

Upper bounds for the reach-avoid probability via robust optimization

Nikolaos Kariotoglou^a, Maryam Kamgarpour^a, Tyler H. Summers^a, and John Lygeros^a

^a*Automatic Control Laboratory, Department of Information Technology and Electrical Engineering, ETH Zürich, Zürich 8092, Switzerland*

Abstract

We consider finite horizon reach-avoid problems for discrete time stochastic systems. Our goal is to construct upper bound functions for the reach-avoid probability by means of tractable convex optimization problems. We achieve this by restricting attention to the span of Gaussian radial basis functions and imposing structural assumptions on the transition kernel of the stochastic processes as well as the target and safe sets of the reach-avoid problem. In particular, we require the kernel to be written as a Gaussian mixture density with each mean of the distribution being affine in the current state and input and the target and safe sets to be written as intersections of quadratic inequalities. Taking advantage of these structural assumptions, we formulate a recursion of semidefinite programs where each step provides an upper bound to the value function of the reach-avoid problem. The upper bounds provide a performance metric to which any suboptimal control policy can be compared, and can themselves be used to construct suboptimal control policies. We illustrate via numerical examples that even if the resulting bounds are conservative, the associated control policies achieve higher reach-avoid probabilities than heuristic controllers for problems of large state-input space dimensions (more than 20). The results presented in this paper, far exceed the limits of current approximation methods for reach-avoid problems in the specific class of stochastic systems considered.

Key words: Reachability, approximate dynamic programming, value function bounds, Markov decision processes, stochastic control.

1 Introduction

We deal with Markov decision processes of specific structure and focus on the stochastic reach-avoid problem where one maximizes the probability of reaching a target set while staying in a safe set, under given control constraints. This type of optimal control problem can be solved by a dynamic programming recursion [1,2] but its practical applications are limited due to the infamous curse of dimensionality that affects state-of-the-art space-gridding techniques. Despite their limitations, space-gridding methods are theoretically attractive since they provide straightforward ways to estimate approximation errors with respect to the optimal solution function, under general Lipschitz continuity assumptions on the associated system dynamics [3,4,5]. In an attempt to extend the possible applications of the reach-avoid problem, we follow a less standard approach [6] than dynamic programming to characterize the optimal value function of the stochastic reach-avoid problem. We then construct an upper bound for the optimal value function and evaluate in practice the performance of the associated approximate control policy .

The main tools used in this work stem from convex optimization and methods used to reformulate robust inequalities [7,8,9] into semidefinite constraints. The authors in [7] used a similar framework and showed its success in terms of control performance in a different type of optimal control problem. The framework considered here is fundamentally

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

Email addresses: karioto@control.ee.ethz.ch (Nikolaos Kariotoglou), mkamgar@control.ee.ethz.ch (Maryam Kamgarpour), tsummers@control.ee.ethz.ch (Tyler H. Summers), lygeros@control.ee.ethz.ch (John Lygeros).

different due to the different type of basis approximation functions used and the way that reach-avoid problems handle state and control constraints naturally via the definition of their cost function. Another approach that addresses the approximation of reach-avoid problems based on the method suggested in [6] is presented in [10,11]. Compared to the results in [11], the technique developed here provides deterministic upper bounds to the reach-avoid value function (as opposed to probabilistic ones), scales to significantly higher dimensional problems, but requires stronger restrictions on the kernel of the stochastic processes.

Our results are based on restricting the decision space of an infinite dimensional linear program [6,11] to the space spanned by a finite collections of Gaussian radial basis functions (RBFs) and then searching for means, variances and weights of a sum of such functions to upper bound the reach-avoid probability. We show that for some process kernels, such as the kernel of a linear system with additive Gaussian mixture noise, this restriction allows us to analytically compute the expectation of the stochastic process over the basis functions and reduce the problem of upper bounding the optimal reach-avoid value function to determining the parameters of a Gaussian RBF sum that upper bounds another such sum. We derive sufficient conditions in terms of linear matrix inequalities (LMIs) to characterize dominance between Gaussian RBF sums and then use them to formulate a semidefinite program (SDP) with which one can construct an upper bound to the optimal value function at each step of the reach-avoid recursion. We show via numerical examples that approximate control policies defined using these bounds scale to problems of significantly higher dimensions than what is currently possible with state gridding techniques.

The rest of the paper is organized as follows: In Section 2 we recall the basic definition of the reach-avoid problem for Markov decision processes and formulate a robust optimization problem whose solution is an upper bound to the reach-avoid value function. We focus most of Section 3 to reformulate the constraints of the robust optimization problem into sufficient conditions that can be checked efficiently with existing convex optimization software. Using these conditions we provide an algorithm to construct upper bounds for the value function of every step of a reach-avoid recursion. We then use the constructed bounds to define approximate control policies that maximize the reach-avoid probability. In Section 4 we provide numerical examples to estimate the suboptimality gap between the constructed bounds and the optimal value function (computed via gridding) and illustrate that the approximate control policies can outperform heuristic controllers even when the bounds are not tight.

2 Stochastic Reach-Avoid Problem

2.1 Dynamic programming approach

Let $\mathcal{X} = \mathbb{R}^n$ denote a continuous state space and $\mathcal{U} \subset \mathbb{R}^m$ a continuous and compact control space. 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} being 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 Borel-measurable function on $\mathcal{X} \times \mathcal{U}$ for each $A \in \mathcal{B}(\mathcal{X})$ and $Q(\cdot|x, u)$ is a probability measure on \mathcal{X} for each (x, u) . For the rest of the paper all measurability conditions refer to Borel measurability. We consider a safe set $K' \in \mathcal{B}(\mathcal{X})$ and a target set $K \subseteq K'$. We define an admissible T -step control policy to be a sequence of measurable functions $\pi = \{\pi_0, \dots, \pi_{T-1}\}$ where $\pi_i : \mathcal{X} \rightarrow \mathcal{U}$ 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 π as: $r_{x_0}^\pi(K, K') = \mathbb{P}_{x_0}^\pi \{\exists j \in [0, T] : x_j \in K \wedge \forall i \in [0, j-1], x_i \in K' \setminus K\}$ and operate under the convention that $[0, -1] = \emptyset$, which implies that the requirement on i is automatically satisfied when $x_0 \in K$.

In [2], $r_{x_0}^\pi(K, K')$ is shown to be equivalent to the following sum multiplicative cost function:

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

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 [12] but for simplicity

we assume here that they are constant. We denote the difference between the safe and target sets by $\bar{\mathcal{X}} := K' \setminus K$ to simplify the presentation of our results.

The solution to the reach-avoid problem is given by a dynamic recursion [2]. Define $V_k^* : \mathcal{X} \rightarrow [0, 1]$ for $k = T-1, \dots, 0$ by :

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

The value of the above recursion at $k = 0$ and for any initial state x_0 is the supremum of (1) over all admissible policies, i.e. $V_0^*(x_0) = \sup_{\pi} r_{x_0}^{\pi}(K, K')$ [2]. In [11] the authors proved measurability of the value functions (and hence attainability of the supremum) in (2) under the assumption that Q is continuous with respect to $u \in \mathcal{U}$ for any given $x \in \mathcal{X}$. For the rest of this work we operate under the same kernel continuity assumption:

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

We define the optimal reach-avoid feedback policy at each state k by

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

The only established way to approximate the solution of (2) is by gridding $\mathcal{X} \times \mathcal{U}$ and computing it backwards, starting 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 the grid points of \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 [3,4]) 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 dimensions of the state and control spaces increase. In what follows we formulate an infinite dimensional linear program equivalent to the reach-avoid recursion and use tools from robust optimization to construct upper bound functions.

