Approximate dynamic programming for stochastic reachability

Nikolaos Kariotoglou, Sean Summers, Tyler Summers, Maryam Kamgarpour and John Lygeros

Abstract— **In this work we illustrate how approximate dynamic programing can be utilized to address problems of stochastic reachability in infinite state and control spaces. In particular we focus on the reach-avoid problem and approximate the value function on a linear combination of radial basis functions. In this way we get significant computational advantages with which we obtain tractable solutions to problems that cannot be solved via generic space gridding due to the curse of dimensionality. Numerical simulations indicate that control policies coming as a result of approximating the value function of stochastic reachability problems achieve close to optimal performance.**

I. Introduction

The main reason stochastic optimal control problems have been of particular interest to a wide range of researchers is the inherent uncertainty of real-world dynamical systems that give rise to various stochastic formulations. Among these formulations, we will be concerned with a version of Markov decision processes (MDPs) which can be related to discrete time stochastic hybrid systems (DTSHS) as shown in [1] and [2]. For discrete time MDPs, several stage cost optimal control problems (e.g discounted cost, average cost, etc) have been shown to be solved via dynamic programming (DP) as in [3], [4], [5]. Furthermore, due to the infamous curse of dimensionality, approximation techniques have been proposed to solve DPs. A well studied approach is that of linear programming [6], [7], [8] where the goal is to approximate the value function of these problems on a space spanned by a predefined set of basis functions.

The situation is somewhat different for problems of stochastic reachability [1], [2] where dynamic programming formulations have been shown to exist but without a systematic way to efficiently approximate the solution via linear programming. Stochastic reachability problems such as safety, reach-avoid and target hitting are of particular interest due to the interpretation of their value function as the probability of achieving the underlying objectives. For the purpose of model checking and verification [9] these value functions are used to directly infer system properties.

We restrict ourselves to the reach-avoid problem and illustrate that close approximations can be achieved in the case of finite horizon problems defined on compact but infinite control and state spaces. This is achieved by choosing a special class of basis functions to approximate the value function and stochastic kernel of the problem. Later on, the randomized sampling technique of [10], [11] is used to handle infinite dimensional constraints. An alternative method for handling the infinite constraint set using polynomial basis functions and sum of squares techniques is explored in a companion paper [12].

The paper is organized as follows: In the first section we introduce the specific reach-avoid stochastic reachability problem that we will deal with. Using previous results, we illustrate how this can be solved by a dynamic program and equivalently a linear program which is infinite dimensional in both decision variables and constraints. When the dimension of the product space of state and control spaces is low, standard (grid based) dynamic programming techniques can be used to obtain solutions. The limitations are quite strict, due to the curse of dimensionality, highlighting the need of approximation methods.

In the second part we introduce a specific basis function that for a certain class of systems transforms the given infinite problem to one with finite decision variables but still infinite constraints; we call this the semi infinite LP. Moreover, due to properties of the chosen basis, integral calculations in high dimensions admit an analytic solution (over hyperrectangles) making them computationally efficient. To tackle the infinite constraints we follow a scenario based approach which provides a lower bound on the sample number needed to satisfy the constraints probabilistically, with a certain confidence [10], [11]. Finally, using the idea of sampling again, we synthesize an approximate control policy and evaluate the performance of the proposed methodology both on the approximation of the optimal value function and the resulting control policy.

II. Stochastic reachability

The basic framework for stochastic reachability in the context of discrete time stochastic hybrid systems (DTSHS) is presented in [1], [2]. For the purpose of this work we will focus on the reach-avoid problem for Markov decision processes (MDPs) which are an equivalent description for DTSHSs. More precisely 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 depending on the current state and control action. The state space $\mathcal{X} \subseteq \mathbb{R}^n$ and control space $\mathcal{U} \subseteq \mathbb{R}^m$ are both infinite.

Consider a safe set $K' \in \mathcal{B}(\mathcal{X})$ and a target set $K \subseteq K'$ where $\mathcal{B}(\mathcal{X})$ denotes the Borel σ-algebra of $\mathcal{X}$. The reach-avoid problem over a finite time horizon N is to maximize the probability of x_t hitting the set K at some time $t_K \leq N$ while staying in K' for all $t \leq t_K$. We denote the control policy associated with this probability $\mu = \{\mu_0, \dots, \mu_{N-1}\}$ with $\mu_i : \mathcal{X} \to \mathcal{U}, i \in \{0, \dots, N-1\}$ and for an initial state x_0 we denote it as: $r^{\mu}_{x_0}(K, K') := \mathbb{P}^{\mu}_{x_0}\{\exists j \in [0, N] : x_j \in$

