A Semidefinite Programming Approach to Control Synthesis for Stochastic Reach-Avoid Problems (Tool Presentation)

Dalibor Držajić[1], Nikolaos Kariotoglou[1], Maryam Kamgarpour[1], and John Lygeros[1]

Automatic Control Laboratory, Department of Information Technology and Electrical Engineering, ETH Zürich, Zürich 8092, Switzerland
www.control.ee.ethz.ch

Abstract

We propose a computational approach to approximate the value function and control policies for a finite horizon stochastic reach-avoid problem as follows. First, we formulate an infinite dimensional linear program whose solution characterizes the optimal value function of the stochastic reach-avoid. Next, we introduce sum-of-squares polynomials to approximate the solution of this linear program through a semidefinite program. We compare our proposed tool to alternative numerical approaches via several case studies.

1 Introduction

Finite horizon reach-avoid problems are a special class of reachability problems in which the goal is to reach a given target set within a time horizon while avoiding an unsafe set [1]. It is known that reach-avoid problems for discrete time stochastic hybrid systems (DTSHS) can be cast as control problems for a Markov decision process (MDP) [2]. In this framework, the objective is to synthesize a control policy that maximizes the reach-avoid probability over the trajectory of the MDP. One can formulate this problem using a dynamic programming (DP) recursion by introducing indicator functions for the so-called target (reach) and unsafe (avoid) sets [2]. A straightforward way to numerically approximate the value function of the reach-avoid DP recursion is by constructing a grid for the continuous state and control spaces. Under suitable assumptions on the MDP kernel, gridding approaches provide performance guarantees on the resulting approximate solution [3, 4], but their complexity grows exponentially with the dimension of continuous states (curse of dimensionality) [3].

In an attempt to overcome the limitation of gridding based approximations, the authors in [9] used an optimization based approximation method for the reach-avoid value function and the optimal policy. Their approach was based on formulating the reach-avoid DP recursion as an infinite dimensional linear program (LP). Under suitable assumptions, the optimizer of this LP is the reach-avoid value function [6]. To numerically solve the infinite dimensional LP, they introduced Gaussian radial basis functions to restrict the decision space corresponding to the value function to a finite dimensional space and utilized randomized sampling to ensure the infinite constraints probabilistically [19]. Even though the approach in [9] pushes the limit of numerical solutions further than the state-of-the-art, it is still computationally intensive and the probabilistic upper bounds provided are elusive when used for verification purposes.

We consider the same infinite dimensional linear programming formulation [9] and introduce a new approximation scheme based on polynomial basis functions. The infinite constraints of the LP can then be equivalently formulated as polynomial non-negativity constraints on semi-algebraic sets, defined by the target and unsafe sets of the reach-avoid problem. We then use

a sufficient condition to reformulate polynomial non-negativity based on sum-of-squares (SOS) polynomials. Consequently, the problem of approximating the stochastic reach-avoid probability becomes equivalent to a semidefinite program (SDP). An advantage of the formulation is that we provide a guaranteed upper bound (as opposed to the probabilistic ones obtained in [9, 10]) for the approximated value function of the reach-avoid problem. In order to further investigate the scalability of our proposed method, we use recent results based on alternative relaxations of SOS polynomials involving diagonally dominant sum-of-squares (DSOS) and scaled diagonally dominant-sum-of-squares (SDSOS) polynomials [16, 17].

The paper is organized as follows. In Section 2 we review the basics of stochastic reach-avoid problem for discrete time stochastic hybrid control systems. We define the value function of the reach-avoid problem through a DP recursion and formulate an equivalent infinite dimensional LP. In Section 3 we choose polynomial basis functions for the approximation of the infinite dimensional LP and reformulate the resulting semi-infinite LP as a semidefinite program through sum-of-squares polynomials. In Section 4 we evaluate the performance of the proposed approach in terms of accuracy and computational complexity with several numerical examples. In Section 5 we summarize our work and discuss future directions.

