safe set $K' = [-1, 1]^n$ (see Figure 1 for a 2D illustration).

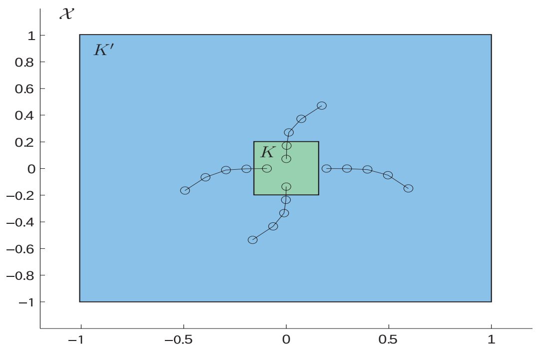


Figure 1: 2D depiction of safe and target sets and sample trajectories.

$\dim(\mathcal{X} \times \mathcal{U})$	4D	6D	8D
M_k	100	500	1000
N_k	4184	20184	40184
ε_k	0.05	0.05	0.05
$1-\beta_k$	0.99	0.99	0.99
$\\|\tilde{V}_0 - V_{\text{ADP}}\\|$	0.0692	0.104	0.224
Construction (sec)	4	85	450
LP solution (sec)	2	50	520
Memory (MB)	3.2	80	320

Table 1: Parameters and properties of the value function approximation scheme.

The objective is to reach the target set while staying in the safe set over a horizon of $T = 5$ steps. We approximated the value function using Algorithm 1 for a range of system dimensions $\dim(\mathcal{X} \times \mathcal{U}) = 4, 6, 8$, to analyze scalability and accuracy of the LP-based reach-avoid solution in a benchmark problem that scales up in a straightforward way.

The transition kernel of the considered linear system is Gaussian $x_{k+1} \sim \mathcal{N}(x_k + u_k, \Sigma)$. The sets K and K' are hyper-rectangles. Thus, the GRBF framework applies. We chose 100, 500 and 1000 GRBF elements for the reach-avoid problems of $\dim(\mathcal{X} \times \mathcal{U}) = 4, 6, 8$, respectively (Table 1). We used uniform measures supported on $\bar{\mathcal{X}}$ and $[0.02, 0.095]^n$ to sample the GRBFs' centers and variances, respectively. The violation and confidence levels for every $k \in \{0, \dots, 4\}$ were set to $\varepsilon_k = 0.05$, $1-\beta_k = 0.99$ and the measure $\mathbb{P}_{\bar{\mathcal{X}} \times \mathcal{U}}$ required to generate samples from $\bar{\mathcal{X}} \times \mathcal{U}$ is chosen to be uniform. Since there is no reason to favor some states more than others, we also chose ν as a uniform measure, supported on $\bar{\mathcal{X}}$. Following Algorithm 1 we obtain a sequence of approximate value functions $\{\tilde{V}_k\}_{k=0}^4$.

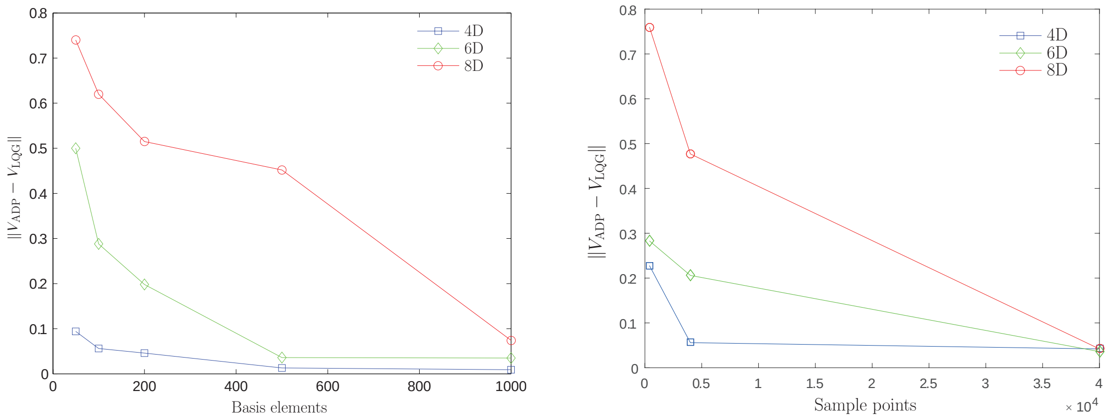
To evaluate the performance of the approximation, we sampled 100 initial conditions x_0, uniformly from $\bar{\mathcal{X}}$. For each initial condition we generated 100 noise trajectories $\{\omega_k\}_{k=0}^{T-1}$. We computed the policy along the resulting state trajectory using (15). We then counted the number of trajectories that successfully completed the reach-avoid objective, i.e. reach K without leaving K' in T steps. In Table 1 we denote by $\|\tilde{V}_0 - V_{\text{ADP}}\|$ the mean absolute difference between the empirical success denoted by V_{ADP}, and the predicted performance $\tilde{V}_0$, evaluated over the considered initial conditions. The memory and computation times reported correspond to constructing and solving each LP.