This research was partially supported by the European Commission under project MoVeS (FP7-ICT-2009-5) - Grant number 257005. The work of Nikolaos Kariotoglou was supported by SNF grant number 2000211_37876

$K \wedge \forall i \in [0, j-1], \quad x_i \in K' \setminus K\}$. Therefore, for a given time indexed policy μ, we are looking at the probability that x_j hits K before $K' \setminus K$ for some time $j \leq N$. In [2], $r^{\mu}_{x_0}(K, K')$ is shown to be equivalent to the expected value of the following sum multiplicative cost function:

$$r^{\mu}_{x_0}(K, K') = \mathbb{E}^{\mu}_{x_0}\left[\sum_{j=0}^{N}\left(\prod_{i=0}^{j-1} \mathbb{1}_{K' \setminus K}(x_i)\right)\mathbb{1}_K(x_j)\right] \tag{1}$$

and the task is to select a state feedback control policy $\mu(x)$ such that (1) is maximized. Note that to calculate μ that achieves this maximum (μ^*), one needs to calculate $\{\mu^*_0, \ldots, \mu^*_{N-1}\}$. For the rest of this work we drop the notation of time since it is clear from context and denote x_t, u_t simply as x, u.

A. Solution via dynamic programming

In [2], the solution to this problem is shown to be given by a dynamic recursion involving the value functions $V^*_k : \mathcal{X} \to [0, 1]$ for $k = 0, 1, \ldots, N-1$:

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

where Q is the process stochastic kernel describing the evolution of x_t and $Z : \mathcal{X} \to [0, 1]$. The optimal control policy for a state $x \in K' \setminus K$ is then given by:

$$\mu^*_k(x) = \arg\sup_{u \in \mathcal{U}} \left\{\mathbb{1}_K(x) + \mathbb{1}_{K' \setminus K}(x) H(x, u, V^*_{k+1})\right\}. \tag{3}$$

The value of the above recursion at $k = 0$ and for any initial state x_0 is the solution to (1), $V^*_0(x_0) = \sup_\mu r^{\mu}_{x_0}$.

B. Reach avoid expressed as an infinite LP

Solving problem (2) requires gridding the state and control spaces and working backwards from the known value function $V^*_N(x)$ at $k = N$. In this way, the value of $V^*_0(x)$ can be calculated on the grid points of $\mathcal{X}$ while the optimal control policy at each step k is calculated and stored when evaluating the supremum over the grid of $\mathcal{U}$. This is a computationally expensive process and as the size of the state and control spaces grows, it becomes intractable.

We define the following operators for any function $V : \mathcal{X} \to [0, 1]$:

$$\begin{aligned} \mathcal{T}_u[V](x) &:= \mathbb{1}_K(x) + \mathbb{1}_{K' \setminus K}(x) H(x, u, V) \\ \mathcal{T}[V](x) &:= \sup_{u \in \mathcal{U}} T_u[V](x) \end{aligned} \tag{4}$$

and use them to express problem (2) as an infinite LP, the solution of which we will approximate in the following section.

Proposition 1. *For each $k \in \{0, \ldots, N-1\}$, $V^*_k(x)$ is the solution to the following optimization problem:*

$$\begin{aligned} \min_{V_k(\cdot)} \quad & \int_{\mathcal{X}} V_k(x)\, \mathrm{dx} \\ \textit{s.t} \quad & V_k(x) \geq \mathcal{T}[V^*_{k+1}](x) \quad \forall x \in \mathcal{X} \end{aligned} \tag{5}$$

Proof. Let $J^*(x)$ be the solution to the above optimization problem and assume $\exists A_c \in \mathcal{B}(\mathcal{X})$ (with non-zero measure) such that $\forall x' \in A_c, J^*(x') \neq V^*_k(x')$. Since $J^*(x')$ is feasible, it must be that $J^*(x') > V^*_k(x')$. Then decreasing $J^*(x')$ on A_c until $J^*(x') = V^*_k(x)$ decreases $\int_{\mathcal{X}} J^*(x)\, \mathrm{dx}$. With this we conclude that $J^*(x)$ could not be optimal unless $J^*(x) = V^*_k(x)$, $\forall x \in A_c$. □

The problem stated in Proposition 1 is equivalent to the following one, using the fact that $V_k(x) \geq \mathcal{T}[V^*_{k+1}](x), \forall x \in \mathcal{X} \iff V_k(x) \geq \mathcal{T}_u[V^*_{k+1}](x), \forall (x, u) \in \mathcal{X} \times \mathcal{U}$:

$$\begin{aligned} \min_{V_k(\cdot)} \quad & \int_{\mathcal{X}} V_k(x)\, \mathrm{dx} \\ \text{s.t} \quad & V_k(x) \geq \mathcal{T}_u[V^*_{k+1}](x) \quad \forall (x, u) \in \mathcal{X} \times \mathcal{U}. \end{aligned} \tag{6}$$