2 Stochastic Reach-Avoid Problem

Let $\mathbb{X} \subseteq \mathbb{R}^n$, $\mathbb{U} \subseteq \mathbb{R}^m$ and $\mathbb{W} \subseteq \mathbb{R}^p$ be respectively the state space, control space and uncertainty space of a stochastic system. Consider a Markov Decision Process (MDP) with dynamics described by $x_{k+1} = f(x_k, u_k, w_k)$ with $x_k \in \mathbb{X}$, $u_k \in \mathbb{U}$, $w_k \in \mathbb{W}$ and the discrete time index $k \in \mathbb{N}$. We assume the stochastic nosie w_k is independent identically distributed with Borel measurable stochastic kernel $\mathcal{Q}$. Let $\mathbb{K}' \subset \mathbb{X}$ be a given safe set and $\mathbb{K} \subseteq \mathbb{K}'$ be the target set that the system needs to reach within a finite time horizon $\mathbb{T} = \{0, \dots, T\}$. For notational convenience, introduce $\bar{\mathbb{X}} := \mathbb{K}' \setminus \mathbb{K}$. The sets $\mathbb{X}, \mathbb{U}, \mathbb{W}, \mathbb{K}, \mathbb{K}'$ are assumed to be Borel measurable. We consider deterministic Markov policies at each time step k, $\psi_k : \mathbb{X} \to \mathbb{U}$. The reach-avoid problem for the MDP is defined as follows.

Definition 1 (Reach-avoid problem)**.** *Given a time horizon $\mathbb{T}$, find the policies $\psi^* = (\psi_0^*, \dots, \psi_{T-1}^*)$ such that starting from any initial state $x_0 \in \bar{\mathbb{X}}$, the probability to hit the target set $\mathbb{K}$ within time horizon $\mathbb{T}$ while staying inside the safe set $\mathbb{K}'$ at all prior times is maximized.*

The value function $V_k^* : \mathbb{X} \to \mathbb{R}$ of the reach-avoid problem at every time $k \in \mathbb{T}$ can be characterized by a dynamic programming (DP) recursion [2] of the form:

$$\begin{aligned} V_T^*(x_T) &= \mathbb{1}_{\mathbb{K}}(x_T), \\ V_k^*(x_k) &= \sup_{u_k \in \mathbb{U}} \left\{ \mathbb{1}_{\mathbb{K}}(x_k) + \mathbb{1}_{\mathbb{K}' \setminus \mathbb{K}}(x_k) \cdot \mathbb{E}_w[V_{k+1}^*(f(x_k, u_k, w_k))] \right\}, \end{aligned} \tag{1}$$

where $\mathbb{1}_A(x) = 1$ whenever $x \in A$ and $\mathbb{1}_A(x) = 0$ otherwise, for $A \subset \mathbb{X}$. At every state x_k, the function $V_k^*(x_k)$ returns the maximum probability of reaching the set $\mathbb{K}$ within the time interval $[k, T]$ [2]. Moreover, using the functions V_k^* one can define the optimal control policy at each time $k \in \mathbb{T}$ as

$$\psi_k^*(x) = \arg\max_{u_k \in \mathbb{U}} \left\{ \mathbb{1}_{\mathbb{K}}(x_k) + \mathbb{1}_{\mathbb{K}' \setminus \mathbb{K}}(x_k) \cdot \mathbb{E}_w[V_{k+1}^*(f(x_k, u_k, w_k))] \right\}. \tag{2}$$

The optimal policy ψ^*, over the horizon is then defined as $\psi^* = \{\psi_0^*, \dots, \psi_{T-1}^*\}$. We assume continuity of f in u and compactness of $\mathbb{U}$, to ensure that the optimal policy is attained. Furthermore, it follows that the value function is bounded and Borel measurable.