Since the system is linear, the noise is Gaussian and the target and safe sets are symmetric and centered around the origin, we assume that a properly tuned LQG controller will perform close to optimal for the reach-avoid objective. Thus, we use the known closed-form LQG solution as a heuristic method to compare the proposed approach.

The LQG problem for a linear stochastic system $x_{k+1} = Ax_k + Bu_k + \omega_k$, as the one considered above, is defined by an expected value quadratic cost function:

$$\min_{\{u_k\}_{k=0}^{T-1}} \mathbb{E}_{x_0}^{\mu}\left(\sum_{k=0}^{T-1} x_k^\top Q x_k + u_k^\top R u_k\right) + x_T^\top Q x_T.$$

Above, $Q \in \mathcal{S}_+^n$ and $R \in \mathcal{S}_{++}^m$, where $\mathcal{S}_+^n$ and $\mathcal{S}_{++}^m$ denote the cones of $n \times n$ positive semidefinite and $m \times m$ positive definite matrices, respectively. Q and R were chosen to



(a) Mean absolute difference between the empirical reach-avoid probabilities achieved by the ADP (V_{ADP}) and LQG (V_{LQG}) policies as a function of number of basis functions.

(b) Mean absolute difference between the empirical reach-avoid probabilities achieved by the ADP (V_{ADP}) and LQG (V_{LQG}) policies as a function of number of samples.

Figure 2: Example 1 - performance of the algorithm as a function of parameters.

correspond to the largest ellipsoids inscribed in K and $\mathcal{U}$, respectively. Through this choice the level sets of the LQG cost function proportionally correspond to the size of the target and control constraint sets. Intuitively, the penalization of states through the quadratic cost Q drives the state to the origin. The penalization of the input does not guarantee feasibility of the input constraints. Therefore, we project the LQG control on the feasible set $\mathcal{U}$. Using the same initial conditions and noise trajectories as those used with the ADP controller above, we simulated the performance of the LQG controller. We counted the number of trajectories that reach K without leaving K' over the horizon of $T = 5$ steps.

Figure 2a shows the mean over the initial conditions of the absolute difference between V_{LQG} and V_{ADP} as a function of number of basis functions. We observe a trend of increasing accuracy with increasing number of basis functions. Figure 2b shows the same metric but as a function of the total number of sample pairs from $\mathcal{X} \times \mathcal{U}$ for a fixed number of basis functions. Changing the number of samples N, affects the violation level ε_k (assuming constant β_k) and the approximation quality seems to improve with increasing N. In Table 2, we observe a trade-off between accuracy and computational time for the 6D problem varying the number of samples; the result is analogous in the 4D and 8D problems.

N	$\|V_{\mathrm{ADP}} - V_{\mathrm{LQG}}\|$	Construction (sec)	LP solution (sec)	Memory (MB)
400	0.283	2.20	3.57	1.60
4000	0.206	17.0	97.0	16.0
40000	0.036	170	162	160

Table 2: Accuracy and computation time as a function of number of sampled points in $\dim(\mathcal{X} \times \mathcal{U}) = 6$, with $M_k = 500$ and $1 - \beta_k = 0.99$.

