

A numerical approach to stochastic reach-avoid problems for Markov Decision Processes

Nikolaos Kariotoglou, Maryam Kamgarpour
Tyler Summers and John Lygeros

Abstract

We consider finite horizon reach-avoid problems for discrete time stochastic systems with additive Gaussian mixture noise. Our goal is to approximate the optimal value function of such problems on dimensions higher than what can be handled via state-space gridding techniques. We achieve this by relaxing the recursive equations of the finite horizon reach-avoid optimal control problem into inequalities, projecting the optimal value function to a finite dimensional basis and sampling the associated infinite set of constraints. We focus on a specific value function parametrization using Gaussian radial basis functions that enables the analytical computation of the one-step reach-avoid reward in the case of hyper-rectangular safe and target sets, achieving significant computational benefits compared to state-space gridding. We analyze the performance of the overall method numerically by approximating simple reach-avoid control problems and comparing the results to benchmark controllers based on well-studied methods. The full power of the method is demonstrated on a nonlinear control problem inspired from constrained path planning for autonomous race cars.

I. INTRODUCTION

A wide range of controlled physical systems can be modeled using the framework of Markov Decision processes (MDPs) [1], [2]. As such, one can formulate optimal control problems where the objective is to compute control policies for the MDP that maximize an expected reward, or minimize an expected cost over a finite or infinite time horizon. In this work, we consider the stochastic reach-avoid problem for MDPs in which one maximizes the probability of reaching a target set while staying in a safe set. This problem has been formally stated in [3] for discrete time stochastic hybrid systems which are reformulated as MDPs. Current state-of-the-art methods express the solution to the stochastic reach-avoid problem as a dynamic programming (DP) recursion and then approximate it on a finite grid of the state and control spaces. State-space gridding techniques are theoretically attractive since they can provide explicit error bounds for the approximation of the value function under general Lipschitz continuity assumptions [4], [5], [6]. In practice, the complexity of gridding based techniques suffers from the infamous Curse of Dimensionality and thus, gridding is 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 limited (to the best of our knowledge to no more than 5). As a result, it is worth investigating alternative approximation techniques to push this limit further.

Several researchers have developed approximate dynamic programming (ADP) techniques for various classes of stochastic control problems [7], [8]. 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 [9] where the authors develop an approximation method based on Linear Programming (LP) for finite state and control spaces. Although the LP approach has been extensively used to approximate standard MDP problems with additive stage costs, it has not yet been considered for stochastic reach-avoid problems. In general, LP approaches to ADP are desirable since several commercially available software packages can handle LP problems with large numbers of variables and constraints.

A preliminary study on using linear programming to solve stochastic reach-avoid problems over uncountable state and control spaces was presented in [10]. Here, we extend the results of [10] in several ways. First, we quantify the relation between the original reach-avoid DP recursion and an infinite dimensional

LP formulated over the space of Borel-measurable functions defined on the MDP state space. We then show that restricting the infinite dimensional decision space to a finite dimensional subspace using a collection of basis functions is equivalent to the projection of the value functions onto the basis function span, intersected with the feasible region of the infinite LP. We prove that recursively projecting the approximate value functions results in computing an upper bound on the optimal ones. In the general case, the sequence of infinitely constrained LP problems constructed by our method is not computationally tractable and we rely on the scenario sampling approach [11] to compute solutions which are feasible in probability.

For the numerical implementation of our method, we propose the use of finite dimensional subspaces spanned by radial basis functions (RBFs). We show that this particular choice of basis allows the analytic computation of integrals for a wide class of safe and target sets and a wide class of MDP transition kernels. Using the RBF structure we present a novel algorithm to approximate the value function of reach-avoid problems which results in solving finite dimensional LP problems. The performance of the developed method is illustrated via numerical examples where we approximate the solution to reach-avoid dynamic recursions that are intractable using state-space gridding. In particular, we formulate two types of simple reach-avoid problems for which approximate solutions can be efficiently computed using existing control methods and confirm that our approximations are accurate by comparing their performance. We also formulate a more complex reach-avoid problem inspired by car racing where constructing a benchmark method is not straightforward but our method is still applicable.

The rest of the paper is organized as follows: In Section II-A we recall the basic details of the stochastic reach-avoid problem for MDPs with uncountable state and control spaces and formulate an infinite dimensional LP for which the reach-avoid value function is an optimal solution. In Section II-B we formulate an infinite dimensional LP equivalent to the reach-avoid dynamic recursion and use basis function approximation and randomized sampling techniques to probabilistically upper bound its solution. In Section IV we restrict the form of basis elements to Gaussian RBFs and analyze some of their approximation and numerical computation properties. We use the proposed methodology in Section V to approximate the value function of high dimensional reach-avoid problems for linear systems. We conclude our analysis with an 8D reach-avoid problem for a nonlinear system, demonstrating the full power of the proposed method. In the last section of the paper we summarize the approach and describe potential future work.

II. STOCHASTIC REACH-AVOID PROBLEM

A. Dynamic programming approach

We consider a discrete-time controlled stochastic process $x_{t+1} \sim Q(dx|x_t, u_t)$, $(x_t, u_t) \in \mathcal{X} \times \mathcal{U}$ with a transition kernel $Q : \mathcal{B}(\mathcal{X}) \times \mathcal{X} \times \mathcal{U} \rightarrow [0, 1]$ where $\mathcal{B}(\mathcal{X})$ 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} falling in a set $A \in \mathcal{B}(\mathcal{X})$. The transition kernel Q is a Borel measurable stochastic kernel, that is, $Q(A|\cdot)$ is a 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) . 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 functions $\mu = \{\mu_0, \dots, \mu_{T-1}\}$ where $\mu_i : \mathcal{X} \rightarrow \mathcal{U}$ for each $i \in \{0, \dots, T-1\}$. 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 $t_K \leq T$ while staying in K' for all $t \leq t_K$. For any initial state x_0 we denote the reach-avoid probability associated with a given μ as: $r_{x_0}^\mu(K, K') = \mathbb{P}_{x_0}^\mu\{\exists j \in [0, T] : x_j \in K \wedge \forall i \in [0, j-1], x_i \in K' \setminus K\}$ and operate under the assumption that $[0, -1] = \emptyset$ which implies that the requirement on i is automatically satisfied when $x_0 \in K$.

In [12], $r_{x_0}^\mu(K, K')$ is shown to be equivalent to the following sum multiplicative cost function:

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

where $\prod_{i=k}^j(\cdot) = 1$ if $k > j$. The function $\mathbb{1}_A(x)$ denotes the indicator function of a set $A \in \mathcal{B}(\mathcal{X})$ with $\mathbb{1}_A(x) = 1$ if $x \in A$ and $\mathbb{1}_A(x) = 0$ otherwise. The sets K and K' can be time dependent or even stochastic [13] but for simplicity we assume 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.

The solution to the reach-avoid problem is given by a dynamic recursion [12]. Define $V_k^* : \mathcal{X} \rightarrow [0, 1]$ for $k = T - 1, \dots, 0$ by :

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

The value of the above recursion at $k = 0$ and for any initial state x_0 is the maximum value of (1) over all admissible policies, i.e. $V_0^*(x_0) = \sup_{\mu} r_{x_0}^\mu(K, K')$. Differentiating from [12] where the authors proved universal measurability of the value functions in (2), we impose a mild additional assumption on the continuity of the transition kernel and show that the value functions are Borel measurable and the supremum at each step is achieved.

Proposition 1. *Assume that for every $x \in \mathcal{X}$, $A \in \mathcal{B}(X)$ the mapping $u \mapsto Q(A|x, u)$ is continuous. Then, for every k the supremum in (2) is achieved and $V_k^* : \mathcal{X} \rightarrow [0, 1]$ is Borel measurable.*

Proof. By induction. First, note that the indicator function $V_T^*(x) = \mathbb{1}_K(x)$ is Borel measurable. Assuming that V_{k+1}^* is Borel measurable we will show that V_k^* is also Borel measurable. Define $F(x, u) = \int_{\mathcal{X}} V_{k+1}^*(y) Q(dy|x, u)$. Due to continuity of the map $u \mapsto Q(A|x, u)$, the mapping $u \mapsto F(x, u)$ is continuous for every x (by Fact 3.9 of [14]). Now, since \mathcal{U} is compact, by Corollary 1 of [15], there exists a Borel measurable function $u_k^*(x)$ that achieves the supremum. Furthermore, by Proposition 7.29 of [16], the mapping $(x, u) \mapsto F(x, u)$ is Borel measurable. It follows that $F(x, u_k^*(x))$ (and hence V_k^*) is Borel measurable as it is the composition of Borel measurable functions. We conclude by induction that at each time step k there exists a Borel measurable function u_k^* achieving the supremum and that V_k^* is Borel measurable for every k . \square

Proposition 1 allows one to compute an optimal feedback policy at each stage k by solving,

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

The dynamic recursion in (2) implies that the functions $V_k^*(x)$ are defined on three disjoint regions of \mathcal{X} , namely

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

and it can be shown [12] that the value of each V_k^* on \mathcal{X} is restricted to $[0, 1]$.

Proposition 2. *Assume that for every $A \in \mathcal{B}(X)$ the mapping $(x, u) \mapsto Q(A|x, u)$ is continuous, then $V_k^*(x)$ is piecewise continuous on \mathcal{X} .*

Proof. From continuity of $(x, u) \mapsto Q(A|x, u)$ we conclude that the mapping $(x, u) \mapsto F(x, u)$ is continuous (Fact 3.9 in [14]). From the Maximum Theorem [17], it follows that $F(x, u_k^*(x))$ and thus each $V_k^*(x)$, is continuous on $\bar{\mathcal{X}}$. By construction, each V_k^* is then piecewise continuous on \mathcal{X} . \square

The only established way to approximate the solution of (2) is by gridding $\mathcal{X} \times \mathcal{U}$ and working backwards from the known value function $V_T^*(x) = \mathbb{1}_K(x)$. In this way, the value of $V_0^*(x)$ is approximated on the grid points of \mathcal{X} while an approximate control policy at each step k is computed by taking the maximum of (3) over \mathcal{U} . The advantages of this approach are that it is straightforward to estimate the approximation accuracy as a function of the grid size (under suitable regularity assumptions [4]) and that the approximate feedback control policy can be stored as a look-up table rather than computed online at each state. The disadvantage is that it quickly becomes intractable as the size of the state and control spaces increase.