Let us define the Bellman operator $\mathcal{T}_u : \mathcal{F} \to \mathcal{F}$ as $\mathcal{T}_u[V](x) := \mathbb{1}_{\mathbb{K}}(x) + \mathbb{1}_{\mathbb{K}' \setminus \mathbb{K}}(x) \cdot \mathbb{E}_w[V(f(x,u,w))]$, where $\mathcal{F}$ is the space of real-valued bounded Borel measurable functions on $\mathbb{X}$. Let ς denote a probability measure on $\mathbb{X}$, interpreted as a state relevance weight. Given V^*_{k+1}, the value function V^*_k can be found (up to a set of ς-measure zero) as a solution of the following infinite dimensional linear program (LP) [5, 9, 11].

$$\begin{aligned} \min_{V(\cdot)\in\mathcal{F}} \quad & \int_{\mathbb{X}} V(x)\varsigma(dx) \\ \text{subject to} \quad & V(x) \geq \mathcal{T}_u[V^*_{k+1}](x), \quad \forall\ (x,u) \in \mathbb{X}\times\mathbb{U} \end{aligned} \tag{3}$$

Retrieving the solution of (3) is an NP-Hard problem [13], motivating the need to develop numerical approximation methods. In the following Section, we restrict the decision space $\mathcal{F}$ to polynomial basis functions and reformulate the infinite LP as a semi-infinite LP that can be addressed using well-known results from polynomial optimization research. We consequently obtain a pointwise upper bound on the reach-avoid value function. Once an approximate (upper-bounding) polynomial value function is found at each $k \in \mathbb{T}$, we define an associated approximate control policy by solving another polynomial optimization problem.

3 Approximated LP with Polynomials

Adopting polynomials to approximate the value function of stochastic control problems through semidefinite programming (SDP) has been discussed in [12]. Here, we apply this approach for the reach-avoid problem at hand. We start by assuming that the value function at each time step k is approximated as $V_k(x) \approx \hat{V}_k(x) = \sum_{i=1}^{N} c_{i,k} m_i(x)$ where $m_i(x)$ are $N = \binom{n+r}{r}$ monomials up to degree r, in n variables and $c_{i,k}$ are their respective coefficients. Using this form of polynomial basis, the infinite dimensional decision space in (3) is equivalently restricted to the space of polynomials of order r in n variables, $\hat{\mathcal{F}} = \mathbb{R}^r_n[x]$. The decision variable in the resulting optimization problem is the vector of N coefficient $c_{i,k} \in \mathbb{R}$, of the monomials.

In order to ensure the constraints in (3) can be formulated as a polynomial non-negativity constraint, we make the following assumptions.

Assumption 1. *The dynamics of the system $f(x,u,w)$ have a polynomial representation.*

Assumption 2. *The sets $\mathbb{X}, \mathbb{K}, \mathbb{K}', \mathbb{U}$ are semi-algebraic.*

Assumption 3. *Moments of the noise $\mathbb{E}[w^\alpha] = \int_{\mathbb{R}^p} w^\alpha \mathcal{Q}(w)dw$ and the state relevant weight ς up to a sufficient order (dependent on the order of the polynomial value function and the dynamics) are given.*

Under the above assumptions, the linear objective in (3) is by definition a linear function of the polynomial coefficients $c_{i,k}$. Furthermore, the infinite constraints in (3) can be equivalently written as a non-negativity constraint on the approximating polynomial $\hat{V}_k(x) = \sum_{i=1}^{N} c_{i,k} m_i(x)$. The resulting semi-infinite linear program is given by

$$\begin{aligned} \min_{\hat{V}_k(x)\in\mathbb{R}^n_r[x]} \quad & \int_{\mathbb{X}} \hat{V}_k(x)\varsigma(dx) \\ \text{subject to} \quad & \begin{cases} \hat{V}_k(x) - \mathbb{E}_w[\hat{V}^*_{k+1}(f(x,u,w))] \geq 0, \quad \forall\ (x,u) \in \bar{\mathbb{X}}\times\mathbb{U} \\ \hat{V}_k(x) - 1 \geq 0, \quad \forall\ x \in \mathbb{K} \end{cases} \end{aligned} \tag{4}$$

The objective function $\int_{\mathbb{X}} \hat{V}_k(x)\varsigma(dx)$ can be written as a scalar product between the coefficient vector $c_k = [c_{1,k}, \ldots, c_{N,k}] \in \mathbb{R}^N$ of $\hat{V}_k(x)$ and moments $\eta^{\varsigma}_{\alpha} := \int_{\mathbb{X}} x^{\alpha}\varsigma(dx)$. Since the dynamics are assumed to be polynomial, $\hat{V}^*_{k+1}(f(x,u,w))$ is a polynomial in x, u and w. Consequently, the expectation $\mathbb{E}_w[\hat{V}^*_{k+1}(f(x,u,w))]$ can be computed using moments of the noise distribution, which are available/computed up to the required order. Notice that until now we reduced the infinite dimensional LP to a yet intractable semi-infinite LP.

