Piecewise Barrier Functions for Stochastic Systems

Rayan Mazouz [a], Frederik Baymler Mathiesen [b], Luca Laurenti [b], Morteza Lahijanian [a]

[a] *Department of Aerospace Engineering Sciences, University of Colorado Boulder, USA*

[b] *Delft Center for Systems and Control, Delft University of Technology, The Netherlands*

Abstract

This paper proposes a novel framework to compute probabilistic safety guarantees for nonlinear stochastic systems based on a Piecewise Barrier (PWB) approach. The general theory of stochastic PWBs with general noise distributions is provided. For high efficiency and scalability purposes, the proposed methods are based on the class of constant piecewise functions. In particular, three novel and efficient methods for constructing a Piecewise Constant (PWC) barrier are proposed. These are based on (1) dual Linear Programming (LP), (2) Counter-Example Guided Synthesis (CEGS), and (3) Gradient Descent (GD). The efficacy of the methods are evaluated on various verification and robotics benchmarks, which include neural network dynamic models, nonlinear switched systems and high-dimensional linear systems. Extensive experimental studies, illustrating the performance of the proposed methods against state-of-the-art Sum-of-Squares (SOS) and Neural Barrier Function (NBF) synthesis, show-case the efficacy and superiority of this novel approach. Piecewise barriers provide a new path to ensuring safety for stochastic systems in the joint disciplines of robotics, formal methods, control theory, and machine learning.

Key words: Safety Certificate, Stochastic Nonlinear Systems, Piecewise Barrier Functions, Convex Optimization, Robotics

1 Introduction

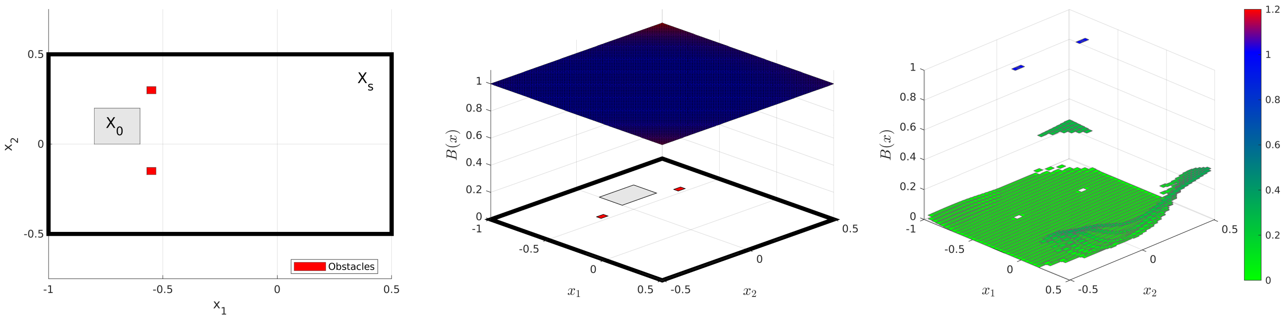
The deployment of autonomous systems in *safety-critical* applications, examples spanning medical robotics [1] and autonomous vehicles [2], requires vital robustness guarantees. The behavior of such systems is often ambiguous, which is attributed to the inherent complexity and nonlinearities of the dynamics, subject to physics and algorithm uncertainty. Nonetheless, for real-life deployment, these systems require formal safety guarantees, making stochasticity and verification a major aspect of system design. Consequently, the question *how to ensure safety for stochastic systems?* has emerged as a central research problem in disciplines that range from robotics, formal methods, control theory to machine learning [3–6].

In the general sense, safety for stochastic systems generalizes the standard notion of stochastic stability [3]. This is formalized as the probability of not exhibiting unsafe behavior, of which an exact computation is generally infeasible. A very common approach to verify the safety of stochastic systems, by providing a lower bound on the probability of safety, is based on *stochastic barrier functions* [4,7,8]. State-of-the-art methods to synthesize these barrier functions rely on Sum-of-Squares (SOS) optimization [5, 8] and Neural Barrier Function (NBF) training [6, 9]. The NBF approach is mainly limited by the required learning process and the scalability of the verification. At large, the SOS approach is limited due to (i) a very specific class of required SOS functions, (ii) restriction to use of linear or polynomial dynamics [8], and (iii) the large scope of the required convex optimization due to the construction and conversion to Semi-Definite Programming. This makes scaling to larger dimensional systems intractable [10]. Further, as illustrated through Example (1), SOS exhibits major difficulties with non-convex sets and non-symmetric initial set placement.