2.2 Robust optimization problem

Inspired by the linear programming approach to dynamic programming [6], the authors in [11] formulated very similar version to the following infinite dimensional linear program over the space of measurable functions $\mathcal{F} = \{f : \mathcal{X} \rightarrow \mathbb{R}, f \text{ measurable}\}$ to characterize the reach-avoid value function at time k :

$$\begin{aligned} \min_{V(\cdot) \in \mathcal{F}} \quad & \int_{\mathcal{X}} V(x) \nu(dx) \\ \text{subject to} \quad & V(x) \geq \mathbb{1}_K(x) + \mathbb{1}_{\bar{\mathcal{X}}}(x) \int_{\mathcal{X}} V_{k+1}^*(z) Q(dz|x, u), \quad \forall (x, u) \in \bar{\mathcal{X}} \times \mathcal{U} \end{aligned} \tag{4}$$

where we use \min instead of \inf since, as shown in [11], Assumption 1 ensures attainability of the optimizer. The constraints in (4) are equivalent to the pair of constraints

$$\begin{aligned} V(x) &\geq 1, \quad \forall x \in K \\ V(x) &\geq \int_{\mathcal{X}} V_{k+1}^*(z) Q(dz|x, u), \quad \forall (x, u) \in \bar{\mathcal{X}} \times \mathcal{U}. \end{aligned}$$

The measure ν is interpreted as a state relevance measure [6] that can be chosen to bias the value of the cost function in different places of the state space. In order to search for a feasible point in (4), we restrict the infinite dimensional function space \mathcal{F} to a finite dimensional subspace $\tilde{\mathcal{F}}$. Assume that V_{k+1}^* is known and consider the

following semi-infinite linear program defined over functions $V \in \tilde{\mathcal{F}} \subset \mathcal{F}$

$$\begin{aligned} \min_{V \in \tilde{\mathcal{F}}} \quad & \int_{\mathcal{X}} V(x) \nu(\mathrm{d}x) \\ \text{subject to} \quad & V(x) \geq \int_{\mathcal{X}} V_{k+1}^*(z) Q(\mathrm{d}z | x, u), \quad \forall (x, u) \in \bar{\mathcal{X}} \times \mathcal{U} \\ & V(x) \geq 1, \quad \forall x \in K. \end{aligned} \tag{5}$$

To guarantee that the minimum is attained in (5), additional assumptions may be required on $\tilde{\mathcal{F}}$. For instance, $\tilde{\mathcal{F}}$ would have to be bounded to exclude situations where the optimization program has an unbounded cost function, e.g. $-\infty$ on the complement of $\bar{\mathcal{X}}$ which is unconstrained. An alternative approach is to explicitly add constraints on the value of V on the complement of $\bar{\mathcal{X}}$, such as $V(x) \geq 0, \forall x \in \mathcal{X} \setminus \bar{\mathcal{X}}$ since V represents a reach-avoid probability. The rest of the section is devoted to reformulating the constraints of (5) into conditions that can be checked efficiently using convex optimization software.

3 Robust constraint reformulation

Checking the constraints in (5) numerically is, in general, intractable due to the infinite cardinality of the constraint set. Depending on the structure of $K, \bar{\mathcal{X}} \times \mathcal{U}$, the function $\int_{\mathcal{X}} V_{k+1}^*(z) Q(\mathrm{d}z | x, u)$ and the representation of V in $\tilde{\mathcal{F}}$, it may be possible to reformulate the constraints. In this section we impose specific structure to these elements and derive sufficient conditions to construct feasible solutions for (5).

3.1 Gaussian radial basis functions

Let the restricted decision space be $\tilde{\mathcal{F}} = \mathcal{F}^M$ where \mathcal{F}^M denotes the span of a set $\{\phi(x, \mu_i, \Sigma_i)\}_{i=1}^M$ of $M \in \mathbb{N}_+$ Gaussian radial basis functions (RBFs) each one defined as,

$$\phi(x, \mu_i, \Sigma_i) := \frac{1}{\sqrt{(2\pi)^n |\Sigma_i|}} \exp \left[-\frac{1}{2} (x - \mu_i)^\top \Sigma_i^{-1} (x - \mu_i) \right] \tag{6}$$

where the operator $|\cdot|$ denotes the determinant of the argument matrix. Note that each function $\phi(x, \mu_i, \Sigma_i)$ corresponds to the density function of a multivariate Gaussian random variable $\mathcal{N}(\mu_i, \Sigma_i)$ with $x, \mu_i \in \mathbb{R}^n, \Sigma_i \in \mathbf{S}_+^n$ where \mathbf{S}_+^n denotes the space of (symmetric) positive definite matrices. We use the standard notation $X \succcurlyeq 0$ to denote that a matrix $X \in \mathbb{R}^{n \times n}$ is in \mathbf{S}_+^n and hence we have that $\Sigma_i \succcurlyeq 0$. Every element of $\psi \in \mathcal{F}^M$ can be written as a Gaussian RBF sum defined as:

Definition 1 (Gaussian RBF sum) *A Gaussian RBF sum $\psi : \mathcal{X} \rightarrow \mathfrak{R}$ is a linear combination of functions in the form of (6), i.e. $\psi(x) = \sum_{i=1}^M w_i \phi(x, \mu_i, \Sigma_i)$ for some $M \in \mathbb{N}$ and $w \in \mathbb{R}^M$ with $\Sigma_i \succcurlyeq 0$ and $\mu_i \in \mathfrak{R}^n$ for $i = 1, \dots, M$ where n denotes the dimension of \mathcal{X} .*

In what follows, we derive conditions by which one can choose $\{w_i, \mu_i, \Sigma_i\}_{i=1}^M$ to upper bound the reach-avoid value function. We restrict ourselves to quadratic sets, defined as:

Definition 2 (Quadratic set in \mathbb{R}^n) *A quadratic set is any set $A \subset \mathbb{R}^n$ that can be written as*

$$A = \left\{ x \in \mathbb{R}^n \mid \begin{bmatrix} x \\ 1 \end{bmatrix}^\top A_j \begin{bmatrix} x \\ 1 \end{bmatrix} \geq 0, \quad \forall j \in \{1, \dots, N\} \right\}$$

for some $N \in \mathbb{N}_+$ and symmetric matrices $A_j \in \mathbb{R}^{n+1}$.