This is a linear optimization problem (linear cost and constraints) over the infinite space of functions $V_k(\cdot)$ and with an infinite number of constraints. To find a solution we rely on approximation methods to make both decision variables and constraints finite. In the following sections we restrict ourselves to a given basis function set and illustrate how (6) can be transformed to obtain tractable (approximate) solutions. As a first step we will transform (6) by replacing the function $V_k(x)$ with its representation $\hat{V}_k(x)$ on a given finite set of M basis functions $\phi : \mathcal{X} \to \mathbb{R}$ for each $k \in \{0, \ldots, N-1\}$, such that:

$$V_k(x) \approx \hat{V}_k(x) = \sum_{i=1}^{M} w_i \phi(x) \tag{7}$$

with appropriately selected weights w_i. Note that since $\hat{V}_k(x)$ is linear on the basis weights w_i, any such approximation can be written as an infinite dimensional linear program once the functions $\phi(x)$ are fixed. We now discuss the choice of a particular type of function ϕ.

III. Radial basis functions

We choose the Gaussian function, which is a type of radial basis function (RBF) and the sum of RBFs is known to be a universal approximator for any continuous function on a compact subspace of $\mathbb{R}^n$. In particular, the sum of RBFs forms a neural network that with sufficiently many hidden layers (basis elements) allows universal approximation [13]. The parametrized radial basis functions $\phi : \mathbb{R}^n \to \mathbb{R}$ considered here are:

$$\phi(x; c, \nu) := \prod_{i=1}^{n} \psi(x_i; c_i, \nu_i). \tag{8}$$

with elements $x_i, c_i, \nu_i \in \mathbb{R}$, $i \in \{1, \dots, n\}$ and $x, c, \nu \in \mathbb{R}^n$. The functions $\psi : \mathbb{R} \to \mathbb{R}$ are Gaussian functions defined as:

$$\psi(x_i; c_i, \nu_i) := \frac{1}{\sqrt{2\nu_i \pi}} e^{-\frac{1}{2}\frac{(x_i - c_i)^2}{\nu_i}}.$$

which are parametrized by the mean c_i and variance ν_i, $i \in \{1, \dots, n\}$. A neural network is then constructed by taking a finite sum of such functions ϕ as in (7)

A. *Multi-dimensional integration*

A property of the chosen RBFs is that they admit an analytic expression to multi-dimensional integration over hyperrectangles, involving the well-studied error function erf. More precisely, for the integral of a function $V(x)$ approximated by $\hat{V}(x)$ on a n-dimensional hyperrectangle $AB = [a_1, b_1] \times \cdots \times [a_n, b_n]$, it holds:

$$\begin{aligned}\int_{AB} \hat{V}(x)\,\mathrm{dx} &= \sum_{i=1}^{M} w_i \prod_{k=1}^{n} \int_{a_k}^{b_k} \psi(x_j, c_{i,j}, \nu_{i,j}) \\ &= \sum_{i=1}^{M} w_i \prod_{k=1}^{n} -\frac{1}{2} \operatorname{erf}\left(\frac{x_k - c_{i,k} - b_k}{\sqrt{2\nu_{i,k}}}\right) \\ &\quad + \frac{1}{2} \operatorname{erf}\left(\frac{x_k - c_{i,k} - a_k}{\sqrt{2\nu_{i,k}}}\right)\end{aligned}$$

since on a compact interval $[a, b]$:

$$\begin{aligned}\int_a^b \frac{1}{\sqrt{2\nu\pi}} e^{-\frac{1}{2}\frac{(x-c)^2}{\nu}}\,\mathrm{dx} &= -\operatorname{erf}(\frac{x - c - b}{\sqrt{\nu}}) \\ &\quad + \operatorname{erf}(\frac{x - c - a}{\sqrt{\nu}})\end{aligned}$$

In the following sections we have implicitly assumed that the stochastic kernel Q is restricted to the set of models that can be expressed as a summation of RBFs. As a result, $H(x, u, \hat{V})$ reduces to an analytical expression similar to the one given above for the integral of $\hat{V}(x)$. The result is particularly useful for problems where integration over high dimensions is carried out repeatedly. In our attempt to address stochastic reachability problems with approximate dynamic programming it eliminates the need for Monte Carlo integration which otherwise occupies most of the computation time. In the approximation of the value function, an integral over the dimension of $\mathcal{X}$ is evaluated for every pair (x, u). Even if both spaces are finite, the computation time required for evaluating the integral is enough to make non-trivial problems intractable.