Example 1 *This example presently show-cases major limitations of the SOS approach through a straightforward running example in* $\mathbb{R}^2$*. The dynamics are linear and the distribution of the random variable is Gaussian, which is an ideal setup for SOS optimization. However, obstacles are placed within the domain as depicted in Fig.* (1)*, making the set non-convex. As such, the SOS approach has difficulties fitting a function, forcing the SOS barrier to be greater than one in the entire state space, as is evident from Fig.* (1b)*. This phenomenon does not improve for a high SOS barrier degree, as it is an inherent SOS limitation. Consequently, SOS reports a lower bound on the safety probability of* $\approx 7\%$ *(barrier degree*

[⋆] Corresponding author name

Email address: rayan.mazouz@colorado.edu (Rayan Mazouz).



(a) X_s and X_0

(b) Degree-8 SOS-polynomial SBF

(c) PWC SBF

Fig. 1. Example of a 2D stochastic system with non-convex safe set and dynamics $\mathbf{x}_{k+1} = 0.5\mathbf{x}_k + \mathbf{v}$, where noise $\mathbf{v}$ has a Gaussian distribution. (a) shows the safe and initial sets, and (b) and (c) show SOS (degree 8) and PWC SBFs, respectively.

30, 10 time-steps), which is overly conservative for a system known to be inherently safe. Our novel piecewise methods assign a barrier value of 1 to the obstacle regions, and optimize over the remainder of the state space, resulting in the barrier as depicted in Fig. (1c)*. The difference is likewise highly notable in terms of the verified safety probability, as the PWC barriers report* $\approx$ 93% *(10 time-steps) as the lower probabilistic safety bound.*

In this paper, we propose a new framework for the safety verification of stochastic systems, based on a novel *piecewise continuous stochastic barrier* approach. The fundamental methodology levitates the major limitations exhibited by SOS and NBFs. In summary, the main contributions of the paper are:

- A general derivation of stochastic piecewise barrier theory for nonlinear stochastic systems with non-standard (e.g., non-affine, non-symmetric, non-unimodal) noise distributions.
- A derivation showing the equivalence of piecewise linear and piecewise constant barriers under interval transition probabilities.
- Three novel convex optimizations methods for constructing piecewise constant barriers: a Linear Programming (LP) duality-based approach, a Counter Example Guided Synthesis (CEGS) approach, and a Gradient Descent (GD) approach.
- Safety certification of stochastic neural network dynamic models, switched nonlinear systems and high-dimensional linear systems with convex and non-convex state spaces.
- Extensive case-studies to illustrate the performance of the proposed methods against state-of-the-art SOS and NBFs, and an analysis hereof.

Related Work Two of the most commonly employed approaches for the verification of stochastic systems are *stochastic barrier functions* [7,8] and *abstraction-based methods* [11,12]. The latter abstract the original system into a finite, discrete-space stochastic process, a variant of a Markov chain, for which efficient algorithms exist that compute the probability of safety [13,14].

Stochastic barrier functions, on the other hand, were first proposed in [3] to study the safety probability of a stochastic system based on martingale theory. A major benefit of using barriers for safety analysis over other methods is, given their static properties, they allow making probabilistic statements on the time evolution of the system without evolving the system. In recent years, verification of stochastic systems has emerged as a prominent research field for the joint disciplines of robotics [8], machine learning [5,6], and control theory [4,15].

2 Problem Formulation

We consider a stochastic process described by the following stochastic difference equation

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{v}_k), \tag{1}$$

where state $\mathbf{x}_k$ takes values in $\mathbb{R}^n$, and noise $\mathbf{v}_k$ is an independent and identically distributed random variable taking values in $\mathbb{R}^{\mathrm{v}}$, with associated continuous probability distribution $p_{\mathbf{v}}$. Further, $f : \mathbb{R}^n \times \mathbb{R}^{\mathrm{v}} \to \mathbb{R}^n$ is the vector field representing the one step dynamics of System (1). We assume that f is almost everywhere continuous. Intuitively, $\mathbf{x}_k$ is a general model of a stochastic process with possibly nonlinear dynamics and non-additive noise.