Sets satisfying the condition in Definition 2 include intersections of ellipsoids and half spaces that can be used to approximate convex sets arbitrarily closely. However, quadratic sets are not a subset of convex sets since one can, for example, express the difference between ellipsoids and half spaces as a quadratic set which implies that a wide

range of non-convex sets satisfy Definition 2. Given two function elements in \mathcal{F}^M , the following lemma establishes sufficient conditions on their weights, means and variances such that one uniformly dominates the other on a given quadratic set:

Lemma 3 (RBF dominance) *Let $\{\hat{w}_i, \hat{\mu}_i, \hat{\Sigma}_i\}_{i=1}^M$ and $\{w_i, \mu_i, \Sigma_i\}_{i=1}^M$ denote the weights, means and variances of two Gaussian RBF sum functions $\hat{\psi}, \psi \in \mathcal{F}^M$ and assume the weights $\{w_i\}_{i=1}^M$ are non-negative. Consider a quadratic set $A \subset \mathbb{R}^n$ described by N inequalities. We have that $\hat{\psi}(x) \geq \psi(x), \forall x \in A$ if for all $i = 1, \dots, M$, $\exists \tau_i \geq_{\mathbb{R}^N} \mathbf{0}$, such that $X_i \succcurlyeq \mathbf{0}$*

$$X_i = \begin{bmatrix} Q_{\hat{\Sigma}_i}^{-1} & Q_{\hat{\mu}_i} \\ Q_{\hat{\mu}_i}^\top & Q_i + Q_{\mu_i, \Sigma_i} - \sum_{j=1}^N \tau_{i,j} A_j \end{bmatrix}$$

where

$$Q_i = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 2 \log \left(\frac{\hat{w}_i \sqrt{|\Sigma_i|}}{w_i \sqrt{|\hat{\Sigma}_i|}} \right) \end{bmatrix}, \quad Q_{\hat{\mu}_i} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I} & -\hat{\mu}_i \end{bmatrix}, \quad Q_{\hat{\Sigma}_i} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \hat{\Sigma}_i^{-1} \end{bmatrix}, \quad Q_{\mu_i, \Sigma_i} = \begin{bmatrix} \Sigma_i^{-1} & -\Sigma_i^{-1} \mu_i \\ -\mu_i^\top \Sigma_i^{-1} & \mu_i^\top \Sigma_i^{-1} \mu_i \end{bmatrix}$$

and $\mathbf{0}, \mathbf{I}$ denote zero and identity matrices of appropriate dimensions. The terms $\tau_i \geq_{\mathbb{R}^N} \mathbf{0}$ denote element-wise non-negativity where each τ_i is an element of space \mathbb{R}^N .

PROOF. Notice that a sufficient condition to ensure that $\hat{\psi}(x) \geq \psi(x), \forall x \in A$ is that $\hat{w}_i \phi(x, \hat{\mu}_i, \hat{\Sigma}_i) \geq w_i \phi(x, \mu_i, \Sigma_i), \forall x \in A, i = 1, \dots, M$. Consider $x \in \mathbb{R}^n$ and let $z = \begin{bmatrix} x \\ 1 \end{bmatrix} \in \mathbb{R}^{n+1}$. Since all w_i are assumed non-negative, the weights \hat{w}_i also have to be non-negative and we can take the natural logarithm on the above relation to get $\log(\hat{w}_i \phi(x, \hat{\mu}_i, \hat{\Sigma}_i)) - \log(w_i \phi(x, \mu_i, \Sigma_i)) \geq 0, \forall x \in A$ if and only if

$$z^\top A_j z \geq 0, \quad j = 1, \dots, N \implies z^\top (Q_{\mu_i, \Sigma_i} - Q_{\hat{\mu}_i, \hat{\Sigma}_i} + Q_i) z \geq 0 \quad (7)$$

where $Q_{\mu_i, \Sigma_i}, Q_{\hat{\mu}_i, \hat{\Sigma}_i}$ and Q_i are defined as

$$Q_{\mu_i, \Sigma_i} = \begin{bmatrix} \Sigma_i^{-1} & -\Sigma_i^{-1} \mu_i \\ -\mu_i^\top \Sigma_i^{-1} & \mu_i^\top \Sigma_i^{-1} \mu_i \end{bmatrix}, \quad Q_{\hat{\mu}_i, \hat{\Sigma}_i} = \begin{bmatrix} \hat{\Sigma}_i^{-1} & -\hat{\Sigma}_i^{-1} \hat{\mu}_i \\ -\hat{\mu}_i^\top \hat{\Sigma}_i^{-1} & \hat{\mu}_i^\top \hat{\Sigma}_i^{-1} \hat{\mu}_i \end{bmatrix}, \quad Q_i = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 2 \log \left(\frac{\hat{w}_i \sqrt{|\Sigma_i|}}{w_i \sqrt{|\hat{\Sigma}_i|}} \right) \end{bmatrix}$$

Applying the S-procedure to the condition in (7), we have that if

$$\exists \tau_{i,j} \geq 0 \text{ such that } Q_{\mu_i, \Sigma_i} - Q_{\hat{\mu}_i, \hat{\Sigma}_i} + Q_i - \sum_{j=1}^N \tau_{i,j} A_j \succcurlyeq \mathbf{0}$$

then

$$z^\top A_j z \geq 0, \quad j = 1, \dots, N \implies z^\top (Q_{\mu_i, \Sigma_i} - Q_{\hat{\mu}_i, \hat{\Sigma}_i} + Q_i) z \geq 0.$$

Consider now the transformation $Q_{\hat{\mu}_i, \hat{\Sigma}_i} = Q_{\hat{\mu}_i}^\top Q_{\hat{\Sigma}_i} Q_{\hat{\mu}_i}$ with $Q_{\hat{\mu}_i} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I} & -\hat{\mu}_i \end{bmatrix}$, $Q_{\hat{\Sigma}_i} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \hat{\Sigma}_i^{-1} \end{bmatrix}$. Notice that $Q_{\mu_i, \Sigma_i} - Q_{\hat{\mu}_i}^\top Q_{\hat{\Sigma}_i} Q_{\hat{\mu}_i} + Q_i - \sum_{j=1}^N \tau_{i,j} A_j$ is the Schur complement of the matrix

$$X_i = \begin{bmatrix} Q_{\hat{\Sigma}_i}^{-1} & Q_{\hat{\mu}_i} \\ Q_{\hat{\mu}_i}^\top & Q_{\mu_i, \Sigma_i} + Q_i - \sum_{j=1}^N \tau_{i,j} A_j \end{bmatrix}$$

and since $Q_{\hat{\Sigma}_i}^{-1}$ is positive definite by definition ($\hat{\Sigma}_i$ is the covariance matrix of a Gaussian RBF), the condition $Q_{\mu_i, \Sigma_i} - Q_{\hat{\mu}_i}^\top Q_{\hat{\Sigma}_i} Q_{\hat{\mu}_i} + Q_i - \sum_{j=1}^N \tau_{i,j} A_j \succcurlyeq 0$ is equivalent to $X_i \succcurlyeq 0$.

Notice that there may be cases where multipliers τ_i do not exist unless additional boundedness assumptions are imposed on the set A . In what follows we restrict the structure of the reach-avoid problem and illustrate how one can express the right-hand-side of the constraints in (5) as a sum of Gaussian RBFs. We then use the result of Lemma 3 to construct upper bound approximations for the reach-avoid value functions.

3.2 Upper bound for the reach-avoid value functions

We impose the following assumptions in order to express the optimization problem in (5) into the framework of Gaussian RBFs. The process involves two steps: we first derive a general result concerning the expectation of a Gaussian RBF sum applied to a random variable whose density function is a Gaussian RBF sum and then apply it to the constraints in (5). We make use of the following:

Lemma 4 *Given a bounded set $A \subset \mathbb{R}^n$, there always exists a Gaussian RBF sum \hat{p}_A such that $\hat{p}_A(x) \geq \mathbb{1}_A(x)$ for all $x \in \mathbb{R}^n$.*

The proof of Lemma 4 is omitted since it is trivial to show that even a single Gaussian RBF in the form of (6) is enough to upper bound the indicator function of a bounded set by simply taking a large enough scaling weight.