B. *Semi infinite LP*

The approximation of $V(x) \approx \hat{V}(x)$ in (7) is linear in the weights w_i which leads to the reformulation of problem (6) for $k \in \{0, \dots, N-1\}$ to a semi-infinite LP with a finite number of decision variables and an infinite number of constraints:

$$\begin{aligned}\min_{w_1, \dots, w_M} \quad & \int_{\mathcal{X}} \hat{V}_k(x)\,\mathrm{dx} \\ \text{s.t} \quad & \hat{V}_k(x) \geq \mathcal{T}_u[\hat{V}^*_{k+1}], \quad \forall (x, u) \in \mathcal{X} \times \mathcal{U} \\ & \hat{V}_k(x) = \sum_{i=1}^{M} w_i \prod_{j=1}^{n} \psi(x_j; c_{i,j}, \nu_{i,j})\end{aligned} \tag{9}$$

The above results allow the reach-avoid problem described in the previous section to be approximately solved. It is well known that current methods utilize space gridding and are limited due to the curse of dimensionality [14] and the exponential growth in storage and computation requirements with respect to the dimension of $\mathcal{X} \times \mathcal{U}$. Being able to reduce this by an order of magnitude is a significant improvement.

IV. Approximate dynamic programming

The solution to the semi-infinite LP reach-avoid problem is a projection of the optimal value function onto the span of the proposed RBF mixture. It has been impossible so far however to state conditions on the system's transition kernel Q and the reachability sets K, K' for finding particular basis such that the value function belongs in the span. Hence, the solution to the above LP is only approximate. However, along the lines of [6] we make the following proposition:

Proposition 2. *The solutions for each $k \in \{1, \dots, N-1\}$ to problem (9) minimize the following norm:*

$$\|\hat{V}^*_k(x) - V^*_k(x)\|_1 = \int_{\mathcal{X}} |\hat{V}^*_k(x) - V^*_k(x)|\,\mathrm{dx}$$

over the same constraints $\hat{V}_k(x) \geq \mathcal{T}_u[\hat{V}^*_{k+1}]$. To prove Proposition 2 we will make use of the following claim:

Claim 1. *For every $k \in \{1, \dots, N\}$ it holds that $\hat{V}^*_k(x) \geq V^*_k(x)$ where $V^*_k(x), \hat{V}^*_k(x)$ are the solutions to problems (2),(9) respectively.*

Proof. By induction. First of all we assume that $\hat{V}^*_N(x) = V^*_N(x)$. For $k = N-1$, due to feasibility, it holds that $\hat{V}^*_{N-1}(x) \geq \mathcal{T}[\hat{V}^*_N](x) = \mathcal{T}[V^*_N](x) = V^*_{N-1}(x)$. Assuming that for $k+1$ it holds that $\hat{V}^*_{k+1}(x) \geq V^*_{k+1}(x)$ we show that $\hat{V}^*_k(x) \geq V^*_k(x)$.

$$\hat{V}^*_{k+1} \geq V^*_{k+1} \Rightarrow \mathcal{T}[\hat{V}^*_{k+1}] \geq \mathcal{T}[V^*_{k+1}] \Rightarrow$$
$$\hat{V}^*_k \geq \mathcal{T}[\hat{V}^*_{k+1}] \geq \mathcal{T}[V^*_{k+1}] = V^*_k(x) \Rightarrow \hat{V}^*_k \geq V^*_k.$$

□

Proof. According to Claim 1, $\hat{V}^*_k(x) \geq V^*_k(x)$, $\forall x \in \mathcal{X}$ hence:

$$\begin{aligned}\|\hat{V}^*_k(x) - V^*_k(x)\|_1 &= \int_{\mathcal{X}} |\hat{V}^*_k(x) - V^*_k(x)|\,\mathrm{dx} \\ = \int_{\mathcal{X}} \hat{V}^*_k(x) - V^*_k(x)\,\mathrm{dx} &= \int_{\mathcal{X}} \hat{V}^*_k(x)\,\mathrm{dx} - \int_{\mathcal{X}} V^*_k(x)\,\mathrm{dx}\end{aligned}$$

which implies that minimizing $\|\hat{V}^*_k(x) - V^*_k(x)\|_1$ is equivalent to minimizing $\int_{\mathcal{X}} \hat{V}^*_k(x)\,\mathrm{dx}$. □