B. Linear programming approach

We express the reach-avoid DP value function implicitly as a solution to an infinite dimensional LP, which we will later approximate using tools from function approximation and randomized convex optimization. Let \mathcal{F} denote the set of Borel measurable functions $\mathcal{F} := \{f : \mathcal{X} \rightarrow \mathbb{R}, f \text{ is Borel measurable}\}$. To simplify the presentation of our results we introduce two operators, defined for any Borel measurable function $V \in \mathcal{F}$:

$$\begin{aligned} \mathcal{T}_u[V](x) &= \mathbb{1}_K(x) + \mathbb{1}_{\bar{\mathcal{X}}}(x) \int_{\mathcal{X}} V(y) Q(dy|x, u) \\ \mathcal{T}[V](x) &= \max_{u \in \mathcal{U}} \mathcal{T}_u[V](x). \end{aligned} \quad (5)$$

For every $k \in \{0, \dots, T-1\}$ the solution to the recursive step in (2) can be constructed from the solution of an infinite dimensional LP using the relaxed inequality $V_k^*(x) \geq \mathcal{T}_u[V_{k+1}^*](x)$, $\forall (x, u) \in \bar{\mathcal{X}} \times \mathcal{U}$ of the equation $V_k^*(x) = \mathcal{T}[V_{k+1}^*](x)$, $\forall x \in \bar{\mathcal{X}}$. For the rest of the paper let ν be a probability measure supported on $\bar{\mathcal{X}}$.

Proposition 3. For $k \in \{0, \dots, T-1\}$, let V_{k+1}^* be the optimal value function at step $k+1$ in (2). Define the following infinite dimensional linear program:

$$\begin{aligned} J^* &:= \inf_{V(\cdot) \in \mathcal{F}} \int_{\bar{\mathcal{X}}} V(x) \nu(dx) \\ \text{subject to} \quad & V(x) \geq \mathcal{T}_u[V_{k+1}^*](x), \quad \forall (x, u) \in \bar{\mathcal{X}} \times \mathcal{U}. \end{aligned} \quad (6)$$

Then, V_k^* is a solution to (6) and any other solution is equal to V_k^* on $\bar{\mathcal{X}}$ ν -almost everywhere.

Proof. Since V_k^* is equal to the supremum of the right hand side of the constraint in (6), for any feasible $u(\cdot) \in \mathcal{F}$, $u(x) \geq V_k^*(x)$ for all $x \in \bar{\mathcal{X}} \times \mathcal{U}$ and thus $\int_{\bar{\mathcal{X}}} u(x) \nu(dx) \geq \int_{\bar{\mathcal{X}}} V_k^*(x) \nu(dx)$. On the other hand, $J^* \leq \int_{\bar{\mathcal{X}}} V_k^*(x) \nu(dx)$ since it is the least cost among the set of feasible Borel measurable functions. Putting these together we have that $J^* = \int_{\bar{\mathcal{X}}} V_k^*(x) \nu(dx)$ and V_k^* is an optimal solution. Notice that any other feasible function which differs from V_k^* on a set of ν -measure zero, is also optimal as it has the same cost. \square

Consider the following semi-norm on \mathcal{F} induced by the non-negative measure ν :

Definition 1 (ν -norm). $\|V\|_{1,\nu} := \int_{\bar{\mathcal{X}}} |V(x)| \nu(dx)$.

Any feasible solution to (6) is a point-wise upper bound on the value function V_k^* . Thus, Proposition 3 implies that for any solution V^* to the infinite dimensional LP we have $\|V^* - V_k^*\|_{1,\nu} = 0$.

In theory, we could start at $k = T-1$ and recursively solve problem (6) to construct the required value function V_0^* , up to a set of ν -measure zero. However, computing a solution to infinite dimensional LP problems in the form of (6) is generally intractable [18], [19] unless the set \mathcal{F} has a finite dimensional representation and the infinite constraints over $\bar{\mathcal{X}} \times \mathcal{U}$ can be reformulated or evaluated efficiently.

III. APPROXIMATION WITH A FINITE LINEAR PROGRAM

The reformulation of the stochastic reach-avoid problem in Section II-B results in an infinite dimensional LP. In this section, we discuss a finite dimensional approximation constructed in two steps. First, we replace the infinite dimensional function space \mathcal{F} with a finite dimensional one and discuss the effects of this restriction. Second, we replace the infinite constraints enforced on $\bar{\mathcal{X}} \times \mathcal{U}$ with a finite number of constraints, and provide probabilistic feasibility guarantees. As discussed in [20], the restriction to a finite dimensional basis can be interpreted as a type of state-space gridding in a function space. However, as demonstrated in the paper, it allows one to approximate stochastic reach-avoid problems of dimensions larger than what has been proposed in literature through gridding.

A. 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 $\{\phi_i\}_{i=1}^M$. We present the results of this section using abstract spaces \mathcal{F}^M and discuss in the next section the selection of basis elements that are particularly useful for the reach-avoid problem under consideration.

Proposition 4. Fix a function $f \in \mathcal{F}$ and let V^* be a solution to the following optimization problem:

$$\begin{aligned} \min_{V(\cdot) \in \mathcal{F}} \quad & \int_{\bar{\mathcal{X}}} V(x) \nu(dx) \\ \text{subject to} \quad & V(x) \geq \mathcal{T}_u[f](x), \quad \forall (x, u) \in \bar{\mathcal{X}} \times \mathcal{U}. \end{aligned} \quad (7)$$

Consider the following finite dimensional optimization problem:

$$\begin{aligned} \min_{w_1, \dots, w_M} \quad & \sum_{i=1}^M w_i \int_{\bar{\mathcal{X}}} \phi_i(x) \nu(dx) \\ \text{subject to} \quad & \sum_{i=1}^M w_i \phi_i(x) \geq \mathcal{T}_u[f](x), \quad \forall (x, u) \in \bar{\mathcal{X}} \times \mathcal{U} \\ & w \in \mathbb{R}^M. \end{aligned} \quad (8)$$

Denote by $\hat{V}^*(x) = \sum_{i=1}^M w_i^* \phi_i(x)$ the function constructed using an optimal solution of (8). Then $\hat{V}^*(x) \geq V^*(x)$ for all $x \in \bar{\mathcal{X}}$ and \hat{V}^* is a solution to the following optimization problem:

$$\begin{aligned} \min_{V(\cdot) \in \mathcal{F}^M} \quad & \|V - V^*\|_{1, \nu} \\ \text{subject to} \quad & V(x) \geq \mathcal{T}_u[f](x), \quad \forall (x, u) \in \bar{\mathcal{X}} \times \mathcal{U}. \end{aligned} \quad (9)$$

Proof. The feasibility regions of (8) and (9) are the same and since $V^*(x) := \max_{u \in \mathcal{U}} \mathcal{T}_u[f](x)$, any feasible $V \in \mathcal{F}^M$ upper bounds V^* on $\bar{\mathcal{X}}$, i.e. $V(x) \geq V^*(x)$ for all $x \in \bar{\mathcal{X}}$ and in particular $\hat{V}^*(x) \geq V^*(x)$ for all $x \in \bar{\mathcal{X}}$. Thus,

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

where C is equal to the integral of V^* on $\bar{\mathcal{X}}$ with respect to ν and is a constant since V^* is fixed. Since any function in $V \in \mathcal{F}^M$ can be written as $V(x) = \sum_{i=1}^M w_i \phi_i(x)$, minimizing $\int_{\bar{\mathcal{X}}} V(x) \nu(dx) - C$ is equivalent to minimizing $\int_{\bar{\mathcal{X}}} \sum_{i=1}^M w_i \phi_i(x) \nu(dx)$ over the weights w_1, \dots, w_M . \square

The finite dimensional optimization problem in Proposition 4 can be used to recursively construct an approximation to V_0^* on $\bar{\mathcal{X}}$. 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 denoted by $\{\phi_i^k\}_{i=1}^{M_k}$.

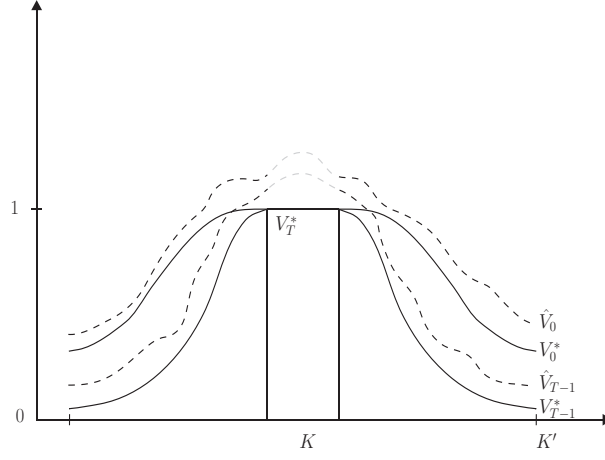


Fig. 1: The chosen basis for approximating the value function must be rich in order to avoid introducing errors. In this hypothetical 1-dimensional example it is illustrated that the solution obtained by Proposition 5 is not necessarily tight to the optimal value function on $\bar{\mathcal{X}}$.

Proposition 5. Fix a collection of basis sets $\{\phi_i^k\}_{i=1}^{M_k}$. Start with the known value function V_T^* and recursively construct $\hat{V}_{T-1}(x), \dots, \hat{V}_0(x)$ where $\hat{V}_k(x) = \sum_{i=1}^{M_k} \hat{w}_i^k \phi_i^k(x)$ and \hat{w}_i^k is obtained by Proposition 4. For a fixed $\hat{V}_{k+1} \in \mathcal{F}^{M_k}$ each function \hat{V}_k is a solution to the corresponding optimization problem:

$$\begin{aligned} \hat{V}_k \in \arg \min_{V(\cdot) \in \mathcal{F}^{M_k}} \quad & \|V - V_k^*\|_{1,\nu} \\ \text{subject to} \quad & V(x) \geq \mathcal{T}_u[\hat{V}_{k+1}](x) \quad \forall (x, u) \in \bar{\mathcal{X}} \times \mathcal{U} \end{aligned} \quad (10)$$

and $\hat{V}_k(x) \geq V_k^*(x)$ for all $x \in \bar{\mathcal{X}}$.