Assumption 2 *The stochastic kernel $Q(\cdot|x, u)$ is defined through a Gaussian RBF sum density function of the form $\sum_{j=1}^J w_j \phi(y, \mu_j(x, u), \Sigma_j)$ for $J \in \mathbb{N}_+$ and functions μ_j affine in (x, u) .*

The condition imposed on the transition kernels by Assumption 2 holds for a wide range of stochastic dynamical systems. For example, linear systems affected by general additive Gaussian mixture noise satisfy Assumption 2 and can be used as a basis to approximate the kernel of more general non-linear systems.

Lemma 5 (RBF expectation) *Let $y \in \mathbb{R}^n$ be a random variable distributed according to a Gaussian RBF sum density function of the form $\sum_{j=1}^J \bar{w}_j \phi(x, \bar{\mu}_j, \bar{\Sigma}_j)$ for some $J \in \mathbb{N}_+$ and $x \in \mathbb{R}^n$. Consider a Gaussian RBF sum function $g \in \mathcal{F}^M$ defined by $g(x) = \sum_{i=1}^M w_i \phi(x, \mu_i, \Sigma_i)$ for any $x \in \mathbb{R}^n$. The expected value of the random variable $g(y)$ is a Gaussian RBF sum given by $\sum_{i=1}^M \sum_{j=1}^J w_i \bar{w}_j \phi(\bar{\mu}_j, \mu_i, \Sigma_i + \bar{\Sigma}_j)$*

PROOF. Consider the function $g(x) = \sum_{i=1}^M w_i \phi(x, \mu_i, \Sigma_i)$ and a random variable $y \in \mathbb{R}^n$ with density function given by $\sum_{j=1}^J \bar{w}_j \phi(x, \bar{\mu}_j, \bar{\Sigma}_j)$, $x \in \mathbb{R}^n$. We have that

$$\begin{aligned} \mathbb{E}[g(y)] &= \int_{\mathbb{R}^n} \sum_{i=1}^M w_i \phi(x, \mu_i, \Sigma_i) \sum_{j=1}^J \bar{w}_j \phi(x, \bar{\mu}_j, \bar{\Sigma}_j) dx \\ &= \int_{\mathbb{R}^n} \sum_{i=1}^M \sum_{j=1}^J w_i \bar{w}_j \phi(x, \mu_i, \Sigma_i) \phi(x, \bar{\mu}_j, \bar{\Sigma}_j) dx \\ &= \sum_{i=1}^M \sum_{j=1}^J w_i \bar{w}_j \int_{\mathbb{R}^n} \phi(x, \mu_i, \Sigma_i) \phi(x, \bar{\mu}_j, \bar{\Sigma}_j) dx \\ &= \sum_{i=1}^M \sum_{j=1}^J w_i \bar{w}_j \phi(\bar{\mu}_j, \mu_i, \Sigma_i + \bar{\Sigma}_j) \end{aligned}$$

The last equality follows from the fact that the product of two functions in the form of (6) is proportional to another function of the same form [13, Section 2]. With

$$\phi(x, \mu_i, \Sigma_i) \phi(x, \bar{\mu}_j, \bar{\Sigma}_j) = \phi(\mu_i, \bar{\mu}_j, \Sigma_i + \bar{\Sigma}_j) \phi(x, \bar{\mu}, \bar{\Sigma}),$$

$\bar{\mu} = (\Sigma_i^{-1} + \bar{\Sigma}_j^{-1})^{-1} (\Sigma_i^{-1} \mu_i + \bar{\Sigma}_j^{-1} \bar{\mu}_j)$, and $\bar{\Sigma} = (\Sigma_i^{-1} + \bar{\Sigma}_j^{-1})^{-1}$. The result follows as the proportionality constant $\phi(\mu_i, \bar{\mu}_j, \Sigma_i + \bar{\Sigma}_j)$ is independent of x and the function $\phi(x, \bar{\mu}, \bar{\Sigma})$ integrates to 1 over \mathbb{R}^n .

Proposition 6 (Upper bound) *Consider Lemma 4 and Assumption 2. Let $\hat{V}_T^*(x) = \hat{p}_K(x)$ where \hat{p}_K upper bounds $\mathbb{1}_K$. The following recursion of optimization problems provides an upper bound on the reach-avoid value function for each $k = T - 1, \dots, 0$, for any choice of non-negative measure ν .*

$$\begin{aligned} \hat{V}_k^* &= \arg \min_{\hat{V}_k \in \bar{\mathcal{F}}} \int_{\mathcal{X}} \hat{V}_k(x) \nu(dx) \\ \text{subject to } \hat{V}_k(x) &\geq \int_{\mathcal{X}} \hat{V}_{k+1}(z) Q(dz | x, u), \quad \forall (x, u) \in \bar{\mathcal{X}} \times \mathcal{U} \\ \hat{V}_k(x) &\geq 1 \quad \forall x \in K. \end{aligned} \tag{8}$$

PROOF. We use induction. The reach-avoid value function at time $T - 1$ is defined by

$$V_{T-1}^*(x) = \max_{u \in \mathcal{U}} \left\{ \mathbb{1}_K(x) + \mathbb{1}_{\bar{\mathcal{X}}}(x) \int_{\mathcal{X}} V_T^*(z) Q(dz | x, u) \right\}, \forall x \in \mathcal{X}.$$

By definition we have that $\hat{V}_T^*(x) = \hat{p}_K(x) \geq \mathbb{1}_K(x) =: V_T^*(x)$ for all $x \in \mathcal{X}$. We also have that for all $(x, u) \in \mathcal{X} \times \mathcal{U}$, $\int_{\mathcal{X}} \hat{V}_T^*(z) Q(dz | x, u) \geq \int_{\mathcal{X}} V_T^*(z) Q(dz | x, u)$. Hence, for any feasible point \hat{V}_{T-1} of (8), we have that $\hat{V}_{T-1}(x) \geq \mathbb{1}_{\bar{\mathcal{X}}}(x) \int_{\mathcal{X}} \hat{V}_T^*(z) Q(dz | x, u) \geq \mathbb{1}_{\bar{\mathcal{X}}}(x) \int_{\mathcal{X}} V_T^*(z) Q(dz | x, u)$ for all $(x, u) \in \mathcal{X} \times \mathcal{U}$. Moreover, the constraint $\hat{V}_{T-1}(x) \geq 1$, $\forall x \in K$ ensures that all feasible functions satisfy $\hat{V}_{T-1}(x) \geq \mathbb{1}_K(x)$ for all $x \in \mathcal{X}$. We conclude that for any feasible function in (8) it holds that $\hat{V}_{T-1}(x) \geq \mathbb{1}_K(x) + \mathbb{1}_{\bar{\mathcal{X}}}(x) \int_{\mathcal{X}} V_T^*(z) Q(dz | x, u)$, $\forall (x, u) \in \bar{\mathcal{X}} \times \mathcal{U}$ and it thus upper bounds $V_{T-1}^*(x)$. Using induction the same result follows for all $k = T - 2, \dots, 0$.