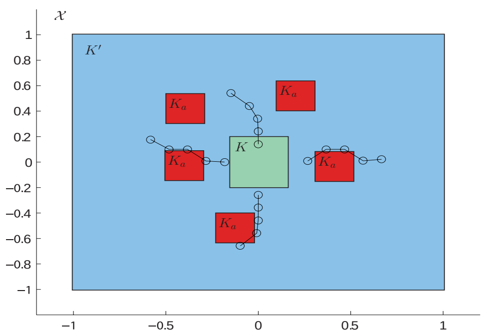


Figure 3: Example 2 - 2D depiction of obstacles and sample trajectories.

$\dim(\mathcal{X} \times \mathcal{U})$	4D	6D	8D
M_k	100	500	1000
N_k	4184	20184	40184
ε_k	0.05	0.05	0.05
$1-\beta_k$	0.99	0.99	0.99
$\lVert \tilde{V}_0 - V_{\text{ADP}} \rVert$	0.095	0.118	0.191
Construction (sec)	4.20	130	671
LP solution (sec)	3.2	80	700
Memory (MB)	3.20	80.0	320

Table 3: Parameters and properties of the value function approximation scheme.

5.2 Example 2

We consider the same linear dynamical system $x_{k+1} = x_k + u_k + \omega_k$, with target set K as defined in Section 5.1. In addition, in this example, the avoid set includes obstacles placed randomly within the state space as depicted in Figure 3. The safe set is $(K' \setminus \bigcup_{j=1}^{5} K_\alpha^j)$, where K' was defined in the previous example, and each K_α^j denotes one of the hyper-rectangular obstacle sets. The reach-avoid time horizon is $T = 7$. We use Algorithm 1 to approximate the optimal reach-avoid value function and compute the optimal policy.

We chose the same basis function numbers, basis parameters, sampling and state-relevance measures as well as violation and confidence levels as in Section 5.1, shown in Table 3. We simulated the performance of the ADP controller starting from 100 different initial conditions, selected such that at least one obstacle blocks the direct path to the origin. For every initial condition we sample d100 different noise trajectory realizations and applied the corresponding control policies computed through (15). We then computed the empirical ADP reach-avoid success probability (denoted by V_{ADP}) by counting the total number of trajectories that reach K while avoiding reaching the obstacles or leaving K'.

The problem of reaching a target set without passing through any obstacles is an instance of a path planning problem and has been studied thoroughly for deterministic systems (see for example, [36, 37, 38]). For a benchmark comparison we use the approach of [37] and formulate the reach-avoid problem for the noise-free system as a constrained mixed logic dynamical system (MLD) [39]. This problem can in turn be recast as a mixed integer quadratic program (MiQP) and solved to optimality using standard branch and bound techniques. To account for noise in the dynamics ω_k, we used a heuristic approach as follows. We truncated the density function of the random variables ω_k at 95% of their total mass and enlarged each obstacle set K_α by the maximum value of the truncated ω_k in each dimension. This resembles the robust (worst-case) approach to control design.

Starting from the same initial conditions as in the ADP approach, we simulated the performance of the MiQP-based control policy on the 100 trajectory realizations used in the ADP controller. We implemented the policy in receding horizon by measuring the state at each horizon step. The empirical success probability of trajectories that reach K while staying safe is denoted by V_{MiQP}. The mean difference $\lVert V_{\text{ADP}} - V_{\text{MiQP}} \rVert$ is presented in Table 4 and is computed by averaging the corresponding empirical reach-avoid success probabilities over the initial conditions. As seen in this table, as the number of basis

M_k	$\|V_{\text{ADP}} - V_{\text{MiQP}}\|$	Construction (sec)	LP solution (sec)	Memory (MB)
50	0.214	1.67	0.18	0.784
100	0.168	5.59	2.66	3.20
200	0.084	22.0	4.30	12.8
500	0.070	130	80.0	80.0
1000	0.045	507	1210	320

Table 4: Example 2 - Accuracy and computational requirements for $\dim(\mathcal{X} \times \mathcal{U}) = 6$.

functions increases, $\|V_{\text{ADP}} - V_{\text{MiQP}}\|$ decreases. This can indicate that the reach-avoid value function approximation is increasing in accuracy. Note that for an increasing planning horizon T, the number of binary variables (and hence the computational complexity) in MiQP grows exponentially, whereas in the LP-based reach-avoid approach, the computation effort grows linearly with the horizon.