Proof. For $k = T - 1$, the constraint in (10) implies that $\hat{V}_{T-1}(x) \geq \mathcal{T}[V_T^*](x) = V_{T-1}^*(x)$. Applying \mathcal{T} on both sides we have by the monotonicity of \mathcal{T} (see [12]) that $\mathcal{T}[\hat{V}_{T-1}](x) \geq \mathcal{T}[V_{T-1}^*](x) = V_{T-2}^*(x)$. By the constraints in (10) at $k = T - 2$ we have that $\hat{V}_{T-2}(x) \geq \mathcal{T}[\hat{V}_{T-1}](x)$ which implies that $\hat{V}_{T-2}(x) \geq V_{T-2}^*(x)$. Repeating, we arrive at the result $\hat{V}_0(x) \geq V_0^*(x)$. Using the fact that $\hat{V}_k(x) \geq V_k^*(x)$ for all $x \in \bar{\mathcal{X}}$ we conclude in the same way as in Proposition 5 that $\hat{V}_k(x)$ minimizes the ν -norm distance to V_k^* while respecting the corresponding constraint. \square

The feasibility of each problem in (10) can be guaranteed by adding the constant function $\phi(x) = \mathbb{1}_{\bar{\mathcal{X}}}(x)$ to the set of basis elements. Each function \hat{V}_k is bounded by construction and $\mathbb{1}_{\bar{\mathcal{X}}}(x)$ can always be scaled by an arbitrarily large number to form a trivial upper bound on V_k^* .

The process described in Proposition 5 is not directly applicable since it involves solving a sequence of optimization problems with an infinite number of constraints, the pairs $(x, u) \in \mathcal{X} \times \mathcal{U}$. Even when one can solve such problems, there is no guarantee that the obtained solution accurately approximates the optimal value function. To appreciate this more, consider the hypothetical example of Figure 1 where we illustrate that the solution obtained by Proposition 5 is not necessarily a good approximation of the optimal value function. The recursive approximation method described in Proposition 5 at $k = T - 1$ guarantees that $\hat{V}_{T-1} \geq \mathcal{T}_u[V_T^*] \implies \hat{V}_{T-1} \geq V_{T-1}^*$ on $\bar{\mathcal{X}}$ but the equality will not be achieved unless V_{T-1}^* is a member of $\mathcal{F}^{M_{T-1}}$. If each \mathcal{F}^{M_k} is not rich enough, the difference might be growing, resulting in poor approximation of V_0^* .

B. Restriction to a finite number of constraints

Finite dimensional problems with infinite constraints (similar to the ones appearing in the recursive process of Proposition 5) are called semi-infinite or robust optimization problems and are difficult to solve

unless specific structure is imposed on $\bar{\mathcal{X}} \times \mathcal{U}$ and the basis functions ([21], [22], [23]). An alternative approach is to select a finite set of points from $\bar{\mathcal{X}} \times \mathcal{U}$ and impose the constraints on those points. Problems in the form of (8) are then reduced to finite LP problems and can be solved to optimality using commercially available solvers. We will use the scenario approach [24] to quantify the stage-wise feasibility properties of the solutions constructed using sampled data. For a discussion on the related performance properties, see [25], [26].

Let $S_k := \{(x^i, u^i)\}_{i=1}^{N_k}$ denote a set of $N_k \in \mathbb{N}$ elements from $\bar{\mathcal{X}} \times \mathcal{U}$ and for a fixed function \tilde{V}_{k+1} , consider the following parametric, finite LP

$$\begin{aligned} \tilde{w}^k(S_k) := \arg \min_{\tilde{w}_1, \dots, \tilde{w}_{M_k}} \quad & \sum_{i=1}^{M_k} \tilde{w}_i^k \int_{\bar{\mathcal{X}}} \phi_i^k(x) \nu(dx) \\ \text{subject to} \quad & \sum_{i=1}^{M_k} \tilde{w}_i^k \phi_i^k(x) \geq \mathcal{T}_u[\tilde{V}_{k+1}](x), \quad \forall (x, u) \in S_k. \end{aligned} \quad (11)$$

Under the assumption that for any set S_k , the feasible region of (11) is non-empty and contains a unique optimizer, the authors in [24] generalize the feasibility properties of a solution to (11) with respect to a solution to (8), as a function of S_k . The following statement is a straightforward application of Theorem 3.5 in [24] to the problem under consideration here.

Theorem 1. *For every $k \in \{0, \dots, T-1\}$, fix a collection of functions $\{\phi_i^k\}_{i=1}^{M_k}$, choose $\varepsilon_k, \beta_k \in (0, 1)$ and construct S_k by drawing N_k independent identically distributed sample points from $\bar{\mathcal{X}} \times \mathcal{U}$, where*

$$N_k \geq S(\varepsilon_k, \beta_k, M_k) = \min \left\{ N \in \mathbb{N} \mid \sum_{i=0}^{M_k-1} \binom{N}{i} \varepsilon_k^i (1 - \varepsilon_k)^{N-i} \leq \beta_k \right\}.$$

The approximate value function $\tilde{V}_k(x) = \sum_{i=1}^{M_k} \tilde{w}_i^k(S_k) \phi_i^k(x)$ constructed using the optimal solution to (11) satisfies

$$\mathbb{P}_{\bar{\mathcal{X}} \times \mathcal{U}} \left[\tilde{V}_k(x) < \mathcal{T}_u[\tilde{V}_{k+1}](x) \right] \leq \varepsilon_k \quad (12)$$

with confidence $1 - \beta_k$ with respect $\mathbb{P}_{\{\bar{\mathcal{X}} \times \mathcal{U}\}^{N_k}}$ where $\mathbb{P}_{\bar{\mathcal{X}} \times \mathcal{U}}$ denotes the probability measure used to construct S_k and $\mathbb{P}_{\{\bar{\mathcal{X}} \times \mathcal{U}\}^{N_k}}$ the measure constructed by taking N_k products of $\mathbb{P}_{\bar{\mathcal{X}} \times \mathcal{U}}$.

To give some intuition about the bound on the sample number in Theorem 1, we use the explicit bound $N_k \geq \frac{e}{e-1} \frac{2}{\varepsilon_k} \left(M_k + \ln \left(\frac{1}{\beta_k} \right) \right)$ from [27] where e denotes the base of the natural logarithm. Since N_k depends logarithmically on β_k , the confidence level $1 - \beta_k$ can be pushed close to 1 without increasing N_k much. The inner probability statement in (12) can be interpreted as follows: The approximate solution function $\tilde{V}_k(x) = \sum_{i=1}^{M_k} \tilde{w}_i^k(S_k) \phi_i^k(x)$ violates the constraint $\mathcal{T}_u[\tilde{V}_{k+1}](x)$ on a set of at most ε_k measure, with respect to $\mathbb{P}_{\bar{\mathcal{X}} \times \mathcal{U}}$. In contrast to the solution in (8), the constructed value functions are probabilistic upper bounds on \mathcal{X} .

According to [24, Thm 3.3], one can also bound the probability that the objective value of (11) exceeds the optimal value of the semi-infinite optimization problem in (8).

Corollary 1. *Consider the setup of Theorem 1 and let $\tilde{w}_k(S_k)$ denote the optimal solution of (11) and $\tilde{w}_k(S_k^+)$ the optimal solution of (11), using one additional sample point (S_k^+ contains one more element than S_k). It then holds that,*

$$\mathbb{P}_{\bar{\mathcal{X}} \times \mathcal{U}} \left[\sum_{i=1}^{M_k} \tilde{w}_i^k(S_k) \int_{\bar{\mathcal{X}}} \phi_i^k(x) \nu(dx) < \sum_{i=1}^{M_k} \tilde{w}_i^k(S_k^+) \int_{\bar{\mathcal{X}}} \phi_i^k(x) \nu(dx) \right] \leq \varepsilon_k \quad (13)$$

with confidence $1 - \beta_k$ with respect $\mathbb{P}_{\{\bar{\mathcal{X}} \times \mathcal{U}\}^{N_k}}$.

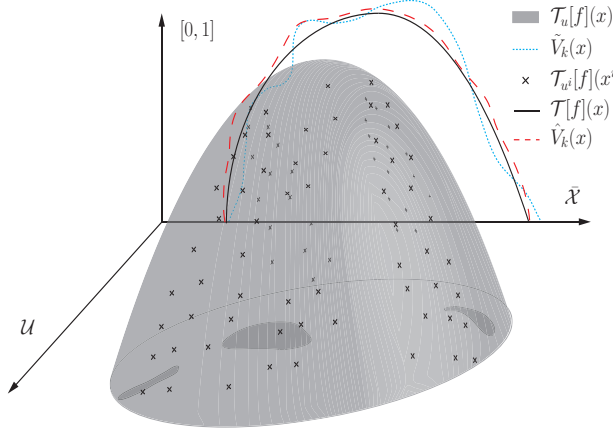


Fig. 2: Approximation after sampling depicted on $\bar{\mathcal{X}} \times \mathcal{U}$.

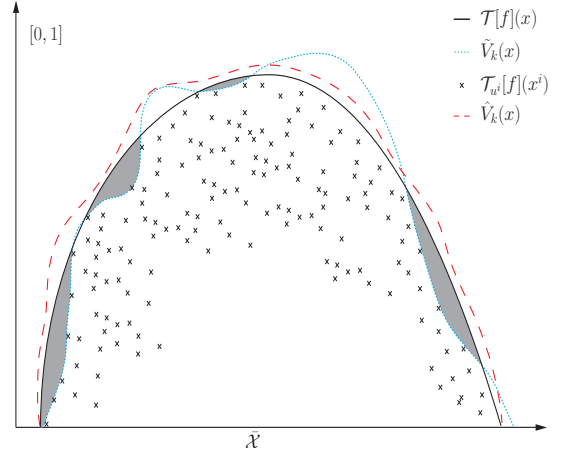


Fig. 3: Approximation after sampling, projected on $\bar{\mathcal{X}}$.