The final step of the process is to derive a tractable convex optimization program to construct a feasible solution to the problem in (8). We can then use it recursively to construct upper bounds to the value function for all $k = 0, \dots, T - 1$. First notice that from Lemma 4 and Assumption 2 both \hat{V}_T^* and $Q(\cdot | x, u)$ are Gaussian RBF sums and the right-hand-side of the first constraint in (8) is an RBF expectation that according to Lemma 5 can be written as another Gaussian RBF sum. As a result, by choosing the number of basis to construct $\bar{\mathcal{F}}$ according to

the number of elements in the RBF expectation in the constraints of (8), we can use the RBF dominance result in Lemma 3 to reformulate the constraints into sufficient semidefinite constraints. Notice however that depending on the number of RBF elements used to describe Q and the number of RBF elements used to approximate the value function at step $k = T$, the number of element required to build $\hat{\mathcal{F}}$ at every step of the recursion may increase, affecting the computational complexity of the process. The following proposition provides a tractable semidefinite program (SDP) to construct an upper bound to the reach-avoid value function.

Proposition 7 *Consider Lemma 4 and Assumption 2 and further assume that the sets $K \subset K'$ and $S := \bar{\mathcal{X}} \times \mathcal{U} = (K' \setminus K) \times \mathcal{U}$ are quadratic sets in \mathbb{R}^n and \mathbb{R}^{n+m} respectively. Let $\hat{V}_{k+1}^*(x) = \sum_{i=1}^M w_i \phi(x, \mu_i, \Sigma_i)$ be a known Gaussian RBF sum for some $M \in \mathbb{N}_+$ for which it holds that $\hat{V}_{k+1}(x) \geq V_{k+1}^*(x)$ for all $x \in \mathcal{X}$. Consider a density function for Q given by a single Gaussian RBF defined as $q(y, x, u) = \phi(y, \mu_0(x, u), \Sigma_0)$ with the function $\mu_0 : \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{X}$ defined as $\mu_0(x, u) = Ax + Bu$ where A, B are known matrices of appropriate dimensions. Let ν be a probability measure over \mathcal{X} and consider the following semidefinite program:*

$$\begin{aligned} & \min_{\hat{\mu}_i, \hat{\Sigma}_i, y_i, \tau_i, \rho_i} \sum_{i=1}^M y_i + \frac{1}{2} \text{tr}(\hat{\Sigma}_i) \\ & \text{subject to} \left. \begin{array}{l} X_i \succcurlyeq 0 \\ \bar{X}_i \succcurlyeq 0 \\ \tau_i, \rho_i \geq_{\mathbb{R}^N} 0 \\ \hat{\mu}_i \in \mathbb{R}^n \\ \hat{\Sigma}_i \in \mathbf{S}_+^n \end{array} \right\} i = 1, \dots, M \end{aligned} \quad (9)$$

where

$$X_i = \begin{bmatrix} Q_{\hat{\Sigma}_i}^{-1} & Q_{\hat{\mu}_i} \\ Q_{\hat{\mu}_i}^\top & Q_i + Q_{\mu_i, \bar{\Sigma}_i} - \sum_{j=1}^N \tau_{i,j} S_j \end{bmatrix}, \quad \bar{X}_i = \begin{bmatrix} Q_{\bar{\Sigma}_i}^{-1} & Q_{\bar{\mu}_i} \\ Q_{\bar{\mu}_i}^\top & \bar{Q}_i - \sum_{j=1}^N \rho_{i,j} K_j \end{bmatrix}$$

with

$$\begin{aligned} Q_i &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 2 \left(y_i + \log \left(\frac{\sqrt{|\hat{\Sigma}_i|}}{w_i} \right) \right) \end{bmatrix}, \quad Q_{\hat{\mu}_i} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & -\hat{\mu}_i \end{bmatrix}, \quad Q_{\hat{\Sigma}_i} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \hat{\Sigma}_i^{-1} \end{bmatrix}, \\ Q_{\mu_i, \bar{\Sigma}_i} &= \begin{bmatrix} A^\top \bar{\Sigma}_i A & A^\top \bar{\Sigma}_i B & -A^\top \bar{\Sigma}_i \mu_i \\ B^\top \bar{\Sigma}_i A & B^\top \bar{\Sigma}_i B & -B^\top \bar{\Sigma}_i \mu_i \\ -\mu_i^\top \bar{\Sigma}_i A & -\mu_i^\top \bar{\Sigma}_i B & \mu_i^\top \bar{\Sigma}_i \mu_i \end{bmatrix}, \quad \bar{Q}_i = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 2 \left(y_i + \log \left(\frac{M}{\sqrt{(2\pi)^n}} \right) \right) \end{bmatrix} \end{aligned}$$

and $y_i := \log \left(\frac{\hat{w}_i}{\sqrt{|\hat{\Sigma}_i|}} \right)$, $\bar{\Sigma}_i = (\Sigma_i + \Sigma_0)^{-1}$. Let $\{\hat{\mu}_i^*, \hat{\Sigma}_i^*, y_i^*, \tau_i^*, \rho_i^*\}_{i=1}^M$ denote the optimal solution of (9) and set $\hat{w}_i^* = e^{y_i^*} \sqrt{|\hat{\Sigma}_i^*|}$. The function constructed by $\hat{V}^*(x) = \sum_{i=1}^M \hat{w}_i^* \phi(x, \hat{\mu}_i^*, \hat{\Sigma}_i^*)$ is a feasible solution for (8).

PROOF. First we show how the constraints of problem (9) are sufficient for the constraints of (8). Given the format of functions \hat{V}_{k+1}^* and q we have from Lemma 5 that

$$\begin{aligned} \int_{\mathcal{X}} \hat{V}_T^*(z)Q(dz|x, u) &= \int_{\mathcal{X}} \sum_{i=1}^M w_i \phi(z, \mu_i, \Sigma_i) q(z, x, u) dz \\ &= \int_{\mathcal{X}} \sum_{i=1}^M w_i \phi(z, \mu_i, \Sigma_i) \phi(z, \mu_0(x, u), \Sigma_0) dz \\ &= \sum_{i=1}^M w_i \phi(\mu_0(x, u), \mu_i, \Sigma_i + \Sigma_0). \end{aligned}$$

Since $\hat{V}(x) = \sum_{i=1}^M \hat{w}_i \phi(x, \hat{\mu}_i, \hat{\Sigma}_i)$, the constraint $\hat{V}(x) \geq \int_{\mathcal{X}} \hat{V}_T^*(z)Q(dz|x, u)$ in (8) is equivalent to the RBF dominance constraint

$$\sum_{i=1}^M \hat{w}_i \phi(x, \hat{\mu}_i, \hat{\Sigma}_i) \geq \sum_{i=1}^M w_i \phi(\mu_0(x, u), \mu_i, \Sigma_i + \Sigma_0), \quad \forall (x, u) \in S$$

which according to Lemma 3 is implied by the constraints $X_i \succcurlyeq 0$ and $\tau_i \geq 0$ for $i = 1, \dots, M$. The form of the matrices $Q_i, Q_{\mu_i, \Sigma_i}, Q_{\hat{\mu}_i}, Q_{\hat{\Sigma}_i}$ is equivalent to the one derived in the proof of Lemma 3, considering the variable