To ensure non-negativity of the approximating polynomial described by infinite constraints in (4) in a computationally tractable way, we restrict the decision space to sum-of-square (SOS) polynomials. Since checking for polynomial SOS is a semidefinite programming problem, the semi-infinite LP can be formulated as a SDP. To further improve computational tractability of the resulting SDP, recent research has been devoted to specific subsets of SOS polynomials, namely diagonally dominant (DSOS) and scaled diagonally dominant (SDSOS) polynomials [16, 17]. DSOS and SDSOS result in converting the SDP-based relaxations with attractive linear program and second order cone programs (SOCP) with lower complexity than the SDP.

The optimization program (4) is initialized at $k = T$, by approximating $\mathbb{1}_{\mathbb{K}}(x)$ as tightly as possible with a polynomial $\hat{V}^*_T(x)$. Problem (4) is solved $T-1$ times, resulting in polynomial upper bounds on the value functions at each time step. Algorithm 1 describes the procedure for approximating the maximum reach-avoid probability with polynomial basis functions.

Algorithm 1 Approximating the Value Function (offline computation)

1: **Input Data:**
- Semi-algebraic set description for $\mathbb{X}, \mathbb{K}, \mathbb{K}', \mathbb{U}$.
- Polynomial dynamics $f(x,u,w)$.
- Moments $\mathbb{E}[w^{\alpha}] = \int_{\mathbb{R}^p} w^{\alpha}\mathcal{Q}(w)dw$.

2: **Design Parameters:**
- Degree of polynomials $\hat{V}_k(x)$.
- Polynomial family: SOS/DSOS/SDSOS.
- Polynomial $\hat{p}_{\mathbb{K}}(x) \geq \mathbb{1}_{\mathbb{K}}(x), \quad \forall\, x \in \mathbb{X}$.
- State-relevance measure ς.

3: Initialize $\hat{V}^*_T(x) \leftarrow \hat{p}_{\mathbb{K}}(x)$
4: **for** $k = T-1 : 0$ **do**
5: Compute the cost function $\int_{\mathbb{X}} \hat{V}(x)\varsigma(dx)$ as a linear function of the polynomial coefficients using moments of ς.
6: Compute $\mathbb{E}_w[\hat{V}^*_{k+1}(f(x,u,w))]$ as a polynomial in (x,u) by substituting monomials w^{α} with corresponding moments $\mathbb{E}[w^{\alpha}]$.
7: Solve SDP/LP/SOCP (based on choice of polynomial family) to construct $\hat{V}^*_k$.
8: **end for**

Given the sequence of approximated value functions $\hat{V}^*_1(x), \ldots, \hat{V}^*_T(x)$, define the polynomial $\xi_k(x,u) = \mathbb{E}_w[\hat{V}^*_{k+1}(f(x,u,w))]$. The approximated optimal reach-avoid policy is computed as:

$$\hat{\psi}^*_k(x_k) = \arg\max_{u \in \mathbb{U}} \xi_k(x_k, u). \tag{5}$$

Given $x \in \mathbb{X}$, above is a polynomial maximization problem. A solution to (5) can be found by solving a hierarchy of convex linear matrix inequality (LMI) relaxations (Lasserre Hierarchy) [13]. Algorithm 2 describes the procedure for computing the policy online.