5.3 Example 3

Consider the problem of driving a race car through a tight corner in the presence of static obstacles, illustrated in Figure 4. As part of the ORCA project of the Automatic Control Lab (see `http://control.ee.ethz.ch/~racing/`), a six state variable nonlinear model with two control inputs has been identified to describe the movement of 1:43 scale race cars. The model derivation is discussed in [40] and is based on a unicycle approximation with parameters identified on the experimental platform of the ORCA project using model cars manufactured by Kyosho. We denote the state space by $\mathcal{X} \subset \mathbb{R}^6$, the control space by $\mathcal{U} \subset \mathbb{R}^2$ and the identified dynamics by a function $f : \mathcal{X} \times \mathcal{U} \mapsto \mathcal{X}$. The first two elements of each state $x \in \mathcal{X}$ correspond to spatial dimensions, the third to orientation, the fourth and fifth to body fixed longitudinal and lateral velocities and the sixth to angular velocity. The two control inputs $u \in \mathcal{U}$ are the throttle duty cycle and the steering angle.

As typically observed in practice, the state predicted by the identified dynamics and the state measurements recorded on the experimental platform are different due to process

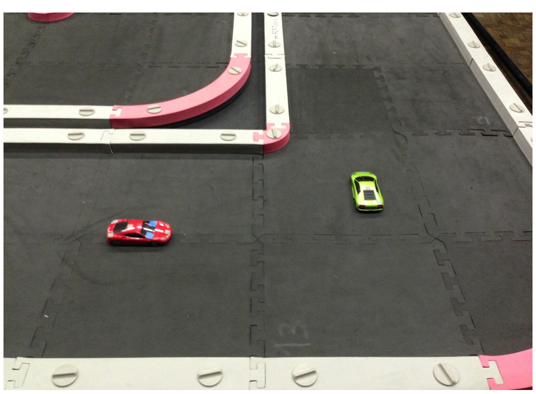


Figure 4: Example 3 - The set up of the Race-car cornering problem.

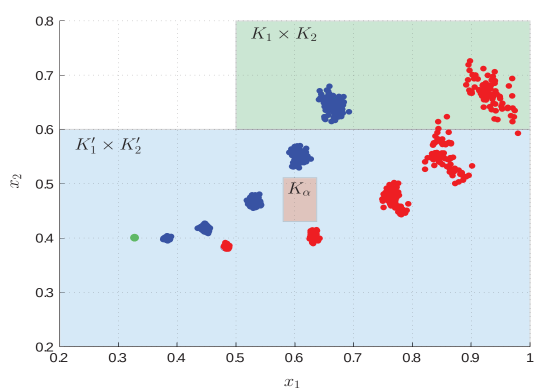


Figure 5: Example 3 - sample trajectories based on reach-avoid computation.

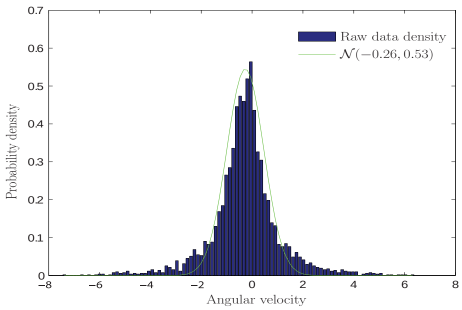


Figure 6: Empirical noise distribution.

Safe region	min	max	variances
K'_1 (m)	0.2	1	$[8 \times 10^{-4}, 1.2 \times 10^{-3}]$
K'_2 (m)	0.2	0.6	$[8 \times 10^{-4}, 1.2 \times 10^{-3}]$
K'_3 (rad)	$-\pi$	π	$[5 \times 10^{-3}, 1.5 \times 10^{-2}]$
K'_4 (m/s)	0.3	3.5	$[5 \times 10^{-3}, 1.5 \times 10^{-2}]$
K'_5 (m/s)	-1.5	1.5	$[5 \times 10^{-3}, 1.5 \times 10^{-2}]$
K'_6 (rad/s)	-8	8	[2.00,4.00]

Table 5: State constraints and basis functions' variances used in ADP approximation.

and measurement noise. Analyzing the deviation between predictions and measurements, we captured the uncertainties in the original model using additive Gaussian noise, $g(x,u) = f(x,u) + \omega, \quad \omega \sim \mathcal{N}(\mu, \Sigma),\ \mu \in \mathbb{R}^6, \Sigma \in \mathcal{S}^6_{++}$. The noise mean μ, and diagonal covariance matrix Σ have been selected such that the probability density function of the Markov decision process describing the uncertain dynamics resembles the empirical data obtained via measurements. As an example, Figure 6 illustrates the fit for the angular velocity where $\mu_6 = -0.26$ and $\Sigma(6,6) = 0.53$. It follows that the kernel of the stochastic process is a GRBF with a single basis function described by the Gaussian distribution $\mathcal{N}(f(x,u) + \mu, \Sigma)$.