$z = \begin{bmatrix} x \\ u \\ 1 \end{bmatrix}$, the function $\mu_0(x, u) = Ax + Bu$ and the transformation $y_i = \log\left(\frac{\hat{w}_i}{\sqrt{|\hat{\Sigma}_i|}}\right)$. We replace the second

constraint $\hat{V}(x) \geq 1, \forall x \in K$ by the sufficient condition $\sum_{i=1}^M \hat{w}_i \phi(x, \hat{\mu}_i, \hat{\Sigma}_i) \geq \sum_{i=1}^M \frac{1}{M}, \forall x \in K$ which is again an RBF dominance constraint that according to Lemma 3 is implied by the constraints $\bar{X}_i \succcurlyeq 0$ and $\tau_i \geq 0$ for $i = 1, \dots, M$. Finally, the cost function in (8) reads

$$\int_{\mathcal{X}} \hat{V}(x) \nu(dx) = \int_{\mathcal{X}} \sum_{i=1}^M \hat{w}_i \phi(x, \hat{\mu}_i, \hat{\Sigma}_i) dx = \sum_{i=1}^M \hat{w}_i = \sum_{i=1}^M e^{y_i} \sqrt{|\hat{\Sigma}_i|}$$

which we replace with the sum of the logarithms, i.e. $\sum_{i=1}^M \log\left(e^{y_i} \sqrt{|\hat{\Sigma}_i|}\right) = \sum_{i=1}^M y_i + \frac{1}{2} \log(|\hat{\Sigma}_i|)$ motivated by the fact that the logarithm is a monotone function. Still, the elements $\frac{1}{2} \log(|\hat{\Sigma}_i|)$ are concave in $\hat{\Sigma}_i$ and the minimization is a non-convex problem. Using the fact that $\frac{1}{2} \log(|\hat{\Sigma}_i|) = \frac{1}{2} \log(\prod_{d=1}^n \lambda_d^i) = \frac{1}{2} \sum_{d=1}^n \log(\lambda_d^i)$ and $\text{tr}(\hat{\Sigma}_i) = \sum_{d=1}^n \lambda_d^i$ we employ $\frac{1}{2} \text{tr}(\hat{\Sigma}_i)$ as a heuristic to approximate the terms $\frac{1}{2} \log(|\hat{\Sigma}_i|)$.

Feasibility of the problem in Proposition 7 is not easy to prove without additional assumptions. One way to guarantee a feasible solution is to add the constant function $c(x) = 1, \forall x \in \mathcal{X}$ in the basis set used to construct \mathcal{F} . Notice that we have expressed the kernel Q as a single Gaussian RBF to simplify the notation in (9); extensions to Gaussian RBF sums with varying functions $\mu_0(x, u)$ are straightforward.

The SDP-based procedure presented in Proposition 7 can be used recursively, starting at $k = T-1$ and $\hat{V}_T^*(x) = \hat{p}_K(x)$ for all $x \in \mathcal{X}$, to construct upper bound approximations to the value functions for every $k = 0, \dots, T-1$. Note that the process introduces conservatism due to the RBF dominance constraints used in (9) which are only sufficient conditions for the original robust constraints in (8). Moreover, the cost function in (8) is non-convex jointly in the weights, means and variances and the approximation used in Proposition 7 constitutes a design choice that can affect how close the bounding functions are to the optimal ones on different areas of the state space. Finally, the state-relevance measure is assumed to be a probability measure on \mathcal{X} but other choices allowing calculation of the integral $\int_{\mathcal{X}} \hat{V}(x) \nu(dx)$ as a function of the RBF parameters can be used (for example measures supported on hyper-rectangular sets - see [11]). Algorithm 1 summarizes the process of using Proposition 7 recursively to compute

Algorithm 1 Approximate value function

1: Input Data:

- Quadratic set constraints for $\bar{\mathcal{X}}$, K and \mathcal{U} .
- Reach-avoid time horizon T .
- Center and variance of the MDP kernel Q .
- Gaussian RBF sum \hat{p}_K for which it holds that $\hat{p}_K(x) \geq \mathbb{1}_K(x)$ for all $x \in \mathcal{X}$.

2: Design parameters:

- State-relevance measure ν .
- Approximation of the cost function $\int_{\mathcal{X}} \hat{V}(x) \nu(dx)$.

3: Initialize $\hat{V}_T^*(x) \leftarrow \hat{p}_K(x)$.

4: for $k = T - 1$ to $k = 0$ do

5: Extract the weights, means and variances $\{w_i, \mu_i, \Sigma_i\}_{i=1}^M$ from the known function \hat{V}_{k+1}^* .

6: Solve the optimization program in (9) to obtain $\{\hat{\mu}_i^*, \hat{\Sigma}_i^*, y_i^*\}_{i=1}^M$.

7: Compute $\hat{w}_i^* = e^{y_i^*} \sqrt{|\hat{\Sigma}_i^*|}$ for $i = 1, \dots, M$.

8: Set $\hat{V}_k^*(x) = \sum_{i=1}^M \hat{w}_i^* \phi(x, \hat{\mu}_i^*, \hat{\Sigma}_i^*)$.

9: end for

bounds down to time $k = 0$. Each SDP in the process has a total of $3M$ semidefinite constraints where the ones corresponding to X_i are of dimension $n + m + 1$, the ones corresponding to \bar{X}_i are of dimension $n + 1$ and the ones corresponding to $\hat{\Sigma}_i$ are of dimension n . If more than one Gaussian RBF elements are used to describe the process kernel Q , the total number of semidefinite constraints grows combinatorially in the elements of Q and M and linearly in the horizon length (see Lemma 5).

3.3 Approximate control policy

Given the value function V_k^* at each $k = \{0, \dots, T - 1\}$, the problem in (3) implicitly defines the optimal reach-avoid control policy at time k . Using the sequence of approximate value functions $\hat{V}_0^*, \dots, \hat{V}_{T-1}^*$ constructed using Proposition 7 recursively, one can compute approximate control policies $\hat{\pi}_k$ at each step $k = \{0, \dots, T - 1\}$. If the kernel Q has a Gaussian RBF sum density function $q(y, x, u) = \phi(y, Ax + Bu, \Sigma)$, as in Proposition 7, given $\hat{V}_{k+1}^*(x) = \sum_{i=1}^M w_i \phi(x, \mu_i, \Sigma_i)$, we define the SDP-based approximate control policy at each time step $k = \{0, \dots, T - 1\}$ as:

$$\begin{aligned}
 \hat{\pi}_k(x) &= \arg \max_{u \in \mathcal{U}} \left\{ \mathbb{1}_K(x) + \mathbb{1}_{\bar{\mathcal{X}}}(x) \int_{\mathcal{X}} \hat{V}_{k+1}^*(z) Q(dz | x, u) \right\} \\
 &= \arg \max_{u \in \mathcal{U}} \int_{\mathcal{X}} \hat{V}_{k+1}^*(z) Q(dz | x, u) \\
 &\stackrel{(*)}{=} \arg \max_{u \in \mathcal{U}} \sum_{i=1}^M w_i \phi(Ax + Bu, \mu_i, \Sigma_i + \Sigma)
 \end{aligned} \tag{10}$$

where we have used Lemma 5 to compute the expectation in $(*)$. Despite the fact that the optimization problem in (10) 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 functions can be analytically computed. 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 [14] 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. Alternative approaches include using randomized techniques similar to the approaches in [15,16,17,18], general purpose non-linear programming solvers (e.g. interior point method) or directly gridding the control space, if computationally feasible, as done in the second example of the next section.