In order to define a probability measure for System (1), for a measurable set X we define the one-step stochastic kernel $T(X \mid x)$ as

$$T(X \mid x) := \int_{\mathbb{R}^{\mathrm{v}}} \mathbf{1}_X(f(x,v))p_{\mathbf{v}}(dv), \tag{2}$$

where $\mathbf{1}_X$ is the indicator function for set X, defined as

$$\mathbf{1}_X(x) = \begin{cases} 1 & \text{if } x \in X \\ 0 & \text{otherwise.} \end{cases}$$

It follows that for a given time horizon $\{0, ..., N\} \subset \mathbb{N}$ and an initial condition $x_0 \in \mathbb{R}^n$, a well-defined probability measure $\Pr^{x_0}$ can be associated to $\mathbf{x}_k$ [16], which is uniquely defined by T such that for measurable sets $X_0, X_k \subseteq X$, it holds that

$$\Pr^{x_0}[\mathbf{x}_0 \in X_0] = \mathbf{1}_{X_0}(x_0),$$
$$\Pr^{x_0}[\mathbf{x}_k \in X_k \mid \mathbf{x}_{k-1} = x] = T(X_k \mid x).$$

The definition of $\Pr^{x_0}$ allows one to make probabilistic statements over the trajectories of System (1). This work seeks to quantify *probabilistic safety*, which together together with its dual, *probabilistic reachability* [17], are commonly used to quantify safety for stochastic dynamical systems and [18] represent a generalization of the notion of invariance.

Definition 1 (Probabilistic Safety) *Let $X_\mathrm{s} \subset \mathbb{R}^n$ be a bounded set representing the safe set, $X_0 \subseteq X_\mathrm{s}$ be the initial set, and $N \in \mathbb{N}$ be the time horizon. Then,* probabilistic safety *is defined as*

$$P_\mathrm{s}(X_\mathrm{s}, X_0, N) = \inf_{x_0 \in X_0} \Pr^{x_0}[\forall k \in \{0, ..., N\}, \mathbf{x}_k \in X_\mathrm{s}].$$

The problem we consider in this work is as follows.

Problem 1 (Safety Certificate) *Consider a safe set $X_\mathrm{s} \subset \mathbb{R}^n$ and an initial set $X_0 \subseteq X_\mathrm{s}$. Then, for a given threshold $\delta_\mathrm{s} \in [0, 1]$, certify whether, starting from X_0, System* (1) *remains in X_s for N time steps with at least probability δ_s, i.e.,*

$$P_\mathrm{s}(X_\mathrm{s}, X_0, N) \geq \delta_\mathrm{s}.$$

Problem 1 seeks to compute the probability that $\mathbf{x}_k$ remains within a given safe set, e.g., it avoids obstacles and undesirable states. Computation of this probability is particularly challenging because System (1) is stochastic and f can be a nonlinear (and non-polynomial) function. Further, since set X_s can be non-convex, the problem is generally non-convex. This implies that convex optimization tools cannot be used out-of-the-box for this problem in an efficient way.

Approach overview Our approach to Problem 1 is based on stochastic barrier functions. To address the inefficiencies of existing synthesis techniques for continuous barrier functions due to, e.g., non-convexity of X_s, we focus on *piecewise* functions. We show that, based on such functions, barrier certificates can be provided for the safety of System (1) by establishing a lower bound on P_s. Further, we show that by choosing a piecewise constant function, the synthesis problem reduces to a linear optimization problem, for which we introduce three highly efficient computational frameworks.

3 Stochastic Barrier Functions

In this section, we provide an overview of stochastic barrier functions.

Consider stochastic System (1) and safe set $X_\mathrm{s} \subset \mathbb{R}^n$, initial set $X_0 \subseteq X_\mathrm{s}$, and unsafe set $X_\mathrm{u} = \mathbb{R}^n \setminus X_\mathrm{s}$. Then, a function $B : \mathbb{R}^n \to \mathbb{R}$ is a *stochastic barrier function* (SBF) if there exist scalars $\eta, \beta \geq 0$ such that

$$\begin{aligned}
&B(x) \geq 0 && \forall x \in \mathbb{R}^n, && \text{(3a)}\\
&B(x) \geq 1 && \forall x \in X_\mathrm{u}, && \text{(3b)}\\
&B(x) \leq \eta && \forall x \in X_0, && \text{(3c)}\\
&\mathbb{E}[B(f(x, \mathbf{v}))] \leq B(x) + \beta && \forall x \in X_\mathrm{s}, && \text{(3d)}
\end{aligned}$$

where $\mathbf{v} \in \mathbb{R}^n$ is a random variable with density $p_\mathbf{v}$. Given an SBF B, then a lower bound on the probability of safety for System (1) can be obtained according to the following proposition.