On top of the approximation introduced here we also have to approximate the infinite number of constraints. We will use a well known result by [11], [10], to convert these to a chance constraint with a certificate of confidence on its feasibility. More precisely, randomized sampling will be used to solve the following epigraph reformulation of (9):

$$\begin{aligned} \min_{w_1,\dots,w_M,\gamma} \quad & \gamma \\ \text{s.t} \quad & \mathbb{P}\left\{(x,u)\in\mathcal{X}\times\mathcal{U}\,\middle|\,\hat{V}_k(x) < \mathcal{T}_u[\hat{V}^*_{k+1}]\right\} < \epsilon \\ & \int_{\mathcal{X}} \hat{V}_k(x) \le \gamma, \hat{V}_k(x) = \sum_{i=1}^{M} w_i \prod_{j=1}^{n} \psi(x_j, c_{i,j}, \nu_{i,j}) \end{aligned}$$

Intuitively, as explained by [11], the infinite constraint is now required to be satisfied on all but a fraction ϵ of the space $\mathcal{X}\times\mathcal{U}$ with probability $1-\beta$. To achieve this, the authors prove that the sample size must be at least $N_s \geq 2\epsilon\left(\ln(1/\beta)+d\right)$. To deal with the chance constraint, we follow a scenario based approach. The fraction $\epsilon\in(0,1)$ and confidence $\beta\in(0,1)$ are design choices while $d=\dim(\mathcal{X}\times\mathcal{U})$ refers to the number of the decision variables in the optimization problem. Note that in the version of the problem considered here, the space $\mathcal{X}\times\mathcal{U}$ is treated as an uncertainty set and the results hold irrespective of the probability distribution over $\mathcal{X}\times\mathcal{U}$ used to draw the samples. Drawing N_s samples and solving the following LP gives for each $k\in\{1,\dots N-1\}$ a $\hat{V}^*_k(x)$ that satisfies all but an ϵ fraction of the constraints with a confidence $1-\beta$.

$$\begin{aligned} \min_{w_1,\dots,w_M} \quad & \int_{\mathcal{X}} \hat{V}_k(x)\,\mathrm{dx} \\ \text{s.t} \quad & \hat{V}_k(x^s) \geq \mathcal{T}_{u^s}[\hat{V}^*_{k+1}](x^s), \quad \forall s\in\{1,\dots,N_s\} \\ & \hat{V}_k(x^s) = \sum_{i=1}^{M} w_i \prod_{j=1}^{n} \psi(x^s_j, c_{i,j}, \nu_{i,j}) \end{aligned} \tag{10}$$

The pairs (x^s,u^s) denote the s^{th} sample drawn from $\mathcal{X}\times\mathcal{U}$ and lead to a super-optimal solution without guarantees regarding the super-optimality gain. Consequently, it is no longer guaranteed that $\hat{V}^*_k(x)\geq V^*_k(x)$, $\forall x\in\mathcal{X}$.

A. Impact on reachability

The number N_s corresponds to the number of evaluations of $\mathcal{T}_u[V^*_{k+1}](x)$ needed to achieve the desired confidence. The use of RBFs of form (8) allows the analytic calculation of the integral involved in the introduced operators. The total computational time is therefore reduced significantly for each constraint evaluation. As a consequence, the resulting approximation not only provides a solution when gridding methods fail, but in cases where they don't, reduces calculations requiring hours to ones requiring minutes - See Table I for an example. Finally, as illustrated in section V, the (optimal) approximate controller achieves close to optimal performance.

B. Optimal control policies

The optimal control policy $\mu^*_k(x)$ which comes as a solution to (3) for a given value function $V^*_k(x)$ and initial state x is different from the solution of the following problem where the value function approximation at $k+1$ is used:

$$\hat{\mu}^*_k(x) = \arg\sup_{u\in\mathcal{U}}\{\mathbb{1}_K(x) + \mathbb{1}_{K'\setminus K}(x)H(x,u,\hat{V}^*_{k+1})\} \tag{11}$$

We could not obtain an analytic solution to the problem of (11) even though we have an analytic expression for $\hat{V}^*_{k+1}(x)$. We will employ randomized sampling once again and solve the following optimization problem instead:

$$\begin{aligned} \min_{\gamma} \quad & \gamma \\ \text{s.t} \quad & \mathbb{P}\left\{u\in\mathcal{U}\,\middle|\,\mathcal{T}_u[\hat{V}^*_{k+1}](x) > \gamma\right\} < \epsilon \end{aligned} \tag{12}$$

As stated previously, the solution comes with a certain probabilistic confidence by drawing N_s samples from $\mathcal{U}$. Notice that since x is fixed, we now only sample the control space thus reducing the number of samples needed during the on-line computation of the optimal approximate control policy. Also note that $u^*(x)$ is obtained by looking at the dual variables of (12) since γ is a scalar and there will be only 1 support constraint.