Algorithm 2 Approximate Policy (online computation)

for $k = 0 : T-1$ **do**
  Measure the current state of the system x_k
  if $x_k \notin \bar{\mathbb{X}}$ **then**
    return
  else
    Compute $\xi_k(x,u) = \mathbb{E}_w[\hat{V}^*_{k+1}(f(x,u,w))]$
    Set $x \leftarrow x_k$ and solve $u^* = \arg\max_{u\in\mathbb{U}}\{\xi_k(x_k,u)\}$ as Laserre hierarchy of order ϱ
    if Laserre hierarchy is infeasible **then**
      Increase order of Lasserre hierarchy $\varrho \leftarrow \varrho + 1$
    else
      $\hat{\psi}^*_k(x_k) = u^*$
    end if
  end if
end for

4 Numerical Analysis

In this section we evaluate the introduced approximation method through numerical examples. For each example, we follow a two step procedure: we first construct a sequence of polynomial functions that upper bound the value function of the reach-avoid problem using Algorithm 1 and then use Monte Carlo simulations to evaluate the reach-avoid probability of the control policies constructed using Algorithm 2. Upper bounds are constructed in MATLAB using the SPOT toolbox [16] that supports polynomial optimization and offers the functionality required to reformulate the relaxed semi-infinite LP in Problem (4) as an LP, SOCP or SDP, according to whether the decision space has been restricted to DSOS, SDSOS or SOS polynomials, respectively. The approximate control policies were computed using the Gloptipoly toolbox [15]. For the cases where the polynomial $\xi_k(x_k,u)$ or the control constraint set $\mathbb{U}$ had sparse structure (several monomials with zero coefficients), the computational time of solving (5) was reduced using the toolbox SparsePOP [18]. All the resulting convex programs were solved using MOSEK on an Intel i7-2600K CPU clocked @ 3.5 GHz with 16 GB of RAM memory.

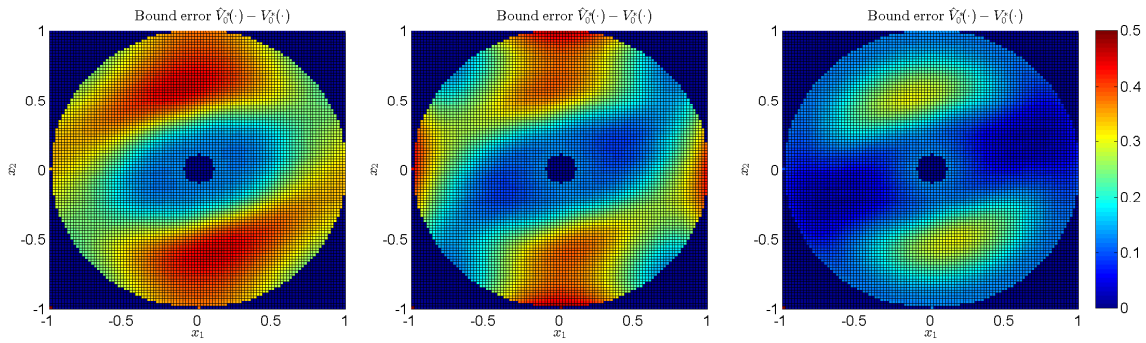
4.1 Benchmark Problem for Performance Bound

Consider the following 2D discretized system ($n = 2$) with cubic dynamics:

$$x(k+1) = \begin{bmatrix} x_1(k) + \tau \cdot [x_1(k) \cdot (\vartheta - x_1(k)^2 - x_2(k)^2) - x_2(k)] + u_1(k) + w_1(k) \\ x_2(k) + \tau \cdot [x_2(k) \cdot (\vartheta - x_1(k)^2 - x_2(k)^2) + x_2(k)] + u_2(k) + w_2(k) \end{bmatrix}. \quad (6)$$

The sampling rate is chosen to be $\tau = 0.1$ and the noise is assumed to be Gaussian $w_k \sim \mathcal{N}(\mathbf{0}_{n\times 1}, 0.001\cdot\mathbb{I}_{n\times n})$. Depending on the sign of parameter ϑ, we can identify a Hopf bifurcation. With $\vartheta = 0.5$, the noise-free system has three equilibria, two of which are stable and one which is unstable. We choose the safe set as $\mathbb{K}' = \{x \in \mathbb{R}^n : \|x\|_2 \leq 1\}$, so that the stable equilibria are not in $\mathbb{K}'$, and the target set as $\mathbb{K} = \{x \in \mathbb{R}^n : \|x\|_2 \leq 0.1\}$ in order to contain the unstable equilibrium. The input space is set to $\mathbb{U} = \{u \in \mathbb{R}^m : \|u\|_2 \leq 0.1\}$, $m = n$ and the reach-avoid specification is defined over a time horizon of $T = 7$ discrete time steps.