Theorem 1 ([17, Theorem 2]) *If there exists function B that satisfies Conditions* (3a)-(3d)*, then for a horizon $N \in \mathbb{N}$, it follows that*

$$P_\mathrm{s}(X_\mathrm{s}, X_0, N) \geq 1 - (\eta + \beta N). \tag{4}$$

Note that Equation (4) provides a lower bound on the probability of remaining in the safe set. Hence, B is called a barrier certificate if the right hand side of Equation (4) is greater than or equal to threshold δ_s, i.e.,

$$1 - (\eta + \beta N) \geq \delta_\mathrm{s}.$$

A major benefit of using barrier certificates for safety analysis is that static Conditions (3a)-(3d) enable probabilistic reasoning on the time evolution of the stochastic system without the need to evolve the system. A popular approach to finding an SBF B is to limit the search to the set of SOS polynomials of a given degree. Then, a SOS optimization problem can be formulated to synthesize the parameters of the barrier polynomial, where the objective function is $\min \eta + N\beta$, subject to Conditions (3a)-(3d) [5, 7, 8]. The SOS approach however becomes very conservative when X_s is non-convex. Recent techniques address this issue by using the power of neural networks as universal approximators to learn a neural barrier function (NBF) [6, 9]. Nevertheless, while training NBFs can be done efficiently, checking Conditions (3a)-(3d) against the network is challenging, which limits scalability of NBFs. To tackle these challenges, we propose piecewise barrier functions below.

4 Piecewise Stochastic Barrier Functions

In this paper, we introduce the notion of piecewise SBF (PW-SBF). Consider a partition $X_1, \ldots, X_K$ of safe set

X_s in K compact sets, i.e.,

$$\bigcup_{i=1}^{K} X_i = X_\mathrm{s} \quad \text{and} \quad X_i \cap X_j = \emptyset,$$

for all $i \neq j \in \{1, \dots, K\}$, such that vector field f is continuous in each region X_i and we further assume that the boundary of each set has measure zero w.r.t. $T(\cdot \mid x)$ for any $x \in X_s$. Further, let $B_i : X_i \to \mathbb{R}$ be a real-valued continuous function for every $i \in \{1, \dots, K\}$. We define piecewise (PW) function

$$B(x) = \begin{cases} B_i(x) & \text{if } x \in X_i \\ 1 & \text{otherwise.} \end{cases} \tag{5}$$

The following corollary characterizes the general result of Theorem 1 for the PW function B.

Corollary 1 (Piecewise SBF) *Piecewise function $B(x)$ in Equation (5) is a stochastic barrier function for System (1) if $\forall i \in \{1, \dots, K\}$ there exist scalars $\beta_i, \eta \geq 0$ such that*

$$B_i(x) \geq 0 \qquad \forall x \in X_i, \tag{6a}$$

$$B_i(x) \leq \eta \qquad \forall x \in X_i \cap X_0, \tag{6b}$$

$$\sum_{j=1}^{K} \mathbb{E}[B_j(\mathbf{x}') \mid \mathbf{x}' \in X_j] \cdot T(X_j \mid x) + T(X_\mathrm{u} \mid x) \leq B_i(x) + \beta_i \qquad \forall x \in X_i, \tag{6c}$$

where $\mathbf{x}' = f(x, \mathbf{v})$ and T is the transition kernel in Equation (2). Then, it holds that

$$P_\mathrm{s}(X_\mathrm{s}, X_0, N) \geq 1 - (\eta + N \cdot \max_{i \in \{1,\dots,K\}} \beta_i). \tag{7}$$

Proof. The proof builds on Theorem 1. It is clear that PW function B in (5) under Conditions (6a)-(6b) satisfies Conditions (3a)-(3c). Then, it is enough to show that $\max_{i\in\{1,\dots,K\}} \beta_i$ from Condition (6c) upper bounds $\mathbb{E}[B(\mathbf{x}')] - B(x)$ in Condition (3d). For $x \in X_i$ and PW $B(x)$, we have

$$\mathbb{E}[B(f(x, \mathbf{v}))] - B(x) \tag{8}$$

$$= \sum_{j=1}^{K} \mathbb{E}[B_j(f(x, \mathbf{v})) \mid f(x, \mathbf{v}) \in X_j] \cdot T(X_j \mid x) + 1 \cdot T(X_\mathrm{u} \mid x) - B_i(x) \tag{9}$$

$$\leq \beta_i, \tag{10}$$

where the equality in (9) holds by the law of total expectation, $B(x' \in X_\mathrm{u}) = 1$, and the inequality in (10) holds by Condition (6c). Hence, for every $x \in X_\mathrm{s} = \bigcup_{i=1}^{K} X_i$, the expression in (8) is $\leq \max_{i\in\{1,\dots,K\}} \beta_i$. □

Corollary 1 enables one to formulate an optimization problem to synthesize a PW-SBF. Specifically, the objective is to find B_is that minimize $(\eta + N \cdot \max\{\beta_i\}_{i=1}^{K})$ subject to Conditions (6a)-(6c). The benefit of this formulation is that both the size of the partition and type of functions B_is (e.g., linear, polynomial, exponential, etc.) are design parameters. This provides a lot of flexibility for the SBF to fit different shapes of X_s. This flexibility, however, may introduce challenges as it can lead to a non-convex optimization problem, even for simple choices for B_i such as polynomial or linear functions. We introduce a set of simple but effective choices that lend themselves to efficient computational tools that outperform the state of the art SBF synthesis techniques.