To further clarify the statement of Theorem 1, consider the illustration in Figures 2 and 3. Assuming that $\bar{\mathcal{X}}$ and \mathcal{U} are one dimensional and for a hypothetical function f , we have plotted the value of the constraint $\mathcal{T}_{u^i}[f](x^i)$ (denoted by crosses) for a collection of sampled points (x^i, u^i) . The dotted lines correspond to the solution of (11) for the particular f , denoted by $\tilde{V}_k(x)$. The dashed lines correspond to the solution of (10) for the particular f , denoted by $\hat{V}_k(x)$, while the solid lines correspond to $\mathcal{T}[f](x)$. As stated in Theorem 1, $\tilde{V}_k(x)$ is greater than or equal to $\mathcal{T}_u[f](x)$ for most pairs (x, u) . The relation of $\tilde{V}_k(x)$ to $\mathcal{T}[f](x)$ remains unclear since $\tilde{V}_k(x)$ could exceed or stay below $\mathcal{T}[f](x)$ arbitrarily many times - as long as the measure of pairs (x, u) where it is below is smaller than ε_k . In Figure 2, we have plotted the sets where $\tilde{V}_k(x) < \mathcal{T}_u[f](x)$ on $\bar{\mathcal{X}} \times \mathcal{U}$. In Figure 3, the shaded areas only indicate that there exist pairs (x, u) such that $\tilde{V}_k(x) < \mathcal{T}_u[f](x)$.

C. Interpretation of results

Proposition 5 relates the optimal value functions to the solution of a semi-infinite LP with finitely many decision variables and infinitely many constraints (10). The restriction to a finite number of decision variables reduces the feasible region of each infinite LP and each \hat{V}_k is an upper bound on V_k^* and sub-optimal in terms of objective value, unless $V_k^* \in \mathcal{F}^{M_k}$. The recursive nature of Proposition 5 leads to an accumulation of sub-optimality over the time horizon and a conservative approximation of V_0^* . Theorem 1 relates the solution of each semi-infinite LP in (10) to that of a parametric, finite LP constructed via constraint sampling (11). The sampling technique proposed leads to a relaxation of the infinite constraints in (10), enlarging the corresponding feasible region. As a result, each approximate value function \tilde{V}_k is super-optimal with respect to \hat{V}_k , feasible in probability and no longer a guaranteed upper bound on V_k^* .

The accuracy of the proposed approach depends on the properties of each \mathcal{F}^{M_k} . In general, larger values for M_k (more basis function elements) result in better approximations since \mathcal{F}^{M_k} becomes more dense in \mathcal{F} at the expense of more decision variables in (11). More decision variables require more sample points to be used in (11) to satisfy the probabilistic feasibility guarantees in (12). Evidently, there is a trade-off between accuracy and complexity which is caused by the fact that the \mathcal{F}^{M_k} that result in the best approximation, are not known a-priori.

IV. IMPLEMENTATION FOR STOCHASTIC REACH-AVOID

In this section, we discuss the implementation of the proposed technique and approximate the infinite dimensional LP associated with the discrete-time stochastic reach-avoid problem for MDPs [3], [12].

A. Basis function choice

A basis type commonly used in the literature consists of piecewise constant functions defined over disjoint regions of $\bar{\mathcal{X}}$ [28], [29]. Such basis elements result in strong statements about the overall approximation quality but require a prohibitively large number of regions to achieve small approximation errors. Another common choice is polynomial basis functions, typically used to approximate the infinite constraints by linear matrix inequality (LMI) constraints using sum-of-squares programming [30] and tools from polynomial optimization [31]. In this way one circumvents the need to sample the constraints at the expense of solving semi-definite programming (SDP) problems that become large and difficult to solve as the degree of polynomials increases. The approach has been successfully used in [32], [33], [34] but to the best of our knowledge it has not yet been applied to reach-avoid problems.

We consider basis function sets where each element is a parametrized Gaussian radial basis function. Our choice is motivated by the strong approximation capabilities of such functions, discussed in [35], [36], [37], [38], as well as the efficient computation of constraints in the LP problems of the previous section. Each RBF element ϕ_i^k is a mapping from \mathbb{R}^n to \mathbb{R} and has the form:

$$\phi_i^k(x) = \prod_{l=1}^n \frac{1}{\sqrt{2\pi s_l^{i,k}}} e^{-\frac{1}{2} \frac{(x_l - c_l^{i,k})^2}{s_l^{i,k}}} \quad (14)$$

where $x, c^{i,k}, s^{i,k} \in \mathbb{R}^n$ and all $c^{i,k}, s^{i,k}$ are chosen a-priori. Each candidate approximation function is constructed by taking a finite sum of weighted functions ϕ_i^k .

1) *RBFs and MDP kernel:* We now restrict attention to MDPs generated by discrete time systems subject to additive Gaussian mixture noise. For $f : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}^n$ consider the dynamics $x_{k+1} = f(x_k, u_k) + w_k$ where each w_k is distributed according to a Gaussian mixture probability density function of the form $\sum_{j \in \mathbb{N}_+} \alpha_j \mathcal{N}(\mu_j, \Sigma_j)$ where $\sum_{j \in \mathbb{N}_+} \alpha_j = 1$. The density function of x_{k+1} is then a Gaussian mixture probability density function with shifted mean $x_{k+1} \sim \sum_{j \in \mathbb{N}_+} \alpha_j \mathcal{N}(f(x_k, u_k) + \mu_j, \Sigma_j)$. We assume that each Σ_j is diagonal so the density function of the controlled transition kernel Q for x_{k+1} can be written as a weighted sum of functions in the form of (14). A useful property of functions in \mathcal{F}^{M_k} is that pairwise products are proportional to a Gaussian RBF with known mean and variance [35, Section 2]. Consider two functions $f^1, f^2 \in \mathcal{F}^{M_k}$ such that $f^1(x) = \sum_{i=1}^{M_k} w_i^1 \phi_i^k(x)$ and $f^2(x) = \sum_{j=1}^{M_k} w_j^2 \phi_j^k(x)$. We then have that $f(x) = f^1(x)f^2(x) = \sum_{i=1}^{M_k} \sum_{j=1}^{M_k} w_{i,j} \tilde{\phi}_{i,j}^k(x)$ where $w_{i,j} = w_i^1 w_j^2$ and $\tilde{\phi}_{i,j}^k$ is equal to

$$\tilde{\phi}_{i,j}^k(x) = \prod_{l=1}^n \frac{\gamma_l^{i,j,k}}{\sqrt{2\pi s_l^{i,j,k}}} e^{-\frac{1}{2} \frac{(x_l - c_l^{i,j,k})^2}{s_l^{i,j,k}}}$$

with

$$c_l^{i,j,k} = \frac{c_l^{i,k} s_l^{j,k} + c_l^{j,k} s_l^{i,k}}{s_l^{i,k} + s_l^{j,k}}, \quad s_l^{i,j,k} = \sqrt{\frac{s_l^{i,k} s_l^{j,k}}{s_l^{i,k} + s_l^{j,k}}}, \quad \gamma_l^{i,j,k} = \frac{1}{\sqrt{2\pi(s_l^{i,k} + s_l^{j,k})}} e^{-\frac{c_l^{i,k} - c_l^{j,k}}{2(s_l^{i,k} + s_l^{j,k})}}.$$

2) *Multi-dimensional integration:* Investigating the constraints in (11) it is clear that the analytical computation of the expectation integral involved in $\mathcal{T}_u[\tilde{V}_{k+1}](x)$ for every sampled point $(x, u) \in \bar{\mathcal{X}} \times \mathcal{U}$ would provide substantial computational benefit. If the sets K and K' can be written as finite unions of disjoint hyper-rectangles (i.e. for a set $A = \bigcup_d A_d = \bigcup_d \{[a_1^d, b_1^d] \times \dots \times [a_n^d, b_n^d]\}$, where $n = \dim(\mathcal{X})$ and \bigcup_d denotes a finite union indexed by d) and the measure ν is constructed as a product measure of known measures ν_l , the integral of functions in \mathcal{F}^{M_k} can be split into one dimensional integrals of Gaussian functions. For simplicity in the calculations we assume that ν is a uniform measure and can be split into $\nu(dx) = \prod_{l=1}^n \nu_l(dx_l)$. The computation can be extended to measures ν that can be written as

Gaussian mixtures.

$$\begin{aligned}
\int_A \tilde{V}_k(x) \nu(dx) &= \sum_d \int_{A_d} \tilde{V}_k(x) \nu(dx) \\
&= \sum_d \sum_{i=1}^{M_k} \tilde{w}_i^k \int_{A_d} \phi_i^k(x) \nu(dx) = \sum_d \sum_{i=1}^{M_k} \tilde{w}_i^k \prod_{l=1}^n \int_{a_l^d}^{b_l^d} \frac{1}{\sqrt{2\pi s_l^{i,k}}} e^{-\frac{1}{2} \frac{(x_l - c_l^{i,k})^2}{s_l^{i,k}}} \nu_l(dx_l) \\
&= \sum_d \sum_{i=1}^{M_k} \tilde{w}_i^k \prod_{l=1}^n -\frac{1}{2} \operatorname{erf} \left(\frac{x_l - c_l^{i,k} - b_l^d}{\sqrt{2s_l^{i,k}}} \right) + \frac{1}{2} \operatorname{erf} \left(\frac{x_l - c_l^{i,k} - a_l^d}{\sqrt{2s_l^{i,k}}} \right)
\end{aligned}$$

where erf denotes the error function defined as $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$ and \sum_d denotes a finite sum indexed by d .