The algorithms for approximating the value function and the associated control policy are summarized below.

Algorithm 1 Recursive value function approximation

```
Randomly place $\{c_1,\dots,c_M\}$ basis centers on $K'\setminus K$
Choose $\{\nu_1,\dots,\nu_M\}$ basis variances
Initialize $\hat{V}^*_N(x) \leftarrow V^*_N(x)$
for all $n\in\{N-1,\dots,1\}$ do
    Sample $N_s$ pairs $(x^s,u^s)$ uniformly from $\mathcal{X}\times\mathcal{U}$
    for all $(x^s,u^s)$ do
        Evaluate $b(x^s,u^s) = \mathcal{T}_{u^s}[\hat{V}^*_{n+1}](x^s)$
    end for
    Solve the LP from (10)
end for
```

Algorithm 2 Approximate control policy

```
Measure the system state $x\in K'$
Sample $N_s$ points $u^s$ uniformly from $\mathcal{U}$
for all $u^s$ do
    Evaluate $p(u^s) = \mathcal{T}_{u^s}[\hat{V}^*_1](x)$
end for
Calculate $\hat{\mu}^*(x) = \arg\max_{u_s} p(u^s)$
Apply $\hat{\mu}^*(x)$ to the system.
```

V. Simulation results

A. Gridded vs approximate value function

To illustrate the performance of the proposed method on problems of stochastic reachability, we set up a reach-avoid

Method	Total time (min)	$\frac{\|V_0^*(x)-\hat{V}_0^*(x)\|_1}{\|V_0^*(x)\|_1}$
DP	748	0
ADP, $N_s = 500$	0.10	0.492
ADP, $N_s = 1000$	0.21	0.371
ADP, $N_s = 10000$	0.45	0.083
ADP, $N_s = 20000$	0.89	0.054
ADP, $N_s = 200000$	9.22	0.023

TABLE I

COMPUTATION TIME COMPARISON BETWEEN THE GRIDDED AND THE APPROXIMATE VALUE FUNCTION

problem for a 2D stochastic linear system with a 2D control input. Both control and state spaces are infinite and compact domains while the noise is a non-zero mean 2D Gaussian distribution with diagonal covariance matrix. Note that the DP solution for this problem requires gridding both the state and control spaces giving rise to a 4D grid. A transition probability matrix calculated for such a grid requires several hundred megabytes for storing and increasing any space to 3D will make the solution intractable. The dynamics chosen for the particular example are of the form $x_{k+1} = Ax_k + Bu_k + w, \quad w \sim \mathcal{N}(\mu, \Sigma)$.

The target set $K = \{x \in [-1,1]^2\}$ and safe set $K' = \{x \in [-7,7]^2\}$ are rectangular such that $K \subseteq K'$ and the time horizon is chosen as $N = 5$. The gridded solution for this problem is denoted as $V_0^*(x)$ and according to (2) requires $N-1$ recursion evaluations, given $V_N^*(x) = \mathbb{1}_K(x)$. Each recursion requires a numerical integral computed on the grid of $\mathcal{X}$ for each point of $\mathcal{U}$ in order to calculate $\sup_{u\in\mathcal{U}}$. Following the approximate dynamic programming method outlined in this work with the particular choice of RBFs, we can improve on the computation time required without sacrificing much accuracy - Table I. Moreover, we can obtain an explicit approximation for the value function in continuous space as opposed to the grid solution. In future work we will use this to further investigate properties of the optimization problem of the optimal control policy (11).

Using $M = 100$ basis centers randomly placed on $K' \setminus K$, we solve problem (10) by uniformly sampling the set $\mathcal{X} \times \mathcal{U}$, N_s times for a choice of $\epsilon = 0.01$ and $\beta = 0.01$. Fig 1 shows $\hat{V}_0^*(x) = \sum_{i=1}^M w_i^* \prod_{j=1}^d \psi(x_j; c_{i,j}, \nu_{i,j})$ for a constant variance of 8, color coded by the normalized 1-norm error with respect to $V_0^*(x)$. Fig. 2 shows the normalized 1-norm error between the DP value function and the approximate one $\|V_0^*(x) - \hat{V}_0^*(x)\|_1/\|V_0^*(x)\|_1$ as a function of constraint samples N_s. The final choice of trade-off between computational time and accuracy depends on the application. As an indication, consider that the particular $\hat{V}_0^*(x)$ was calculated more than 80 times faster than $V_0^*(x)$ and within accuracy of $\approx 2.5\%$.