A major difficulty in the optimization problem results from the product of the expectation term and the transition kernel function in Condition (6c), namely,

$$\mathbb{E}[B_j(\mathbf{x}') \mid \mathbf{x}' \in X_j] \cdot T(X_j \mid x).$$

The first term not only requires to perform an expectation operation, but also a composition of B_j with the nonlinear function f. The second term T is a nonlinear function of x that involves an integral of probability density function $p_\mathbf{v}$. To reduce complexity, one can choose to use constant values for B_is, which simply avoids the need to perform the expectation and composition operations. This leads to a PW Constant (PWC) SBF. Furthermore, while the analytical form of T may be hard to obtain, its bounds can be efficiently computed using, e.g., the procedure in [19], for general f and non-standard $p_\mathbf{v}$ (e.g., non-symmetric, non-unimodal, etc.). In the next section, we detail the optimization problem for these choices.

5 Piecewise Constant SBF Synthesis

In this section, we first formally set up an optimization problem for synthesis of PWC-SBFs. Then, we introduce three efficient computational methods to solve the optimization problem, namely, a LP duality-based approach, a CEGS procedure, and a GD-based method.

5.1 PWC-SBF Synthesis

For $i, j \in \{1, \dots, K\}$, let $\underline{p}_{ij}, \overline{p}_{ij} \in [0, 1]$ denote the lower and upper bounds of the transition kernel $T(X_j \mid x)$ for every $x \in X_i$, respectively, i.e.,

$$\underline{p}_{ij} \leq T(X_j \mid x) \leq \overline{p}_{ij} \qquad \forall x \in X_i. \tag{11}$$

Similarly, we use $\underline{p}_{i\mathrm{u}}, \overline{p}_{i\mathrm{u}} \in [0, 1]$ for the bounds of T to the unsafe set X_u, i.e.,

$$\underline{p}_{i\mathrm{u}} \leq T(X_\mathrm{u} \mid x) \leq \overline{p}_{i\mathrm{u}} \qquad \forall x \in X_i. \tag{12}$$

Note that these bounds can be computed efficiently for general f and $p_{\mathbf{v}}$ by using, e.g., techniques in [14,19–21]. We define the set of all feasible values for the transition kernel for all $x \in X_i$ as

$$\begin{aligned}\mathcal{P}_i = \Big\{ & p_i = (p_{i1}, \ldots, p_{iK}, p_{i\mathrm{u}}) \in [0,1]^{K+1} \quad s.t. \\ & \sum_{j=1}^{K} p_{ij} + p_{i\mathrm{u}} = 1, \\ & \underline{p}_{ij} \leq p_{ij} \leq \overline{p}_{ij} \quad \forall j \in \{1, \ldots, K, \mathrm{u}\} \Big\}. \end{aligned} \tag{13}$$

The following theorem sets up an optimization problem for synthesis of a PWC-SBF.

Theorem 2 (PWC-SBF Synthesis) *Given a K-partition of X_s, let $\mathcal{B}_K$ be the set of PWC functions in the form of Equation* (5) *with $B_i(x) = b_i \in \mathbb{R}$ for every $i \in \{1, \ldots, K\}$, and let $\mathcal{P} = \mathcal{P}_1 \times \ldots \times \mathcal{P}_K$, where each $\mathcal{P}_i$ is the set of probability distributions defined in Equation* (13)*. Then, $B^* \in \mathcal{B}_K$ is a PWC-SBF that maximizes RHS of Equation* (7) *if B^* is a solution to the following optimization problem*

$$B^* = \arg \min_{B \in \mathcal{B}_K} \max_{(p_i)_{i=1}^K \in \mathcal{P}} \eta + N\beta \tag{14}$$

subject to

$$\begin{aligned} & b_i \geq 0 && \forall i \in \{1, \ldots, K\}, && \text{(15a)} \\ & b_i \leq \eta && \forall i : X_i \cap X_0 \neq \emptyset, && \text{(15b)} \\ & \sum_{j=1}^{K} b_j \cdot p_{ij} + p_{i\mathrm{u}} \leq b_i + \beta_i && \forall i \in \{1, \ldots, K\}, && \text{(15c)} \\ & 0 \leq \beta_i \leq \beta && \forall i \in \{1, \ldots, K\}. && \text{(15d)} \end{aligned}$$