4 Numerical examples

In this section we evaluate the SDP-based upper bounds calculated using the process of Algorithm 1. We first investigate the suboptimality of the bound and the performance of the associated control policies by comparing them to value functions and policies computed via space gridding. Since space gridding is limited to low dimensional

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 of $\sum_{i=1}^M w_i \phi(Ax_k + Bu, \mu_i, \Sigma_i + \Sigma)$ with respect to u .
 - 4: Solve the optimization problem in (10) to obtain the control input $\hat{\pi}_k(x_k)$.
 - 5: Apply the calculated control input to the system.
 - 6: **end for**
-

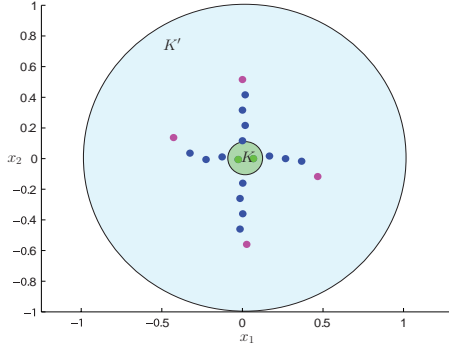


Fig. 1. Reach-avoid safe and target sets with example trajectories

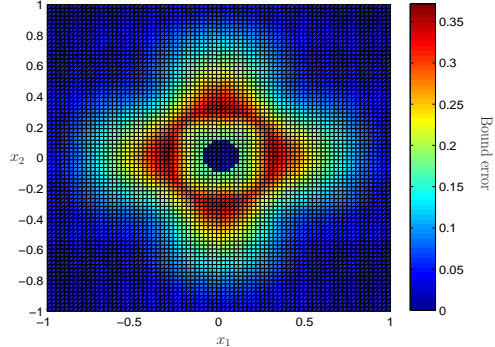


Fig. 2. Value function bound error at time $k = 0$

state-input spaces ($\dim(\mathcal{X} \times \mathcal{U})$ around 4-5), we also evaluate the bounding value functions on a simple benchmark problem where we assume an LQG controller is a suitable heuristic to maximize the reach-avoid probability. All simulations were carried out on an Intel Core i7 Q820 CPU clocked @ 1.73 GHz with 16GB of RAM memory, using MOSEK’s SDP solver in its default settings.

4.1 Reach-avoid probability bounds compared to space gridding

Consider the reach-avoid problem of maximizing the probability that the state of a controlled linear system subject to additive Gaussian noise reaches a target set around the origin within $T = 5$ discrete time steps while staying in a safe set for all previous times. We consider systems described by the equation

$$x_{k+1} = Ax_k + Bu_k + \omega_k \tag{11}$$

where for each $k = \{0, \dots, 4\}$, $x_k \in \mathcal{X} = \mathbb{R}^n$, $u_k \in \mathcal{U} = \{u \in \mathbb{R}^m | u^\top Q_u u \leq \rho_u^2\}$ with $\rho_u = 0.1$ and each ω_k is distributed as a Gaussian random variable $\omega_k \sim \mathcal{N}(\mathbf{0}_{n \times 1}, \Sigma_\omega)$ with $\Sigma_\omega \in \mathbf{S}_+^n$. We consider a target set $K = \{x \in \mathbb{R}^n | x^\top Q_t x \leq \rho_t^2\}$ with $\rho_t = 0.1$ around the origin and a safe set $K' = \{x \in \mathbb{R}^n | x^\top Q_s x \leq \rho_s^2\}$ with $\rho_s = 1$ (see Figure 1). In order to easily change the dimensions of the problem (variables n, m) and keep the system behaviour between problems similar, we fix $A = \mathbf{1}_{n \times n}$, $B = \mathbf{1}_{n \times m}$ and $\Sigma_\omega = 0.001 \cdot \mathbf{1}_{n \times n}$.

For problem dimensions up to $n = 3$ and $m = 2$, we compute the gridding-based value functions by discretizing the spaces $[-\rho_s, \rho_s]^n$ and $[-\rho_u, \rho_u]^m$ and solving the recursion in (2) on the resulting grid, initialized with $V_T^*(x) = \mathbf{1}_K(x)$. Table 1 reports the space discretization for each problem dimension (N_s corresponds to number of elements in each dimension of $[-\rho_s, \rho_s]$ and N_u corresponds to number of elements in each dimension of $[-\rho_u, \rho_u]$) along with the time it took to compute a single iteration (one horizon step) of the recursion. The upper bound value functions are constructed using the steps of Algorithm 1. The transition kernel of (11) has a Gaussian RBF sum density $x_{k+1} \sim \mathcal{N}(Ax_k + Bu_k, \Sigma_\omega)$ with center $Ax_k + Bu_k$ and covariance matrix Σ_ω while the sets K , K' and \mathcal{U} are by definition quadratic sets, satisfying the input requirements of the algorithm. As in Proposition 7, we choose a uniform state-relevance measure ν supported on \mathcal{X} . The only remaining element to execute the algorithm is to construct an upper bound function \hat{p}_K for the indicator function $\mathbf{1}_K$. We achieve this by randomly placing Gaussian RBF elements on K with diagonal covariance matrices $\Sigma_b = 0.0005$, choosing their weights by means of a linear program, making sure that they upper bound $\mathbf{1}_K$ on every point of the grid placed on \mathcal{X} (see [11] for details). Table 1 reports the number of elements used to upper bound $\mathbf{1}_K$ for the different cases of $\dim(\mathcal{X}) = 1, 2, 3$.

n	m	N_s	N_u	M	Grid (sec.)	SDP bound (sec.)	$\hat{V}_0^* - V_0^*$	$\hat{\pi} - \pi^*$
1	1	80	25	10	0.4262	0.62	0.34	0.035
2	1	80	25	50	115	1.88	0.174	-0.0186
2	2	80	25	50	1237	1.83	0.472	-0.087
3	1	30	15	90	477	4.8	0.053	0.0083
3	2	30	15	90	2926	4.15	0.186	0.031

Table 1
Comparison of grid-based and upper bound value functions

To estimate the performance of the approximate control policy, we sample 100 initial states x_0 uniformly from $\bar{\mathcal{X}}$ and for each one generate 100 different noise trajectory realizations, using the grid-based and SDP-based policies computed by (3) and Algorithm 2 respectively. We then count the number of trajectories that successfully complete the reach-avoid objective, i.e. reach K without leaving K' , making sure that $u \in \mathcal{U}$. In Table 1 we denote by $\hat{\pi} - \pi^*$ the mean difference between the empirical success probability of the SDP-based controller ($\hat{\pi}$) and the empirical success probability of the controller based on the value function computed via space gridding (π^*). The value of $\hat{V}_0^* - V_0^*$ corresponds to the mean difference between the values of the upper bound (\hat{V}_0^*) saturated to 1, and the gridding-based value function (V_0^*) at time $k = 0$, over the grid elements within the set \mathcal{X} . Analyzing the results, we notice that the approximate control policy based on the value function upper bound yields near optimal performance even though the average absolute error is non-zero. Overall, the SDP-based controller outperforms the grid-based controller, probably due to coarse discretization of \mathcal{X} and \mathcal{U} . In terms of computation time, the SDP-based approach significantly outperforms space gridding. The time values reported in Table 1 correspond to the computation time required to solve one step of the recursions; it is clear that the curse of dimensionality sets a fundamental limit to potential applications of space gridding.

4.2 High-dimensional problems

In this section we demonstrate the scalability of the SDP-based method by comparing the performance of the control policy computed using the value function bounds constructed with Algorithm 1 to an LQG controller. Instead of using Algorithm 2 to generate the control policy, we discretize the control space \mathcal{U} with 20 points in each dimension, resulting in a 20^m grid over which we compute the approximate control policy. We refrain from using non-linear programming solvers in this comparison as computation times grow significantly with the dimension of \mathcal{U} and are impractical for Monte Carlo-type verification simulations.