We cast the problem of driving the race car through a tight corner without reaching obstacles as a stochastic reach-avoid problem. Despite the highly nonlinear dynamics, the stochastic reach-avoid set-up can readily be applied to this problem.

We consider a horizon of $T = 6$ and a sampling time of 0.08 seconds. The safe region of the spatial dimensions is defined as $(K'_1 \times K'_2) \setminus A$ where $A \subset \mathbb{R}^2$ denotes the obstacle, see Figures 4, 5. The safe set in 6D is thus defined as $K' = ((K'_1 \times K'_2) \setminus A) \times K'_3 \times K'_4 \times K'_5 \times K'_6$ where K'_3, K'_4, K'_5, K'_6 describe the physical limitations of the model car (see Table 5). Similarly, the target set for the spatial dimensions is denoted by $K_1 \times K_2$ and corresponds to the end of the turn as shown in Figure 5. The target set in 6D is then defined as $K = K_1 \times K_2 \times K'_3 \times K'_4 \times K'_5 \times K'_6$, which contains all states $x \in K'$ for which $(x_1, x_2) \in K_1 \times K_2$. The constraint sets are naturally decoupled over the state dimensions.

We used 2000 GRBFs for each approximation step with centers and variances sampled according to uniform measures supported on $\bar{\mathcal{X}}$ and on the hyper-rectangle defined by the product of intervals in the rows of Table 5, respectively. We used a uniform state-relevance measure and a uniform sampling measure to construct each one of the finite linear programs in Algorithm 1. All violation and confidence levels were chosen to be $\varepsilon_k = 0.2$ and $1 - \beta_k = 0.99$ respectively for $k = \{0, \dots, 5\}$. Having fixed all design parameters we implement the steps of Algorithm 1 and compute a sequence of approximate value functions.

To evaluate the quality of the approximations we initialized the car at two different initial conditions $x^1 = (0.33, 0.4, -0.2, 0.5, 0, 0)$ and $x^2 = (0.33, 0.4, -0.2, 2, 0, 0)$. They correspond to entering the corner at low ($x^1_4 = 0.5$ m/s) and high ($x^2_4 = 2$ m/s) longitudinal velocities. The approximate value functions evaluate to $\tilde{V}_0(x^1) = 0.98$, $\tilde{V}_0(x^2) = 1$ and indicate success with high probabilities for both cases. Interestingly, the associated trajectories computed via the greedy policy defined through (15) vary significantly. In the low velocity case, the car avoids the obstacle by driving above it while in the high velocity case, by driving below it (see

Figure 5). Such a behavior is expected since the car can slip if it turns aggressively at high velocities. We also computed empirical reach-avoid probabilities in simulation by sampling 100 noise trajectories from each initial state and implementing the ADP control policy using the associated value function approximation. The sample trajectories are plotted in Figure 5 and the values were found to be $V_{\text{ADP}}(x^1) = 1$ and $V_{\text{ADP}}(x^2) = 0.99$

The controller was tested on the ORCA setup by running 10 experiments from each initial condition. We pre-computed the control inputs at the predicted mean trajectory of the states over the horizon for each experiment. Implementing the feedback policy online would require solving problem (15) within the sampling time of 0.08 seconds. In theory, this computation is possible since the control space is only two dimensional but it requires developing an embedded nonlinear programming solver compatible with the ORCA setup. Here, we have implemented the open loop controller. We note however that if the open loop controller performs accurately, the closed loop computation can only improve the performance by utilizing updated state measurements. As demonstrated by the videos in (youtube:ETHZurichIfA), the car is successfully driving through the corner even when the control inputs are applied in an open loop.