Proof. It suffices to show that if for all i, $B_i(x)$ satisfy Conditions (15a)-(15c), then PW function $B(x)$ as defined in Eq. (5) satisfies Conditions (6a) - (6c). The optimization problem in Equation (14) aims to maximize the safety probability over feasible values for the transition kernel $\mathcal{P}_i$, which is expressed as a minimax problem. Each β_i is bounded according to constraint (15c), and by Corollary 1, $\max_{i \in \{1,\ldots,K\}} \beta_i$ upper bounds $\mathbb{E}[B(\mathbf{x}')] - B(x)$. Hence, optimizing over objective B^* also optimizes the right hand side of (7), thereby maximizing the lower bound on safety probability $P_\mathrm{s}(X_\mathrm{s}, X_0, N)$. □

Given a partition of X_s, Theorem 2 provides a method to synthesize a PWC-SBF that optimizes a lower bound on $P_\mathrm{s}(X_\mathrm{s}, X_0, N)$. This theorem immediately gives rise to questions on optimality (w.r.t. general SBFs) and computibility. We address the computibility question in Section 5.2. For optimality, the following proposition establishes that, in the limit of a partition of size large enough, a PWC-SBF converges to a value of safety smaller or equal to the optimal safety probability $\underline{P}^*_\mathrm{s}(X_\mathrm{s}, X_0, N)$ obtainable with continuous SBFs.

Proposition 1 *Let $\underline{P}^*_\mathrm{s}(X_\mathrm{s}, X_0, N)$ denote the optimal safety probability obtained from Theorem 1 from the class of continuous SBFs, i.e., the set of continuous functions satisfying Conditions* (3a)*-*(3d)*. Consider a uniform partition of X_s in K compact sets and call $(\eta + N\beta)^*_K$ the resulting optimal safety probability bound obtained from solving Theorem 2. Then, it holds that*

$$\underline{P}^*_\mathrm{s}(X_\mathrm{s}, X_0, N) \leq \lim_{K \to \infty} (1 - (\eta + N\beta)^*_K).$$

Proof. First of all, we need to show that for all $x \in X_u$, the optimal choice of the barrier is always $B(x) = 1$. In order to do that note that

$$\mathbb{E}[B(f(x, \mathbf{v}))] = \int_{v : f(x,v) \in X_\mathrm{s}} B(f(x,v)) p_{\mathbf{v}}(dv) + \int_{v : f(x,v) \in X_\mathrm{u}} B(f(x,v)) p_{\mathbf{v}}(dv).$$

Using this fact, we can rewrite Condition (3d) as the fact that $\forall x \in X_\mathrm{s}$ it must hold that

$$\int_{v : f(x,v) \in X_\mathrm{s}} B(f(x,v)) p_{\mathbf{v}}(dv) + \int_{v : f(x,v) \in X_\mathrm{u}} B(f(x,v)) p_{\mathbf{v}}(dv) \leq B(x) + \beta N.$$

As for $x \in X_u$, $B(x)$ only appear on the left hand side of the inequality it follows that the value of β required to satisfy the inequality is minimized by taking the smallest possible value of $B(x)$ for $x \in X_u$. Because of Condition 3b, the smallest possible value is 1. The rest of the proof follows by the fact that within X_s piece-wise constant SBF are dense wrt to the set of continuous SBFs.□

This proposition shows that, despite their simplicity, PWC-SBFs are as expressive as complex forms of SBFs such as polynomials and NBFs. That is, PWC-SBFs can compute the optimal safety probability as well as or even better than continuous SBFs. In fact, our evaluations in Section 6 clearly demonstrate this point. Further, we also note that, from the proof of Proposition 1, it becomes evident that the choice of $B(x) = 1$ for all $x \in X_\mathrm{u}$ in Equation (5) is the optimal choice for PW-SBFs.