We chose quartic polynomials and a uniform state relevance measure $\varsigma(x)$ supported on $\mathbb{X}$ in Algorithm 1. In order to evaluate the quality of the polynomial approximation we computed



(a) DSOS (b) SDSOS (c) SOS

Figure 1: Error between grid-based and polynomial-based value functions, $\hat{V}^*_{0,\circledast}(\cdot) - V^*_0(\cdot)$.

the solution to the DP in (1) on uniformly discretized grids with 101 $\times$ 101 grid points for the state space $\mathbb{X} = \{x \in \mathbb{R}^2 : \|x\|_\infty \leq 1\}$, and 317 grid points for $\mathbb{U} = \{u \in \mathbb{R}^2 : \|u\|_2 \leq 0.1\}$, respectively. Computing the dynamic programming based (DP) value functions on the grid from time $k = T$ to time $k = 0$, denoted by V^*_k, required 638 seconds. In contrast, the polynomial-based approximations $\hat{V}^*_k$, were computed in less than 11 seconds. The difference $\hat{V}^*_0(\cdot) - V^*_0(\cdot)$ shown in Figure 1 corresponds to the pointwise difference (computed on grid elements of the set $\bar{\mathbb{X}}$) between the grid based and polynomial based value functions. The mean error $\bar{\epsilon}$ reported in Table 1, is computed over system states with reach-avoid probability higher than 0.1 in order to avoid bias from areas where both methods predict near-zero success probability.

Using the grid-based and DSOS/SDSOS/SOS approximate value functions, we also computed the associated control policies using Algorithm 2. We considered the grid-based policy as reference and compared it with the policies derived from the other three methods (SOS, DSOS, SDSOS polynomials) using Monte Carlo simulations to generate sample system trajectories. We denote by ψ^* the policy obtained from the grid-based approach and by $\hat{\psi}_\circledast$ the policy obtained by using one of the methods DSOS/SDSOS/SOS, labeled with the index $\circledast$. To estimate the performance of the different control policies, we sampled 100 initial states x_0, uniformly from $\bar{\mathbb{X}}$ and rejected states with low reach-avoid probability predicted by both methods. For the remaining ones, we generated 100 different disturbance trajectory realizations applying the corresponding control policies. The difference in performance $V(\psi^*) - \hat{V}(\hat{\psi}_\circledast)$ is reported in Table 1. One can observe that the approximate control policy based on the polynomials yields satisfactory performance (see Table 1) even though the average error between the value function of the gridding and that of each of the polynomial based approaches is not close to zero. This observation motivates using polynomial approximations to address control synthesis for reach-avoid problems of higher dimensions where gridding the continuous spaces is intractable.

Method $\circledast$	average error $\bar{\varepsilon}$	$V(\psi^*) - \hat{V}(\hat{\psi}_\circledast)$
DSOS	0.363	0.0134
SDSOS	0.295	0.0241
SOS	0.186	-0.0090

Table 1: Average error in value functions and performance of the resulting policies.

4.2 Computational Analysis

We focus here on the computational efforts needed to approximately solve a reach-avoid problem using quartic DSOS/SDSOS/SOS polynomials. We introduce the identity system

$$x_{k+1} = \mathbb{I}_{n\times n} \cdot x_k + \mathbb{I}_{n\times m} \cdot u_k + w_k, \tag{7}$$