B. Gridded vs approximate control policy

Here we compare the performance of the optimal control policy (3) against the approximate policy that comes as a solution to (12). To carry out this test, we uniformly sampled 100 initial states x_0^i from $K' \setminus K$, $i \in \{1, \ldots, 100\}$ and for each ran 1000 simulations of the t_K-step controls $\{\mu_0^*(x_0^i), \mu_1^*(x_1^i), \ldots, \mu_{t_K}^*(x_{t_K}^i)\}$ and $\{\hat{\mu}_0^*(x_0^i), \hat{\mu}_1^*(x_1^i), \ldots, \hat{\mu}_{t_K}^*(x_{t_K}^i)\}$. Note that the gridded policy is only defined on the grid of $\mathcal{X}$ while the approximate one can be calculated for any point $x \in \mathcal{X}$ without interpolation needed. The performance was measured according to how many trajectories of the 1000 simulations (all starting at one of the different x_0^i) managed to hit K within N steps, without hitting $X \setminus K'$ prior to that. Fig. 3 shows the mean absolute error over all x_0^i, of the success probabilities corresponding to $\{\mu_0^*(x_0^i), \mu_1^*(x_1^i), \ldots, \mu_{t_K}^*(x_{t_K}^i)\}$ and $\{\hat{\mu}_0^*(x_0^i), \hat{\mu}_1^*(x_1^i), \ldots, \hat{\mu}_{t_K}^*(x_{t_K}^i)\}$, as a function of sample points N_s.

Although several approximations are taking place in subsequent steps of the presented method, the quality of the final approximate control policy suggests a potential of further exploitations. The main drawback of the presented approach is the fact that an on-line optimization problem (12) needs to be solved at every horizon step, in order to retrieve the control policy. For the exact grid solution, the policy comes for free with the construction of the value function. However, as already mentioned, the calculations become prohibitive as the dimensions increase.

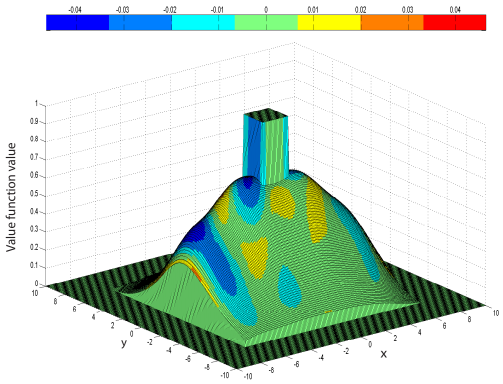

Fig. 1. Value function comparison. The approximate solution is plotted, color coded by the error $\hat{V}_0^*(x) - V_0^*(x)$.

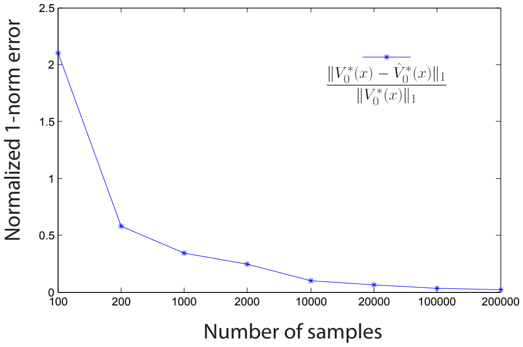

Fig. 2. Value function 1-norm error with respect to number of samples N_s.

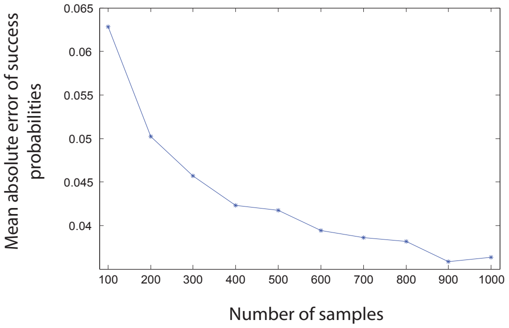

Fig. 3. Mean absolute error between exact and approximate control policies as a function of samples on the space $\mathcal{U}$