As a consequence of the above, we can analytically integrate the approximate value functions constructed using Theorem 1 on \mathcal{X} . In particular, if $K = \bigcup_p K_p$ and $K' = \bigcup_m K'_m$ where K_p and K'_m are finite unions of disjoint hyper-rectangles (indexed by m and p respectively), we have by definition of each \tilde{V}_k that

$$\begin{aligned}
\int_{\mathcal{X}} \tilde{V}_k(y) Q(dy|x, u) &= \sum_m \sum_{i=1}^{M_k} \tilde{w}_i^k \int_{K'_m} \phi_i^k(y) Q(dy|x, u) - \sum_p \sum_{i=1}^{M_k} \tilde{w}_i^k \int_{K_p} \phi_i^k(y) Q(dy|x, u) \\
&\quad + \sum_p \int_{K_p} Q(dy|x, u).
\end{aligned}$$

Whenever the safe and target sets are not unions of hyper-rectangles, one can approximate them as such to arbitrary accuracy using non-overlapping hyper-rectangles¹ (see for example [39]), or proceed by calculating upper bounds on the value of the integral over general polytopic sets as suggested in [40], [41], [42]. Calculating such bounds requires solving SDP problems which are computationally demanding and will significantly increase the total computation time of the proposed method.

3) *RBF centers and variances*: One of the most challenging problems in approximation algorithms using projections to finite basis classes is the selection of the basis function parameters. In this work, we fix the centers and variances of the RBF elements randomly before initializing the recursive algorithm and then optimize over their scalar weights. We sample centers uniformly from the set $\bar{\mathcal{X}}$ and variances from a bounded set that varies according to problem data and constitutes a design choice. The chosen approach is computationally efficient as it reduces to an LP with finitely many variables and constraints that can be solved in polynomial time using existing software. One could try to further optimize the approximation performance by adaptively choosing centers and variances for the constructed basis [43], [44]. Since we are addressing the approximation of a value function of a particular optimal control problem, we can exploit the system structure in choosing the RBF parameters. In particular, one could analyze the system dynamics offline and compute a rough outer approximation of reachable sets (for each time step) within which the RBFs can be placed. Developing such methods would be extremely useful in our approximation scheme and will be investigated in future work.

B. Recursive value function approximation

Theorem 1 suggests a recursive method to approximate the solution to the optimal value function V_0^* of the recursion in (2). As a first step we choose the required violation and confidence levels for each approximate value function in order to use Theorem 1. We initialize the recursion with the known value function at time T and for every step $k \in \{T-1, \dots, 0\}$ sample M_k centers and variances. We then sample

¹The number of hyper-rectangles needed for a given approximation accuracy can be prohibitively large.

Algorithm 1 Approximate value function

- 1: Choose the required violation levels $\{\varepsilon_0, \dots, \varepsilon_{T-1}\}$ and confidences $\{\beta_0, \dots, \beta_{T-1}\}$.
 - 2: Initialize $\tilde{V}_T(x) \leftarrow \mathbf{1}_K(x)$.
 - 3: **for** $k = T - 1$ **to** $k = 0$ **do**
 - 4: Construct \mathcal{F}^{M_k} by sampling M_k centers $\{c_i\}_{i=1}^{M_k}$ uniformly from $\bar{\mathcal{X}}$ and variances $\{s_i\}_{i=1}^{M_k}$ uniformly from a bounded set.
 - 5: Sample $S(\varepsilon_k, \beta_k, M_k)$ pairs (x^s, u^s) from $\bar{\mathcal{X}} \times \mathcal{U}$.
 - 6: **for all** (x^s, u^s) **do**
 - 7: Evaluate $\mathcal{T}_{u^s}[\tilde{V}_{k+1}](x^s)$.
 - 8: **end for**
 - 9: Solve the finite LP in (11) to obtain $\tilde{w}^k = (\tilde{w}_1^k, \dots, \tilde{w}_{M_k}^k)$.
 - 10: Set the approximated value function on $\bar{\mathcal{X}}$ to $\tilde{V}_k(x) = \sum_{i=1}^{M_k} \tilde{w}_i^k \phi_i^k(x)$.
 - 11: **end for**
-

Algorithm 2 Approximate controller

- 1: **for** $k \in \{0, \dots, T - 1\}$ **do**
 - 2: Measure the system state x_k .
 - 3: Compute the gradient and hessian functions of $\int_{\mathcal{X}} \tilde{V}_{k+1}^*(y) Q(dy|x_k, u)$ with respect to u .
 - 4: Solve the optimization problem in (3) using second-order methods to obtain $\tilde{\mu}^*(x_k)$.
 - 5: Apply the calculated control input to the system.
 - 6: **end for**
-

points (x, u) from $\bar{\mathcal{X}} \times \mathcal{U}$, evaluate $\mathcal{T}_u[\tilde{V}_{k+1}](x)$ and compute the weights that minimize the chosen cost function in (11) while being above the value of $\mathcal{T}_u[\tilde{V}_{k+1}](x)$ at the sampled points. Using the computed weights \tilde{w}^k we construct the approximate value function on $\bar{\mathcal{X}}$ as $\tilde{V}_k(x) = \sum_{i=1}^{M_k} \tilde{w}_i^k \phi_i^k(x)$ and carry on with the recursion. The whole process is summarized in Algorithm 1.

C. Control policy based on approximate value function

Using the sequence of approximate value functions one can compute an approximate control policy for any $x \in \bar{\mathcal{X}}$ by solving:

$$\begin{aligned} \tilde{\mu}_k^*(x) &= \arg \max_{u \in \mathcal{U}} \left\{ \mathbf{1}_K(x) + \mathbf{1}_{\bar{\mathcal{X}}}(x) \int_{\mathcal{X}} \tilde{V}_{k+1}^*(y) Q(dy|x, u) \right\} \\ &= \arg \max_{u \in \mathcal{U}} \int_{\mathcal{X}} \tilde{V}_{k+1}^*(y) Q(dy|x, u). \end{aligned} \tag{15}$$

Although the optimization problem in (15) is non-convex, standard gradient based algorithms can be employed to obtain a local solution. In particular, the cost function is by construction smooth with respect to u for a fixed $x \in \bar{\mathcal{X}}$ and the gradient and Hessian information can be analytically calculated using the erf function. Moreover, the decision space \mathcal{U} is typically low dimensional (in most mechanical systems for example $\dim \mathcal{U} < \dim \mathcal{X}$) and mature software is available [45] to compute locally optimal solutions. The process of calculating a control input at time k for a fixed state x_k is summarized in Algorithm 2. An alternative approach would be to use randomized techniques similar to the approach in [11], [24], [46], [47].

One way to evaluate the approximate value function is to investigate the performance of the associated control policies defined by $\tilde{\mu}_k(x)$ for $k \in \{0, \dots, T - 1\}$. That is, starting at some $x_0 \in \bar{\mathcal{X}}$, one can implement the policy in simulation and gather statistics via Monte Carlo methods of how many times the system satisfies the desired reach-avoid objective.

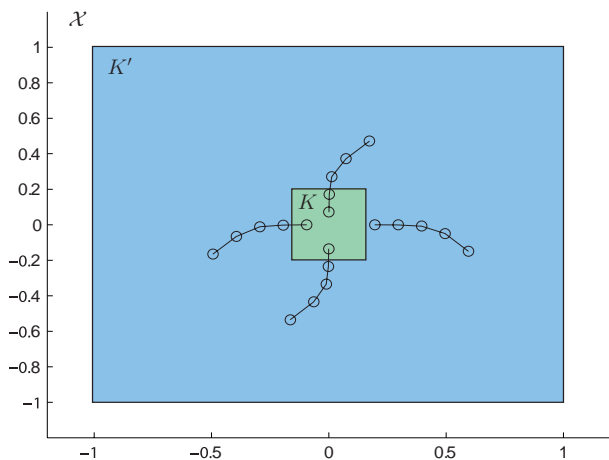


Fig. 4: Regulation example.

$\dim(\mathcal{X} \times \mathcal{U})$	4D	6D	8D
Basis elements	100	500	1000
N_k	3960	19960	39960
$\ \tilde{V}_0 - V_{\text{ADP}}\ $	0.0692	0.1038	0.2241
Constr. time (sec)	4	85	450
LP time (sec)	2	50	520
Memory	3.2MB	80MB	320MB

TABLE I: Parameters and properties of the constructed value function approximation for origin regulation problems of $\dim(\mathcal{X} \times \mathcal{U}) = 4, 6, 8$.

V. NUMERICAL CASE STUDIES

We analyze the numerical performance of the proposed method on three reach-avoid problems that differ in structure and complexity. The first problem considered is a variation of an origin regulation problem with control constraints where a saturated linear quadratic Gaussian (LQG) controller can be designed in a straightforward way for systems of large dimension. The second problem is identical to the first one with the addition of state constraints in the form of obstacles placed on the state space. A benchmark controller for this problem can be computed using a mixed-integer quadratic programming (MIQP) formulation where we explicitly take obstacles into account by introducing integer variables in a model predictive control formulation.

Even though the LQG and MIQP based approaches are not optimal for the reach-avoid problems considered, their scalability to high dimensions motivates their use as a reference. Due to the structure of the considered problems, the resulting heuristic controllers arguably also perform well for the reach-avoid objective. Our comparison however is mostly qualitative since the objective functions used in the design of the LQG and MIQP controllers are inherently different from the reach-avoid objective.

In the final numerical example, we demonstrate the full potential of our method by solving an 8-dimensional nonlinear reach-avoid problem for which designing a benchmark controller is not straightforward. The scenario used is inspired by path planning for autonomous race cars in the presence of obstacles. To compute reach-avoid based controllers for all problems, we have used the method presented in this paper without exploiting any special structure in the systems. All value function approximations and control policy simulations were done on an Intel Core i7 Q820 CPU clocked @1.73 GHz with 16GB of RAM memory, using IBM's CPLEX optimization toolkit in its default settings.