with Gaussian noise $w_k \sim \mathcal{N}(\mathbf{0}_{n\times 1}, 0.001 \cdot \mathbb{I}_{n\times n})$ in order to easily change the dimensions of the problem (variables n, m) and keep the system behavior throughout the problems similar. We approximate the reach-avoid problem for a finite time horizon of $T = 5$ with the safe, target and input sets as introduced in Section 4.1, with appropriate n, m dimensions. Table 2 depicts the CPU times of different computational phases when Algorithm 1 and Algorithm 2 are executed. The column "Setup" refers to all the operations required before solving the SDP in (4). "Optimization" refers to the time required to construct the approximating value functions and "Controller Synthesis" corresponds to the time required to setup the polynomial (5) needed to extract control actions from the value function approximations, averaged between DSOS/SDSOS/SOS methods. Online policy extraction CPU times are reported in Table 3. We have compared the grid-based control policy, the policy based on a standard Lasserre SDP relaxation [13] solved in Gloptipoly and a policy constructed using the sparse Lasserre SDP relaxation solved in SparsePOP [18]. The reported times correspond to the times averaged over the horizon $\mathbb{T}$ to maximize (5) given a fixed initial condition and a sequence of disturbances.

$n = m$	Setup	Optimization			Controller Synthesis
-	-	SOS	DSOS	SDSOS	-
2	0.22 s	1.75 s	0.17 s	0.23 s	0.17 s
4	1.89 s	0.44 s	0.42 s	0.43 s	0.54 s
6	2.55 s	1.21 s	1.01 s	2.59 s	1.82 s
8	6.54 s	22.70 s	3.87 s	4.42 s	5.44 s
10	23.69 s	153.17 s	8.28 s	11.08 s	14.35 s

Table 2: Computational analysis for approximating value function through DSOS/SDSOS/SOS.

m	# of grid elements of $\mathbb{U}$	Gridding $\mathbb{U}$	Dense SDP	Sparse SDP
2	317	0.003 s	0.094 s	0.365 s
4	49'689	0.18 s	0.56 s	0.59 s
6	5'257'045	145.48 s	3.28 s	0.87 s
8	`out of memory`	`N/A` (∞)	16.30 s	1.90 s
10	`out of memory`	`N/A` (∞)	66.30 s	6.74 s

Table 3: Computational analysis of approximating policy comparing control space gridding, dense polynomial maximization and sparse polynomial maximization, both relaxed as SDP.

We have reported different CPU times required for each task of the process in Tables 2 and 3 as function of the dimension. We observe that DSOS/SDSOS polynomials confirm to be more scalable than SOS as Table 2 shows. Sparsity exploitation provides a more scalable policy extraction in higher dimensions as confirmed in Table 3.

4.3 Performance Analysis on Large Scale Problems

Here, we focus on evaluating the performance of approximate control policies in reach-avoid problems of high dimensions. Due to *curse of dimensionality* the gridding based approach for computing the optimal value function and policy cannot serve as a benchmark. Consequently, as a basis of comparison we use the approach in [11] where radial basis functions (RBFs), instead of polynomials, are used to construct a semi-infinite LP (4) and to reformulate the infinite constraint in (4) using a conservative $\mathcal{S}$-procedure. We compare the performance of the RBF approach to the tool presented in this paper. In all simulations we consider quartic polynomials and average the performance of approximate control policies over 50 discrete-time controllable linear time invariant systems of dimension $\dim(\mathbb{X} \times \mathbb{U}) = 8$. System matrices are generated randomly with eigenvalues restricted to $|\lambda_i| \in [0.75, 1],\ \forall\ i = 1, \ldots, n$ in order to avoid stable systems that converge quickly to the origin without any control authority. The target, avoid and input sets are the same as those described in Section 4.1.

For each system we computed the average difference in performance between the two approximate control design approaches. The first controller is based on polynomial basis and is the result of executing Algorithms 1 and 2 for the DSOS/SDSOS/SOS methods while the second controller is based on RBFs and the method outlined in [11]. In Table 4 we denote by $\hat{V}(\hat{\psi}^{\circledast}) - \hat{V}(\hat{\psi}_{\mathrm{RBF}})$ the mean difference of the closed loop value functions of each method. We denote by $\hat{\psi}^{\circledast}$ the policy obtained by using one of the methods DSOS/SDSOS/SOS, labeled with the index $\circledast$ and by $\hat{\psi}_{\mathrm{RBF}}$ the policy obtained from the RBF based approach. The computation is done similar to the problem discussed in Section 4.1. In particular, we sampled 100 initial states x_0 uniformly from $\bar{\mathbb{X}}$. Then, we generated 100 different disturbance trajectory realizations applying the different control policies to the resulting states over the time horizon. Based on these experiments, the RBF approach yields tighter reach-avoid value function bounds. However, a limitation with this approach as discussed in [11] is that it can handle problems of dimension ≤ 8 and only linear systems with additive Gaussian mixture noise.