6. Conclusions

We developed a numerical approach to compute the value function of the stochastic reach-avoid problem for Markov decision processes with uncountable state and action spaces. Since the method relies on solving linear programs we were able to tackle reach-avoid problems with larger dimensions than established state space gridding methods. The potential of the approach was analyzed through two benchmark case studies and a trajectory planning problem for a six dimensional nonlinear system with two inputs. To the best of our knowledge, this is the first time that stochastic reach-avoid problems up to eight continuous state and input dimensions have been addressed.

We are currently focusing on the problem of systematically choosing the basis function parameters by exploiting knowledge about the system dynamics. Furthermore, we are developing decomposition methods for the large linear programs that arise in our approximation scheme to allow addressing control of MDPs with higher dimensions. Finally, we are addressing tractable reformulations of the infinite constraints in the semi-infinite linear programs for stochastic reach-avoid problems to avoid sampling-based methods.

Acknowledgments

The authors would like to thank Alexander Liniger from the Automatic Control Laboratory in ETH Zürich for his help in testing the algorithm on the ORCA platform

References

[1] Feinberg, E. A., Shwartz, A., and Altman, E., 2002. *Handbook of Markov decision processes: methods and applications*. Kluwer Academic Publishers Boston, MA.

[2] Puterman, M. L., 1994. *Markov decision processes: Discrete stochastic dynamic programming.* John Wiley & Sons, Inc.

[3] Abate, A., Prandini, M., Lygeros, J., and Sastry, S., 2008. "Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems". *Automatica,* **44**(11), pp. 2724–2734.

[4] Watkins, O., and Lygeros, J., 2003. "Stochastic reachability for discrete time systems: An application to aircraft collision avoidance". In 42nd IEEE Conference on Decision and Control (CDC), 2003. Proceedings., Vol. 5, IEEE, pp. 5314–5319.

[5] Ding*, J., Kamgarpour*, M., Summers, S., Abate, A., Lygeros, J., and Tomlin, C., 2013. "A stochastic games framework for verification and control of discrete time stochastic hybrid systems". *Automatica,* **49**(9), pp. 2665–2674. *: equal contribution.

[6] Kariotoglou, N., Raimondo, D. M., Summers, S. J., and Lygeros, J., 2015. "Multi-agent autonomous surveillance: a framework based on stochastic reachability and hierarchical task allocation". *Journal of dynamic systems, measurement, and control,* **137**(3), p. 031008.

[7] Wood, T., Summers, S. J., and Lygeros, J., 2013. "A stochastic reachability approach to emergency building evacuation". In 52nd IEEE Conference on Decision and Control (CDC), 2013, IEEE, pp. 5722–5727.

[8] Prandini, M., and Hu, J., 2006. "Stochastic reachability: Theory and numerical approximation". *Stochastic hybrid systems, Automation and Control Engineering Series,* **24**, pp. 107–138.

[9] Abate, A., Amin, S., Prandini, M., Lygeros, J., and Sastry, S., 2007. "Computational approaches to reachability analysis of stochastic hybrid systems". In *Hybrid Systems: Computation and Control*. Springer, pp. 4–17.

[10] Kushner, H. J., and Dupuis, P., 2001. *Numerical methods for stochastic control problems in continuous time*, Vol. 24. Springer.

[11] Powell, W. B., 2007. *Approximate Dynamic Programming: Solving the curses of dimensionality*, Vol. 703. John Wiley & Sons.

[12] Bertsekas, D., 2012. *Dynamic programming and optimal control*, Vol. 2. Athena Scientific Belmont, MA.

[13] de Farias, D. P., and Van Roy, B., 2003. "The linear programming approach to approximate dynamic programming". *Operations Research,* **51**(6), pp. 850–865.

[14] Hernández-Lerma, O., and Lasserre, J. B., 1996. *Discrete-time Markov control processes: basic optimality criteria*. Springer New York.

[15] Kamgarpour, M., Summers, S., and Lygeros, J., 2013. "Control Design for Property Specifications on Stochastic Hybrid Systems". In *Hybrid Systems: Computation and Control*, C. Belta and F. Ivancic, eds., Lecture Notes in Computer Science. ACM, pp. 303–312.

[16] Summers, S. J., and Lygeros, J., 2010. "Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem". *Automatica,* **46**(12), pp. 1951–1961.