5.2 Computation of PWC-SBFs

The optimization problem in Theorem 2 is a minimax problem with the decision variables

$$
\begin{aligned}
&b = (b_1, \dots, b_K) \in \mathbb{R}^K_{\geq 0}, && && (16a)\\
&p_i = (p_{i1}, \dots, p_{iK}, p_{i\mathrm{u}}) \in \mathcal{P}_i && \forall i \in \{1, \dots, K\}, && (16b)\\
&\beta_i \in \mathbb{R}_{\geq 0} && \forall i \in \{1, \dots, K\}, && (16c)\\
&\eta, \beta \in \mathbb{R}_{\geq 0}. && && (16d)
\end{aligned}
$$

A major difficulty in computability of this optimization problem is in Constraint (15c), which includes products of decision variables b_j and p_{ij}. As a consequence, the optimization problem in Theorem 2 is bilinear, meaning that it is linear if either b_j or p_{ij} are fixed. This class of problems is generally non-convex, for which convex solvers cannot be applied for efficiency and guaranteed convergence. We propose three approaches to convexify the problem (lossless) such that simple Linear Programs (LPs) and Gradient Decent (GD) methods can be used to compute the optimal solution.

5.2.1 Dual Linear Program

We first introduce a dual LP approach that exactly solves the optimization problem in Theorem 2 by encoding the bounds on transition kernel T using duality.

We begin by observing that the set of feasible transition kernels $\mathcal{P}_i$ in Equation (13) is a simplex. Hence, it can be represented as

$$\mathcal{P}_i = \{p_i : H_i\, p_i \leq h_i\}, \tag{17}$$

where matrix $H_i \in \mathbb{R}^{2(K+1)\times(K+1)}$ and vector $h_i \in \mathbb{R}^{2(K+1)}$ are defined by the constraints in Equation (13), and inequality relation "$\leq$" is interpreted element-wise on vectors. Next, we define vector $\bar{b} = (b, 1)$, where b is in Equation (16a), and dual variable $\lambda_i \in \mathbb{R}^{2(K+1)}_{\geq 0}$. Then, Constraint (15c) can be re-written as two constraints

$$
\begin{aligned}
h_i^\top \lambda_i &\leq b_i + \beta_i, && (18a)\\
H_i^\top \lambda_i &= \bar{b}. && (18b)
\end{aligned}
$$

Since both constraints are linear in the decision variables λ_i, b, and β_i, the optimization problem in Theorem 2 can be written as an LP. The following lemma and theorem show the LP formulation and that it exactly solves the optimization problem in Theorem 2.

Lemma 1 (Zero Duality Gap) *Consider the following two optimization problems with decision variables $z = (b_1, \dots, b_K, \beta_i)$ and λ_i and feasible transition kernel set $\mathcal{P}_i = \{p_i : H_i p_i \leq h_i\}$*

$$
\begin{aligned}
\min_{z} \quad & \beta_i \\
\text{s.t.} \quad & \bar{b}^\top p_i \leq b_i + \beta_i \qquad \forall p_i \in \mathcal{P}_i
\end{aligned}
\tag{19}
$$

and

$$
\begin{aligned}
\min_{z,\lambda_i} \quad & \beta_i \\
\text{s.t.} \quad & h_i^\top \lambda_i \leq b_i + \beta_i \\
& H_i^\top \lambda_i = \bar{b}, \\
& \lambda_i \geq 0.
\end{aligned}
\tag{20}
$$

Let z_1^* and (z_2^*, λ_i^*) be optimal solutions to the Problems (19) and (20), respectively. Then, $z_1^* = z_2^*$ holds.

Proof. The constraint $\bar{b}^T p_i \leq b_i + \beta_i$, for all $p_i \in \mathcal{P}_i$ can be written as an inner optimization problem

$$\begin{pmatrix} \max\limits_{p_i \in \mathcal{P}_i} & \bar{b}^T p_i \\ \text{s.t.} & H_i p_i \leq h_i \end{pmatrix} \leq b_i + \beta_i.$$

Due to strong duality of LP [22, Chapter 5], we can substitute the inner problem for its equivalent (asymmetric) dual problem

$$\begin{pmatrix} \min\limits_{\lambda_i \geq 0} & h_i^\top \lambda_i \\ \text{s.t.} & H_i^\top \lambda_i = \bar{b} \end{pmatrix} \leq b_i + \beta_i.$$

Since the inner problem is a minimization, it can be lifted out into the outer problem to become the constraints in Problem (20). □

As such, we introduce an exact dual LP approach to solve the problem in Theorem (2).