Method	$\hat{V}(\hat{\psi}^{\circledast}) - \hat{V}(\hat{\psi}_{\mathrm{RBF}})$	σ_{RBF}
DSOS	$-$ 0.123	$\pm$ 0.090
SDSOS	$-$ 0.108	$\pm$ 0.074
SOS	$-$ 0.097	$\pm$ 0.067

Table 4: Average performance difference and standard deviation between polynomials and RBF approach for $\dim(\mathbb{X} \times \mathbb{U}) = 8$.

5 Conclusion

We proposed a novel approximation scheme to construct polynomial upper-bounds to reach-avoid value functions. Using these approximate value functions we synthesized approximate controllers for problems of higher dimension than what has been reported in the literature. Future work will address developing a complete toolbox for handling reach-avoid control problems for Markov Decision Process and discrete time stochastic hybrid systems by integrating features from the various tools available in the polynomial optimization literature. Moreover, we are planning to extend the work to include a basis selection step that exploits symmetry and reduces the number of monomials used. The automated use of such insights will result in sparsity and consequently reduced total computational time.

References

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

[4] M. Prandini, J. Hu. *Stochastic reachability: Theory and numerical approximation.* Stochastic hybrid system, Automation and Control Engineering Series, 24, pp. 107-138, 2006.

[5] M. Kamgarpour, S.Summers, J. Lygeros. *Control design for specifications on stochastic hybrid systems.* Hybrid Systems Computation and Control, 2013.

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

[7] Y. Wang, B. O'Donoghue, S. Boyd. *Approximate Dynamic Programming via Iterated Bellman Inequalities.* International Journal of Robust and Nonlinear Control.

[8] V. Desai, C. Moallemi, V. Farias. *Approximate Dynamic programming via Smoothed Linear Program.* ArXiv 2009.

[9] N. Kariotoglou, S. Summers, T. Summers. M. Kamgarpour, J. Lygeros. *Approximate dynamic programming for stochastic reachability.* In Control Conference (ECC), 2013 European, IEEE, pp. 584-589.

[10] N. Kariotoglou, S. Summers, T. Summers. M. Kamgarpour, J. Lygeros. *A Numerical Approach to Stochastic Reach-Avoid Problems for Markov Decision Processes.* arXiv, 2014. http://arxiv.org/abs/1411.5925.pdf

[11] N. Kariotoglou, M. Kamgarpour, T. H. Summers, J. Lygeros. *Upper Bounds for the Reach-Avoid Probability via Robust Optimization.* ETH Zürich, 2015. http://arxiv.org/abs/1506.03371.pdf

[12] T. Summers, K. Kunz, N. Kariotoglou, M. Kamgarpour, S. Summers, J. Lygeros. *Approximate Dynamic Programming via Sum of Squares Programming.* arXiv 2012.

[13] J. B. Lasserre. *Global Optimization with Polynomials and the Problem of Moments.* 2001

[14] A. Papachristodoulou, J. Anderson, G. Valmorbida, S. Prajna, P. Seiler, P. A. Parillo. *SOSTOOL: Sum of Squares Optimization Toolbox for MATLAB User's Guide.* 2013.

[15] D. Henrion, J-B. Lasserre, J. Löfberg. *GloptiPoly 3: Moments, Optimization and Semidefinite Programming.* 2008.

[16] A. Majumdar, A. A. Ahmadi, R. Tedrake. *Control and Verification of High-Dimensional Systems with DSOS and SDSOS Programming.* 2014.

[17] A. Majumdar, A. A. Ahmadi, R. Tedrake. *Some Applications of Polynomial Optimization in Operations Research and Real-Time Decision Making.* 2015.

[18] H. Waki, S. Kim, M. Kojima, M. Muramatsu, H. Sugimoto. *SparsePOP: a Sparse Semidefinite Programming Relaxation of Polynomial Optimization Problems.* Tokyo Institute of Technology. 2007.

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