C. Approximate solution for a high dimensional problem

To illustrate that the proposed method can be applied on higher dimension problems, we considered a 3D stochastic linear system with a 3D control input and a 3D normally distributed noise with diagonal covariance matrix. The target and safe set were defined as $K = \{x \in [-1,1]^3\}$ and $K' = \{x \in [-7,7]^3\}$ respectively. In order to solve the reach-avoid problem using the exact DP (2) on a grid, one needs to traverse the 6D grid $\mathcal{X} \times \mathcal{U}$. This is a limitation of gridding methods due to the curse of dimensionality and we could not obtain a solution due to the storage requirements. Moreover, even if we could store the fine grid, the computation time required to calculate integrals would make the solution practically intractable. As there is no value function to compare with, we evaluated the performance based on the approximate control policy. We sampled 50 initial states x_0^i, $i \in \{1,\dots,50\}$ and for each ran 1000 simulations of the feedback t_K-step control $\{\hat{\mu}_0^*(x_0^i), \hat{\mu}_1^*(x_1^i), \dots, \hat{\mu}_{t_K}^*(x_{t_K}^i)\}$ recording how many reached K without leaving K'. In 90% of initial states the ADP value functon $\hat{V}_0^*(x)$ is a lower bound for the actual performance of the policy and the mean absolute error is $\approx 17\%$. The time taken to calculate $\hat{V}_0^*(x)$ for this problem was ≈ 10 min.

D. Short note on approximations

The approximations techniques used throughout this work enter in three separate subsequent steps. First, in (9) we approximate the value function in the class of RBFs. Not much is sacrificed by this since RBFs can accurately approximate any continuous function with a large enough number of centers. The approximation is carried out on the set $K' \setminus K \subseteq \mathbb{R}^n$ and the value function is continuous on it. The second approximation is introduced via randomized sampling for solving the semi-infinite LPs in (10). We chose a scenario based approach due to the probabilistic guarantees it provides, independently of the distribution used for sampling. Note however that we are still missing a strong bound on the approximation error in (10) if constraints of are not respected. There is hope however of achieving this by imposing certain conditions on the value function (e.g Lipschitz continuity). The final approximation introduced in (12) can be replaced by local non-convex optimization methods (e.g Gradient based methods). This technique has the potential of producing promising results and reducing the on-line computation requirements allowing real time implementations.

VI. Conclusion

In this work we outlined the method of applying approximate dynamic programming to a reach-avoid problem of stochastic reachability. We restricted the systems that we can deal with in the class of radial basis functions allowing analytic calculations of multi-dimensional integrals as well as analytic expressions for the recursive value function. Based on the results of [6], [15] we converted an infinite LP solving the reach-avoid problem to a finite one both in decision variables and constraints. Although we do get a probabilistic guarantee that the set of pairs $(x,u) \in \mathcal{X} \times \mathcal{U}$ that violate the constraints is small, we have no bound on how much they can violate them. Subsequent work is focusing on answering these questions both for the approximation of the optimal value function as well as the approximation of the optimal control policy.

References

[1] A. Abate, M. Prandini, J. Lygeros, and S. Sastry, "Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems," *Automatica*, vol. 44, no. 11, pp. 2724–2734, 2008.

[2] S. Summers and J. Lygeros, "Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem," *Automatica*, vol. 46, no. 12, pp. 1951–1961, 2010.

[3] C. Guestrin, M. Hauskrecht, and B. Kveton, "Solving factored mdps with continuous and discrete variables," in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pp. 235–242, AUAI Press, 2004.

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

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

[6] D. de Farias and B. Van Roy, "The linear programming approach to approximate dynamic programming," *Operations Research*, vol. 51, no. 6, pp. 850–865, 2003.

[7] M. Hauskrecht and B. Kveton, "Linear program approximations to factored continuous-state markov decision processes," *NIPS-17*, 2003.

[8] Y. Wang and S. Boyd, "Approximate dynamic programming via iterated bellman inequalities," 2010.

[9] F. Ramponi, D. Chatterjee, S. Summers, and J. Lygeros, "On the connections between pctl and dynamic programming," in *Proceedings of the 13th ACM international conference on Hybrid Systems: Computation and Control*, pp. 253–262, ACM, 2010.

[10] M. Campi and S. Garatti, "The exact feasibility of randomized solutions of uncertain convex programs," *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1211–1230, 2008.

[11] G. Calafiore and M. Campi, "The scenario approach to robust control design," *Automatic Control, IEEE Transactions on*, vol. 51, no. 5, pp. 742–753, 2006.

[12] T. Summers, K. Kunz, N. Kariotoglou, M. Kamgarpour, S. Summers, and J. Lygeros, "Approximate dynamic programming via sum of squares programming," *Submitted to the IEEE European Control Conference*, 2013.

[13] J. Park and I. Sandberg, "Universal approximation using radial-basis-function networks," *Neural computation*, vol. 3, no. 2, pp. 246–257, 1991.

[14] D. Bertsekas, *Dynamic programming and optimal control*, vol. 1. Athena Scientific Belmont, MA, 1995.

[15] M. Campi, S. Garatti, and M. Prandini, "The scenario approach for systems and control design," *Annual Reviews in Control*, vol. 33, no. 2, pp. 149–157, 2009.