A. Reach-avoid origin regulation

Consider the reach-avoid problem of maximizing the probability that the state of a controlled linear system with additive Gaussian noise reaches a target set around the origin within T discrete time steps while staying in a safe set. For simplicity, we consider systems described by the equation

$$x_{k+1} = x_k + u_k + w_k \quad (16)$$

where $x_k \in \mathcal{X} = \mathbb{R}^n$, $u_k \in \mathcal{U} = [-0.1, 0.1]^n$ and each w_k is distributed as a Gaussian random variable $w_k \sim \mathcal{N}(\mathbf{0}_{n \times n}, \Sigma)$ with diagonal covariance matrix. We define a target set $K = [-0.1, 0.1]^n$ around the origin and a safe set $K' = [-1, 1]^n$ (Figure 4). For a time horizon of $T = 5$, we fix violation and confidence levels for every $k \in \{0, \dots, 4\}$ to $\varepsilon_k = 0.05$, $1 - \beta_k = 0.99$ and follow the steps of Algorithm 1 to obtain a sequence of approximate value functions. The variances for each basis element

are sampled from $[0.02, 0.095]$. To estimate the performance of the approximate control policy we follow the steps in Algorithm 2 starting from 100 different initial conditions $x_0 \in \bar{\mathcal{X}}$ selected such that the considered time horizon is long enough to reach K . For each initial condition we generate 100 different sample trajectories, using samples from each w_k , and count the total number that achieve the reach-avoid objective. The empirical success probability of the reach-avoid controller is denoted by V_{ADP} . Table I reports the values of all design parameters used in the ADP process to compute the approximate value functions for problems of $\dim(\mathcal{X} \times \mathcal{U}) = 4, 6, 8$. We denote by $\|\tilde{V}_0 - V_{\text{ADP}}\|$ the mean absolute difference between V_{ADP} and \tilde{V}_0 computed over the considered initial conditions. The memory and computation time reported is that of constructing and solving each LP in the recursive process.

A benchmark control policy for this problem can be computed by solving an LQG problem of the form

$$\begin{aligned} \min_{\{u_k\}_{k=0}^{T-1}} \mathbb{E}_{w_k} & \left(\sum_{k=0}^{T-1} x_k^\top Q x_k + u_k^\top R u_k \right) + x_T^\top Q x_T \\ \text{subject to} \quad & (16), x_0 \in \bar{\mathcal{X}} \end{aligned} \quad (17)$$

where Q and R have been chosen to correspond to the largest ellipsoids inscribed in K and \mathcal{U} respectively, computed via convex optimization [40, Ch. 8]. Whenever the resulting LQG control input (calculated via the Riccati difference equation) is infeasible, we project it on the feasible set \mathcal{U} . Starting from the same initial conditions as with the reach-avoid based controller, we simulate the performance of the LQG controller by sampling from each w_k , generating 100 different sample trajectories, counting the total number that reach K without leaving K' . In the comparison with the reach-avoid based controller, the empirical success probability of the LQG controller is denoted by V_{LQG} (see Table II, Table III, Figure 5, Figure 6).

Since the design criteria of the LQG and reach-avoid based methods are different, our comparison is qualitative and we have recorded empirical success probabilities for both policies with respect to the reach-avoid objective (denoted by V_{LQG} and V_{ADP} respectively). Figure 5 is showing the mean absolute difference between V_{LQG} and V_{ADP} over the initial conditions, as a function of the basis elements used to construct the approximate value functions. Each line on the graph corresponds to a problem of different $\dim(\mathcal{X} \times \mathcal{U})$ and we can clearly see the trend of increasing accuracy as the number of basis elements increases. In Table II we show the trade-off between accuracy and computation resources for the 6D problem; the situation is analogous in the 4D and 8D problems. Figure 6 is showing the same difference metric as Figure 5, this time as a function of the total number of samples for a fixed number of basis elements. As expected (inspect Theorem 1), changing the total number of samples N_k has a direct effect on the violation probability ε_k (assuming constant $\beta_k = 0.01$) and consequently on the approximation quality. Again, as the number of sample points increases, the accuracy increases at the expense of computation resources (Table III).

B. Reach-avoid origin regulation with obstacles

Consider the same origin regulation problem of Section V-A with the addition of obstacles placed randomly within the state space (Figure 7). The reach-avoid objective in this case is to maximize the probability that the system in (16) reaches K without leaving K' or reaching any of the obstacle sets K_α . In the reach-avoid framework, this problem is equivalent to the one in Section V-A, using the same target set K and a different safe set $K' = \mathcal{X} \setminus \bigcup_j K_\alpha^j$ where each K_α^j denotes one of the hyper-rectangular obstacle sets and \bigcup_j is a finite union indexed by j . For a time horizon of $T = 7$, we fix the values of the required violation and confidence levels for every $k \in \{0, \dots, 6\}$ to $\varepsilon_k = 0.05$, $1 - \beta_k = 0.99$ and follow the steps of Algorithms 1 and 2. The variances for each basis element in this problem are sampled from $[0.02, 0.095]$. We simulate the performance of the reach-avoid controller starting from 100 different initial conditions selected such that at least one obstacle blocks the direct path to the origin and the horizon is long enough to reach the target set. We sample 100 different instances from each w_k and generate

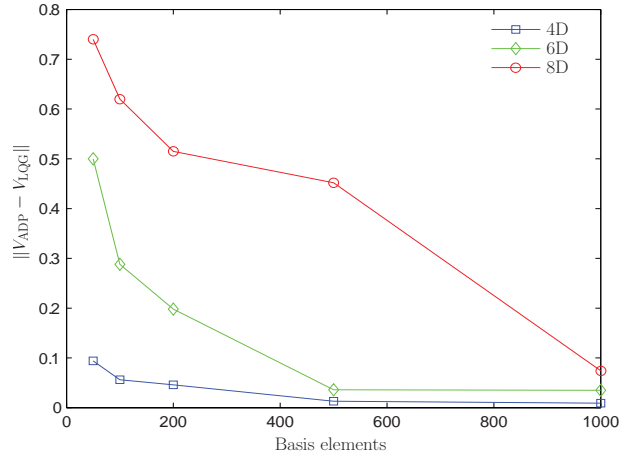


Fig. 5: Mean absolute difference between the empirical reach-avoid probabilities achieved by the ADP (V_{ADP}) and LQG (V_{LQG}) policies as a function of approximation basis elements for problems of $\dim(\mathcal{X} \times \mathcal{U}) = 4, 6, 8$.

Basis elements	$\ V_{\text{ADP}} - V_{\text{LQG}}\ $	Constr. time (sec)	LP time (sec)	Memory
50	0.504	1.2	0.143	784KB
100	0.288	4	2.324	3.2MB
200	0.052	15	3.41	12.8MB
500	0.036	85	50	80MB
1000	0.035	450	850	320MB

TABLE II: Accuracy and computational requirements as a function of basis elements for $\dim(\mathcal{X} \times \mathcal{U}) = 6$.

different sample trajectories, computing the empirical success probability (denoted by V_{ADP}) by counting the total number of trajectories that reach K while avoiding reaching any of the obstacles or leaving K' .

The problem of regulating (17) to the origin without reaching any obstacles is in the category of path planning and collision avoidance problems that have been studied thoroughly in the control community [48], [49], [50], [51]. We adopt the formulation in [49] and formulate the problem as an MiQP using mixed logic dynamics (MLD) [52]. Using the MLD language, the non-convex state constraints of staying outside a hyper-rectangular set can be transformed into linear constraints with binary variables. One can then write pairs² of linear inequalities that will be simultaneously true if and only if the associated binary variable is equal to one and the system state is outside the corresponding facet of the hyper-rectangle. Using this formulation we express the overall path planning and collision avoidance problem as an MiQP that can be solved to optimality using standard branch and bound techniques implemented in most commercially available optimization software. In order to take noise into account, we truncate the density function of the random variables w_k at 95% of their total mass and enlarge each obstacle set K_α by the maximum value of the truncated w_k in each dimension. An alternative approach to handle process noise would be to include the bounded uncertainty implicitly in the control problem formulation, see [53], or employ randomized techniques and solve the problem using sampling, see [25].

Starting from the same initial conditions as in the ADP approach, we simulate the performance of the MiQP based control policy by sampling from each w_k and generating 100 different trajectories, implementing the policy in receding horizon. The empirical success probability of trajectories that reach K while avoiding reaching any of the obstacles or leaving K' is denoted by V_{MiQP} .

Since the objectives of the MiQP and ADP formulations are different (even though both methods exploit knowledge about the target and the avoid sets) our comparison is again qualitative and we have

²One pair for each of the half-spaces defining each obstacle.

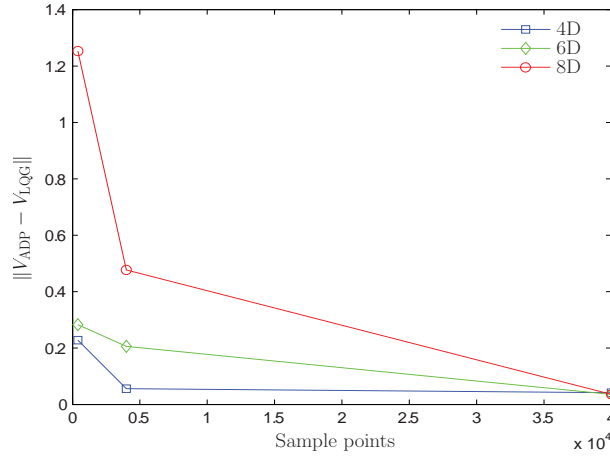


Fig. 6: Mean absolute difference between the empirical reach-avoid probabilities achieved by the ADP (V_{ADP}) and LQG (V_{LQG}) policies as a function of sample points for problems of $\dim(\mathcal{X} \times \mathcal{U}) = 4, 6, 8$.

Samples	ε_k	$\ V_{\text{ADP}} - V_{\text{LQG}}\ $	Constr. time (sec)	LP time (sec)	Memory
400	1	0.283	2.2	3.565	1.6MB
4000	0.25	0.206	17	97	16MB
40000	0.025	0.0360	170	162	160MB

TABLE III: Accuracy and computational requirements as a function of sample number for $\dim(\mathcal{X} \times \mathcal{U}) = 6$, with the parameters of Table I.