We consider the same regulation problem as in Section 4.1 but randomly choose the matrices $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ in every experiment such that the resulting systems are open loop stable and controllable under zero noise. If the system is unstable, the reach-avoid probability over a short horizon can be close to zero, especially when $\dim(\mathcal{X}) \gg \dim(\mathcal{U})$. For the series of experiments carried out in this part, we upper bound $\mathbf{1}_K$ in a different way than in Section 4.1. We use an optimization problem in the form of (9), without the constraints $X_i \succcurlyeq 0, \tau_i \geq 0$, thus getting an upper bound for $\mathbf{1}_K$. This bound will not be as tight as the one computed with the gridding method of the previous example and as a result, all subsequent value function bounds will be less accurate. Our purpose here, however, is to compare controller performance and hence any bound for $\mathbf{1}_K$ will work.

Despite having a different control objective, we assume that using an LQG controller where we choose the state and control penalty matrices according to the safe, target and control constraint sets is an efficient heuristic to maximize the reach-avoid probability. In particular we use the optimal control policy of the following problem:

$$\begin{aligned} \min_{\{u_k\}_{k=0}^{T-1}} \mathbb{E}_{w_k} \left(\sum_{k=0}^{T-1} x_k^\top Q x_k + u_k^\top R u_k \right) + x_T^\top Q x_T \\ \text{subject to} \quad (11), x_0 \in \bar{\mathcal{X}} \end{aligned} \quad (12)$$

where Q and R have been chosen to correspond to the quadratic sets K and \mathcal{U} respectively and penalize directions where they are small in size. 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 states for both LQG and SDP-based controllers, we compare their performance by generating 100 trajectories using the same noise samples, counting the number that reach K without leaving K' . To make sure that the initial states chosen have a non-zero

n	m	$\hat{\pi} - \pi_{LQG}$	LQG control (sec.)	SDP grid control (sec.)	SDP setup (sec.)
3	3	0.142	1.10e-04	0.0015	1.21
3	4	0.126	9.951e-05	0.0302	1.11
4	3	0.128	1.029e-04	0.0017	1.1
4	4	0.254	1.190e-04	0.0341	1.09
6	3	0.042	1.245e-04	0.0021	1.08
5	4	0.163	1.215e-04	0.0430	1.07
6	4	0.072	1.174e-04	0.0504	1.094
5	5	0.176	1.091e-04	0.846	1.07
6	5	0.144	1.256e-04	0.983	1.264
7	5	0.133	1.43e-04	1.14	1.33
8	5	0.141	1.33e-04	1.224	1.39
15	5	0.341	1.55e-04	2.3	1.27

Table 2
Comparison of LQG and SDP-based controllers based on recursion (2) and Algorithm 1.

reach-avoid probability we first run the LQG controller and reject states with too low success probabilities. Note that since the design criteria of the two methods are different, our comparison is qualitative despite the fact that they both drive the system to the origin. Table 2 reports the mean difference between the LQG and SDP-based controllers (denoted by $\hat{\pi} - \pi_{LQG}$) over the initial states for different problem dimensions. We notice that the SDP-based controller consistently outperforms the LQG controller in terms of success probability but scales worse in terms of computation time with the dimensions of the problem. The times reported for the LQG and SDP control, correspond to the computation of a single control input given the system state $x \in \mathcal{X}$. The last column of Table 2 reports the time required to construct an upper bound value function for one step of the reach-avoid recursion using Algorithm 1. Note that a Gaussian RBF sum with a single element is needed to upper bound $\mathbb{1}_K$ in each case ($M = 1$).

5 Conclusion

The method presented in this paper is suitable for constructing upper bounds for the reach-avoid value function of Markov decision processes whose stochastic kernel can be expressed as a sum of Gaussian radial basis functions (RBFs) with affine dependence on the state and control input. In the case where safe and target sets can be written as intersections of quadratic inequalities, we derive conditions that satisfy the robust constraints of an infinite dimensional optimization problem, equivalent to the reach-avoid dynamic programming recursion. We then provide a sequence of semidefinite programs to recursively construct upper bounds for the reach-avoid probability which are computationally efficient and scale to problems with much higher state and input dimensions than what can be handled in the literature. Our numerical examples indicate that the conditions used to reformulate the robust constraints may be conservative but the associated control policies perform close to optimal in low dimensional problems and considerably outperform a tuned LQG-based heuristic controller in higher dimensional problems.

As a part of future research we will investigate methods to reduce the conservatism introduced by the suggested constraint reformulation and couple the dominance conditions between Gaussian RBF sum functions. Moreover, it is worth comparing the suggested method to similar methods based on quadratic and polynomial approximations of value functions of general dynamic programs. We expect our method to provide a compromise between the scalability of quadratic approximations and the accuracy of general polynomial ones.

Acknowledgements

The authors would like to thank Marcello Colombino and Andreas Hempel from the Automatic Control Laboratory in ETH Zurich for discussions around the problem formulation. Special thanks to Mr. Colombino for his help in the reformulation of the constraints in the upper bound semidefinite program.

References

- [1] 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.
- [2] 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.
- [3] 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.
- [4] Prandini, M., and Hu, J., 2006. “Stochastic reachability: Theory and numerical approximation”. *Stochastic hybrid systems, Automation and Control Engineering Series*, **24**, pp. 107–138.
- [5] Kushner, H. J., and Dupuis, P., 2001. *Numerical methods for stochastic control problems in continuous time*, Vol. 24. Springer.
- [6] de Farias, D. P., and Van Roy, B., 2003. “The linear programming approach to approximate dynamic programming”. *Operations Research*, **51**(6), pp. 850–865.
- [7] Wang, Y., O’Donoghue, B., and Boyd, S., 2014. “Approximate dynamic programming via iterated bellman inequalities”. *International Journal of Robust and Nonlinear Control*.
- [8] Bertsimas, D., and Sim, M., 2006. “Tractable approximations to robust conic optimization problems”. *Mathematical Programming*, **107**(1-2), pp. 5–36.
- [9] Boyd, S. P., and Vandenberghe, L., 2004. *Convex optimization*. Cambridge university press.
- [10] Kariotoglou, N., Summers, S., Summers, T., Kamgarpour, M., and Lygeros, J., 2013. “Approximate dynamic programming for stochastic reachability”. In Control Conference (ECC), 2013 European, IEEE, pp. 584–589.
- [11] Kariotoglou, N., Summers, T., Kamgarpour, M., and Lygeros, J., 2014. “A numerical approach to stochastic reach-avoid problems for markov decision processes”. *arxiv*.
- [12] 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.
- [13] 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.
- [14] Waechter, A., Laird, C., Margot, F., and Kawajir, Y., 2009. Introduction to IPOPT: A tutorial for downloading, installing, and using IPOPT.
- [15] Calafiore, G., and Campi, M., 2006. “The scenario approach to robust control design”. *IEEE Transactions on Automatic Control*, **51**(5), pp. 742–753.
- [16] Campi, M. C., and Garatti, S., 2008. “The exact feasibility of randomized solutions of uncertain convex programs”. *SIAM Journal on Optimization*, **19**(3), pp. 1211–1230.
- [17] 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.
- [18] Deori, L., Giulioni, 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.