[17] Summers, S. J., Kamgarpour, M., Tomlin, C., and Lygeros, J., 2013. "Stochastic system controller synthesis for reachability specifications encoded by random sets". *Automatica,* **49**(9), pp. 2906–2910.

[18] Nowak, A., 1985. "Universally measurable strategies in zero-sum stochastic games". *The Annals of Probability*, pp. 269–287.

[19] Brown, L. D., Purves, R., et al., 1973. "Measurable selections of extrema". *The annals of statistics,* **1**(5), pp. 902–912.

[20] Bertsekas, D. P., 1976. "Dynamic programming and stochastic control".

[21] Sundaram, R. K., 1996. *A first course in optimization theory.* Cambridge university press.

[22] Anderson, E. J., and Nash, P., 1987. *Linear programming in infinite-dimensional spaces: theory and applications.* Wiley New York.

[23] Hernández-Lerma, O., and Lasserre, J. B., 1998. "Approximation schemes for infinite linear programs". *SIAM Journal on Optimization,* **8**(4), pp. 973–988.

[24] Bertsimas, D., Brown, D. B., and Caramanis, C., 2011. "Theory and applications of robust optimization". *SIAM review,* **53**(3), pp. 464–501.

[25] Ben-Tal, A., and Nemirovski, A., 2002. "Robust optimization–methodology and applications". *Mathematical Programming,* **92**(3), pp. 453–480.

[26] Hettich, R., and Kortanek, K. O., 1993. "Semi-infinite programming: theory, methods, and applications". *SIAM review,* **35**(3), pp. 380–429.

[27] Calafiore, G. C., and Campi, M. C., 2006. "The scenario approach to robust control design". *IEEE Transactions on Automatic Control,* **51**(5), pp. 742–753.

[28] Campi, M. C., Garatti, S., and Prandini, M., 2009. "The scenario approach for systems and control design". *Annual Reviews in Control,* **33**(2), pp. 149–157.

[29] Mohajerin Esfahani, P., Sutter, T., and Lygeros, J., 2015. "Performance Bounds for the Scenario Approach and an Extension to a Class of Non-convex Programs". *IEEE Transactions on Automatic Control,* **60**(1), pp. 46–58.

[30] Sutter, T., Mohajerin Esfahani, P., and Lygeros, J., 2014. "Approximation of Constrained Average Cost Markov Control Processes". In IEEE Conference on Decision and Control, pp. 6597 – 6602.

[31] Khansari-Zadeh, S. M., and Billard, A., 2011. "Learning stable nonlinear dynamical systems with gaussian mixture models". *IEEE Transactions on Robotics,* **27**(5), pp. 943–957.

[32] Hartman, E. J., Keeler, J. D., and Kowalski, J. M., 1990. "Layered neural networks with Gaussian hidden units as universal approximations". *Neural computation,* **2**(2), pp. 210–215.

[33] Sandberg, I. W., 2001. "Gaussian radial basis functions and inner product spaces". *Circuits, Systems and Signal Processing,* **20**(6), pp. 635–642.

[34] Park, J., and Sandberg, I. W., 1991. "Universal approximation using radial-basis-function networks". *Neural computation,* **3**(2), pp. 246–257.

[35] Cybenko, G., 1989. "Approximation by superpositions of a sigmoidal function". *Mathematics of control, signals and systems,* **2**(4), pp. 303–314.

[36] Borenstein, J., and Koren, Y., 1991. "The vector field histogram-fast obstacle avoidance for mobile robots". *IEEE Transactions on Robotics and Automation,* **7**(3), pp. 278–288.

[37] Richards, A., and How, J. P., 2002. "Aircraft trajectory planning with collision avoidance using mixed integer linear programming". In American Control Conference, 2002. Proceedings of the 2002, Vol. 3, IEEE, pp. 1936–1941.

[38] Van Den Berg, J., Ferguson, D., and Kuffner, J., 2006. "Anytime path planning and replanning in dynamic environments". In Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on, IEEE, pp. 2366–2371.

[39] Bemporad, A., and Morari, M., 1999. "Control of systems integrating logic, dynamics, and constraints". *Automatica,* **35**(3), pp. 407–427.

[40] Liniger, A., Domahidi, A., and Morari, M., 2014. "Optimization-based autonomous racing of 1: 43 scale RC cars". *Optimal Control Applications and Methods.*