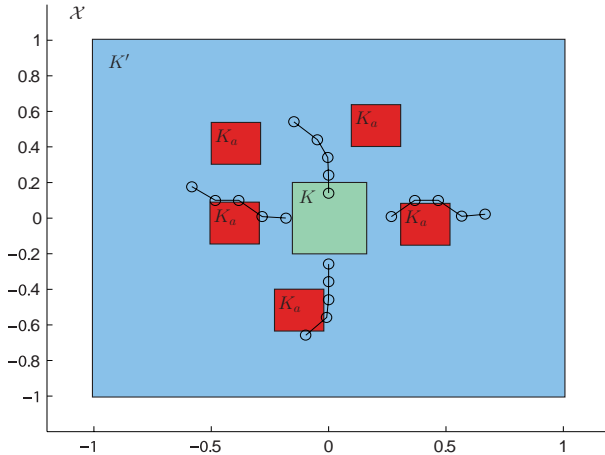


Fig. 7: Non-convex state constraints

$\dim(\mathcal{X} \times \mathcal{U})$	4D	6D	8D
Basis elements	100	500	1000
N_k	3960	19960	39960
$\ \tilde{V}_0 - V_{\text{ADP}}\ $	0.095	0.118	0.191
Constr. time (sec)	4.2	130	671
LP time (sec)	3.2	80	700
Memory	3.2MB	80MB	320MB

TABLE IV: Parameters and properties of the constructed value function approximation for obstacle avoidance problems of $\dim(\mathcal{X} \times \mathcal{U}) = 4, 6, 8$.

recorded empirical success probabilities for both policies (denoted by V_{MiQP} and V_{ADP} respectively) with respect to the reach-avoid objective. Table IV reports the values of all design parameters used in the value function approximation for problems of $\dim(\mathcal{X} \times \mathcal{U}) = 4, 6, 8$. The mean differences $\|\tilde{V}_0^* - V_{\text{ADP}}\|$ and $\|V_{\text{ADP}} - V_{\text{MiQP}}\|$ are computed by averaging the empirical reach-avoid success probabilities over the initial conditions. Figure 8 and Table V indicate a similar trade-off between accuracy and complexity as observed in Section V-A.

C. Race car cornering

Consider the problem of driving a race car through a tight corner in the presence of static obstacles, illustrated in Figure 9. As part of the ORCA project of the Automatic Control Lab (see <http://control.>

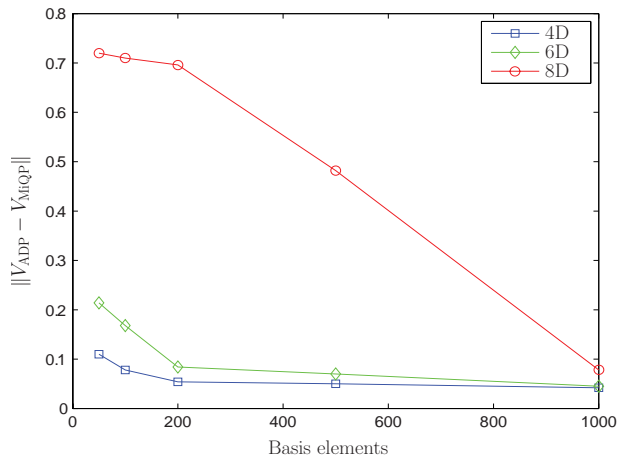


Fig. 8: Mean absolute difference between the empirical reach-avoid probabilities achieved by the ADP and MiQP policies as a function of approximation basis elements for problems of $\dim(\mathcal{X} \times \mathcal{U}) = 4, 6, 8$.

Basis elements	$\ V_{\text{ADP}} - V_{\text{MiQP}}\ $	Constr. time (sec)	LP time (sec)	Memory
50	0.214	1.67	0.18	784KB
100	0.168	5.59	2.66	3.2MB
200	0.084	22	4.3	12.8MB
500	0.07	130	80	80MB
1000	0.045	507	1213	320MB

TABLE V: Accuracy and computational requirements as a function of basis elements for $\dim(\mathcal{X} \times \mathcal{U}) = 6$.

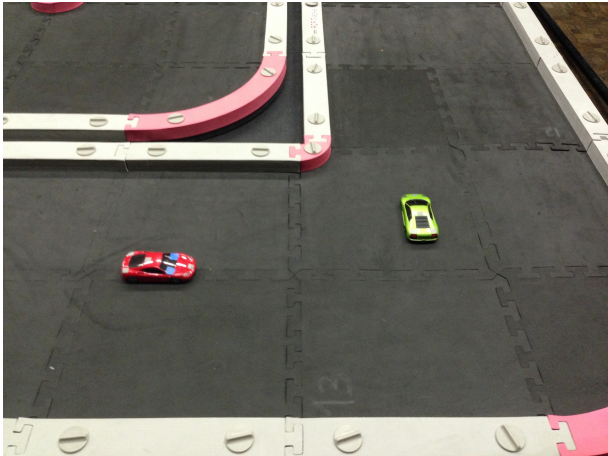


Fig. 9: Race-car cornering example

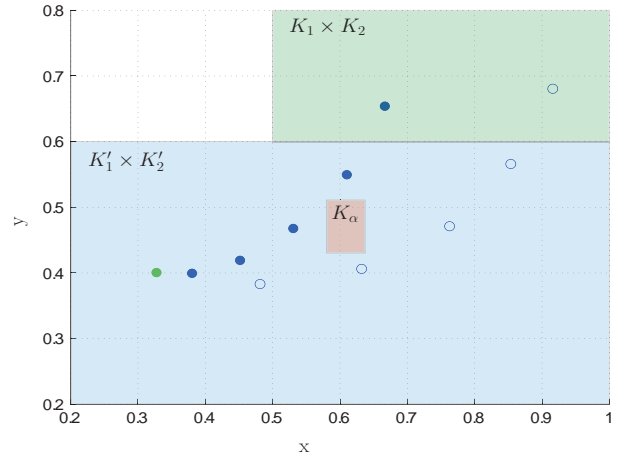


Fig. 10: Velocity dependent cornering trajectories

ee.ethz.ch/~racing/), a 6 state nonlinear model with 2 control inputs has been identified to describe the movement of 1:43 scale race cars. The model derivation is discussed in [54] 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.

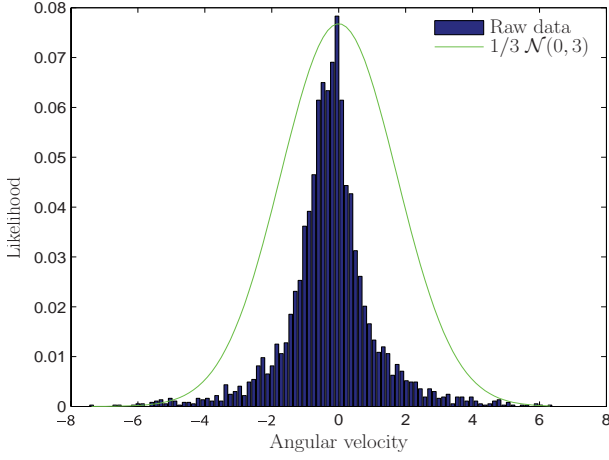


Fig. 11: Example of noise fit for the angular velocity

As is 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 and measurement noise. Analyzing the deviation between predictions and measurements, we identified a stochastic variant to the original model using additive Gaussian noise,

$$g(x, u) = f(x, u) + Gw, \quad w \sim \mathcal{N}(\mu, \Sigma). \quad (18)$$

The noise mean μ , diagonal covariance matrix Σ and diagonal scaling matrix G have been selected such that the distribution of measurements around the state predictions resembles the transition probabilities of the process in (18). Figure 11 illustrates the fit for the angular velocity where $\mu_6 = 0$, $\Sigma(6, 6) = 3$ and $G(6, 6) = 1/3$; the rest of the states are handled in the same way.

We cast the problem of driving the race car through a tight corner without reaching obstacles as a stochastic reach-avoid problem. We chose a horizon of $T = 5$ steps and a sampling time of 0.08 seconds to ensure that the model car can reach the target set. The setup is similar to the one in Section V-B with the crucial difference that the system is highly nonlinear and the control methods used before are not applicable without suitable linearization. On the contrary, the developed ADP technique can be readily applied to this problem once the safe sets, target sets and basis elements are defined.

The safe region on the spatial dimensions is defined as $(K'_1 \times K'_2) \setminus A$ where $A \subset \mathbb{R}^2$ denotes the obstacle set that has to be avoided by the moving car, see Figures 9, 10. The full safe set is then 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 coincide with the physical limitations of the model car (see Table VI). Similarly, the target region for the spatial dimensions is denoted by $K_1 \times K_2$ and corresponds to the end of the turn - Figure 10. The full target set is then defined as $K = K_1 \times K_2 \times K'_3 \times K'_4 \times K'_5 \times K'_6$ and contains all states $x \in K'$ for which $(x_1, x_2) \in K_1 \times K_2$. As in Sections V-A and V-B, the basis centers are placed randomly on $\bar{\mathcal{X}}$ and the associated variances are sampled from the intervals reported in Table VI. We used a total of 2000 basis elements and fixed all violation and confidence levels to $\varepsilon_k = 0.2$ and $1 - \beta_k = 0.99$ respectively for $k = \{0, \dots, 4\}$. The resulting reach-avoid DP recursions are 8-dimensional and the value function approximation is computed via Algorithm 1.

In order to verify our result we initialized the car at two different initial states

$$x^1 = (0.33, 0.4, -0.2, 0.5, 0, 0), \quad x^2 = (0.33, 0.4, -0.2, 2, 0, 0)$$

corresponding to entering the corner at low ($x_4^1 = 0.5$ m/s) and high ($x_4^2 = 2$ m/s) longitudinal velocities. The approximate value function values $\tilde{V}_0(x^1) = 0.98$, $\tilde{V}_0(x^2) = 1$ are both high but the associated trajectories computed via Algorithm 2 vary significantly. In the low velocity case, the car avoids the

Safe region	min	max	Basis Variance
K'_1 (m)	0.2	1	[0.0008, 0.0012]
K'_2 (m)	0.2	0.6	[0.0008, 0.0012]
K'_3 (rad)	$-\pi$	π	[0.005, 0.015]
K'_4 (m/s)	0.3	3.5	[0.005, 0.015]
K'_5 (m/s)	-1.5	1.5	[0.005, 0.015]
K'_6 (rad/s)	-8	8	[2, 4]

TABLE VI: Dimension safety limits and basis variances used in ADP approximation.

obstacle by driving above it while in the high velocity case, by driving below it; see Figure 10. Such a behavior is expected since the car model would slip if it turns aggressively at high velocities. We verified the performance of the control policy on the ORCA setup by running 10 experiments from each initial condition and sequentially applying the computed control inputs. Implementing the whole feedback policy online would require solving problems in the form of (15) within the sampling time of 0.08 seconds. This is conceptually possible since the control space is only two dimensional but requires developing an embedded nonlinear programming solver compatible with the ORCA setup. As demonstrated by the videos in (youtube:ETHZurichIfA), the car is successfully driving through the corner, avoiding the obstacle, even when the control inputs are applied in open loop.

VI. CONCLUSION

We developed and analyzed a value function approximation algorithm for the reach-avoid dynamic programming recursion using linear programming and randomized convex optimization. The method suggested is based on function approximation properties of Gaussian radial basis functions and exploits their structure to encode the transition kernel of Markov decision processes and compute integrals over the reach-avoid safe and target sets. The fact that our method relies on solving linear programs allows us to tackle reach-avoid problems with larger dimensions than state-space gridding methods. The accuracy and reliability of the approach is investigated by comparing to heuristics based on well-established methods in the control of linear systems. The potential of the approach is demonstrated by tackling a reach-avoid control problem for a six dimensional nonlinear system with two control inputs.

We are currently focusing on the problem of systematically choosing the center and variance parameters of basis function elements, exploiting knowledge about the dynamics of the considered systems. Apart from improving initial basis selection, we are looking into adaptive methods of choosing basis parameters according to the importance of states highlighted by the constraint sampling. In terms of computational efficiency, we are looking into decomposition methods for the large linear programming problems that arise in our approximation method to allow addressing reach-avoid problems of even higher dimensions. Finally, we are looking into tractable reformulations of the infinite constraints to avoid sampling based methods.

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., 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] 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.
- [5] Prandini, M., and Hu, J., 2006. “Stochastic reachability: Theory and numerical approximation”. *Stochastic hybrid systems, Automation and Control Engineering Series*, **24**, pp. 107–138.
- [6] Kushner, H. J., and Dupuis, P., 2001. *Numerical methods for stochastic control problems in continuous time*, Vol. 24. Springer.
- [7] Powell, W. B., 2007. *Approximate Dynamic Programming: Solving the curses of dimensionality*, Vol. 703. John Wiley & Sons.
- [8] Bertsekas, D., 2012. *Dynamic programming and optimal control*, Vol. 2. Athena Scientific Belmont, MA.
- [9] de Farias, D., and Van Roy, B., 2003. “The linear programming approach to approximate dynamic programming”. *Operations Research*, **51**(6), pp. 850–865.
- [10] Kariotoglou, N., Summers, S., Summers, T., Kamgarpour, M., and Lygeros, J., 2013. “Approximate dynamic programming for stochastic reachability”. In *IEEE European Control Conference*, pp. 584–589.
- [11] Calafiore, G., and Campi, M., 2006. “The scenario approach to robust control design”. *IEEE Transactions on Automatic Control*, **51**(5), pp. 742–753.
- [12] Summers, S., and Lygeros, J., 2010. “Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem”. *Automatica*, **46**(12), pp. 1951–1961.
- [13] Summers, S., 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.
- [14] Nowak, A. S., 1985. “Universally measurable strategies in zero-sum stochastic games”. *The Annals of Probability*, pp. 269–287.
- [15] Brown, L., Purves, R., et al., 1973. “Measurable selections of extrema”. *The annals of statistics*, **1**(5), pp. 902–912.
- [16] Bertsekas, D., 1976. “Dynamic programming and stochastic control”.

- [17] Sundaram, R. K., 1996. *A first course in optimization theory*. Cambridge university press.
- [18] Hernández-Lerma, O., and Lasserre, J. B., 1998. "Approximation schemes for infinite linear programs". *SIAM Journal on Optimization*, **8**(4), pp. 973–988.
- [19] Anderson, E. J., and Nash, P., 1987. *Linear programming in infinite-dimensional spaces: theory and applications*. Wiley New York.
- [20] Baglietto, M., Parisini, T., and Zoppoli, R., 2001. "Numerical solutions to the Witsenhausen counterexample by approximating networks". *IEEE Transactions on Automatic Control*, **46**(9), pp. 1471–1477.
- [21] Bertsimas, D., Brown, D. B., and Caramanis, C., 2011. "Theory and applications of robust optimization". *SIAM review*, **53**(3), pp. 464–501.
- [22] Ben-Tal, A., and Nemirovski, A., 2002. "Robust optimization—methodology and applications". *Mathematical Programming*, **92**(3), pp. 453–480.
- [23] Hettich, R., and Kortanek, K. O., 1993. "Semi-infinite programming: theory, methods, and applications". *SIAM review*, **35**(3), pp. 380–429.
- [24] Campi, M., and Garatti, S., 2008. "The exact feasibility of randomized solutions of uncertain convex programs". *SIAM Journal on Optimization*, **19**(3), pp. 1211–1230.
- [25] Mohajerin Esfahani, P., Sutter, T., and Lygeros, J., 2014. "Performance Bounds for the Scenario Approach and an Extension to a Class of Non-convex Programs". *IEEE Transactions on Automatic Control (to appear)*.
- [26] Sutter, T., Esfahani, P. M., and Lygeros, J., 2014. "Approximation of Constrained Average Cost Markov Control Processes". In IEEE Conference on Decision and Control.
- [27] Alamo, T., Tempo, R., and Luque, A., 2010. "On the sample complexity of probabilistic analysis and design methods". In *Perspectives in Mathematical System Theory, Control, and Signal Processing*. Springer, pp. 39–50.
- [28] Park, M., Darbha, S., Krishnamoorthy, K., Khargonekar, P., Pachter, M., and Chandler, P., 2012. "Sub-optimal stationary policies for a class of stochastic optimization problems arising in robotic surveillance applications". In ASME 2012 5th Annual Dynamic Systems and Control Conference joint with the JSME 2012 11th Motion and Vibration Conference, American Society of Mechanical Engineers, pp. 263–272.
- [29] Krishnamoorthy, K., Park, M., Darbha, S., Pachter, M., and Chandler, P., 2013. "Approximate dynamic programming applied to uav perimeter patrol". In *Recent Advances in Research on Unmanned Aerial Vehicles*. Springer, pp. 119–146.
- [30] Parrilo, P. A., 2003. "Semidefinite programming relaxations for semialgebraic problems". *Mathematical programming*, **96**(2), pp. 293–320.
- [31] Lasserre, J. B., 2001. "Global optimization with polynomials and the problem of moments". *SIAM Journal on Optimization*, **11**(3), pp. 796–817.
- [32] Wang, Y., O'Donoghue, B., and Boyd, S., 2014. "Approximate dynamic programming via iterated Bellman inequalities". *International Journal of Robust and Nonlinear Control*.
- [33] Summers, T., Kunz, K., Kariotoglou, N., Kamgarpour, M., Summers, S., and Lygeros, J., 2013. "Approximate dynamic programming via sum of squares programming". *IEEE European Control Conference*.
- [34] Lasserre, J. B., Henrion, D., Prieur, C., and Trélat, E., 2008. "Nonlinear optimal control via occupation measures and LMI-relaxations". *SIAM Journal on Control and Optimization*, **47**(4), pp. 1643–1666.
- [35] 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.
- [36] Sandberg, I. W., 2001. "Gaussian radial basis functions and inner product spaces". *Circuits, Systems and Signal Processing*, **20**(6), pp. 635–642.
- [37] Park, J., and Sandberg, I., 1991. "Universal approximation using radial-basis-function networks". *Neural computation*, **3**(2), pp. 246–257.
- [38] Cybenko, G., 1989. "Approximation by superpositions of a sigmoidal function". *Mathematics of control, signals and systems*, **2**(4), pp. 303–314.
- [39] Bemporad, A., Filippi, C., and Torrisi, F. D., 2004. "Inner and outer approximations of polytopes using boxes". *Computational Geometry*, **27**(2), pp. 151–178.
- [40] Boyd, S. P., and Vandenberghe, L., 2004. *Convex optimization*. Cambridge university press.
- [41] Chen, X., 2007. "A new generalization of Chebyshev inequality for random vectors". *arXiv preprint arXiv:0707.0805*.
- [42] Van Parys, B. P., Goulart, P. J., and Kuhn, D., 2014. "Generalized Gauss inequalities via semidefinite programming". *Mathematical Optimization*.
- [43] Chen, S., Cowan, C., and Grant, P., 1991. "Orthogonal least squares learning algorithm for radial basis function networks". *IEEE Transactions on Neural Networks*, **2**(2), pp. 302–309.
- [44] Wettschereck, D., and Dietterich, T., 1991. "Improving the performance of radial basis function networks by learning center locations". In NIPS, Vol. 4, Citeseer, pp. 1133–1140.
- [45] Waechter, A., Laird, C., Margot, F., and Kawajir, Y., 2009. Introduction to IPOPT: A tutorial for downloading, installing, and using IPOPT.
- [46] Petretti, A., and Prandini, M., 2014. "An approximate linear programming solution to the probabilistic invariance problem for stochastic hybrid systems". In IEEE Conference on Decision and Control.
- [47] Deori, L., Giullioni, L., and Prandini, M., 2014. "Optimal building climate control: a solution based on nested dynamic programming and randomized optimization". In IEEE Conference on Decision and Control.
- [48] 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.
- [49] 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.

- [50] 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.
- [51] Patel, R. B., and Goulart, P. J., 2011. "Trajectory generation for aircraft avoidance maneuvers using online optimization". *Journal of guidance, control, and dynamics*, **34**(1), pp. 218–230.
- [52] Bemporad, A., and Morari, M., 1999. "Control of systems integrating logic, dynamics, and constraints". *Automatica*, **35**(3), pp. 407–427.
- [53] Rakovic, S., and Mayne, D., 2005. "Robust time optimal obstacle avoidance problem for constrained discrete time systems". In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, IEEE, pp. 981–986.
- [54] Liniger, A., Domahidi, A., and Morari, M., 2014. "Optimization-based autonomous racing of 1: 43 scale RC cars". *Optimal Control Applications and Methods*.