Theorem 3 (PWC-SBF Dual LP) *An optimal solution to the following LP is an optimal solution to the optimization problem in Theorem 2*

$$\min \eta + \beta N$$

subject to

$$
\begin{aligned}
& 0 \leq b_i \leq 1 && \forall i \in \{1, \dots, K\}, \\
& b_i \leq \eta && \forall i : X_i \cap X_0 \neq \emptyset, \\
& h_i^\top \lambda_i \leq b_i + \beta_i && \forall i \in \{1, \dots, K\}, \\
& H_i^\top \lambda_i = \bar{b} && \forall i \in \{1, \dots, K\}, \\
& \lambda_i \geq 0 && \forall i \in \{1, \dots, K\}, \\
& 0 \leq \beta_i \leq \beta && \forall i \in \{1, \dots, K\}.
\end{aligned}
$$

Proof. By Lemma 1, the dual is equivalent to the primal problem. Solving this robust linear program thus results in an exact solution to the optimization problem in Theorem 2. Hence, piecewise $B(x)$ constitutes a proper stochastic barrier certificate, which guarantees probability of safety $P_\mathrm{s}(X_\mathrm{s}, X_0, N) \geq 1 - (\eta + \beta N)$. □

Algorithm 1 CEGS for PWC-SBF

Input: Initial set X_0, partition $X = \{X_i\}_{i=1}^K$, time horizon N, and feasible transition kernel sets $\{\mathcal{P}_i\}_{i=1}^K$.
Output: Optimal PWC-SBF B^* and safety probability bound $\underline{P}_{\mathrm{s}}^*$.

```
$\{\tilde{\mathcal{P}}_i \leftarrow$ sampleDist$(\mathcal{P}_i)\}_{i=1}^K$      ▷ Initialize $\tilde{\mathcal{P}}_i$
$\beta^* \leftarrow 1$                                  ▷ Initialize $\beta^*$
$\{\beta_i \leftarrow 0\}_{i=1}^K$                          ▷ Initialize $\beta_i$
while $\beta^* > \max_i \beta_i$ do
    $b^*, \eta^*, \beta^* \leftarrow$ pwbSynthLP$(X_0, N, X, \{\tilde{\mathcal{P}}_i\}_{i=1}^K)$
    for $i \leftarrow 1$ to $K$ do
        $p_i, \beta_i \leftarrow$ counterExDistLP$(b^*, \mathcal{P}_i, i)$
        $\tilde{\mathcal{P}}_i \leftarrow \tilde{\mathcal{P}}_i \cup \{p_i\}$        ▷ add counterexamples
return $B^* \leftarrow (b^*, 1)$ and $\underline{P}_{\mathrm{s}}^* \leftarrow 1 - (\eta^* + N\beta^*)$
```

Computational Complexity The time complexity of a standard primal linear program is $\mathcal{O}(n^2 m)$ where n is the number of decision variables and m is the number of constraints [22]. The program in Theorem 3 has $n = 2K^2 + 2K$ decision variables and $m = K^2 + 6K + L + 1$ where $L = |\{i : X_i \cap X_0 \neq \emptyset\}|$ is the number of regions intersecting with the initial set. Combining the complexity of a linear program with the number of decision variables and constraints for our dual linear program, we get $\mathcal{O}(n^2 m) = \mathcal{O}(K^6)$. This may be prohibitive, hence we introduce a computationally more efficient method.

5.2.2 Counter-Example Guided Synthesis

Next, we introduce another PWC-SBF synthesis method that is exact and computationally more efficient than the dual LP approach. The method is based on splitting the minimax problem in Theorem 2 into two separate LPs. One LP generates a candidate PWC-SBF that optimizes the unsafety probability given a set of finite feasible distributions $\tilde{\mathcal{P}}_i \subset \mathcal{P}_i$, and the other LP generates distribution witnesses (counterexamples) that violate the safety probability guarantee of the candidate PWC-SBF. Then, the witnesses are added to $\tilde{\mathcal{P}}_i$, and the process repeats until no more counterexample can be generated. We dub this method as Counter-Example Guided Synthesis (CEGS) for PWC-SBF.

The CEGS algorithm is shown in Alg. 1, which relies on subroutines in Algs. 2 and 3. The main algorithm first sets up the counterexample distribution sets $\tilde{\mathcal{P}}_i$ by sampling a feasible distribution from each $\mathcal{P}_i$. Then, based on $\tilde{\mathcal{P}}_i$s, it synthesizes a candidate optimal vector b^* that minimizes the unsafety probability with its corresponding scalar β^* using the subroutine pwbSynthLP. As shown in Alg. 2, this subroutine is an LP that captures the min component (outer optimization) of the minimax problem in Theorem 1. Next, given b^*, for each partition region X_i, an optimal distribution p_i that maximizes its corresponding martingale gap β_i is computed using subroutine counterExDistLP and then added to the set of witnesses $\tilde{\mathcal{P}}_i$. As shown in Alg. 3, counterExDistLP is also an LP, and it captures the max component (inner optimization) of the minimax problem. If the obtained martingale gap β_i is greater than β^* for some region X_i, it means that the candidate b^* is not optimal and there exists at least a distribution that violates the probabilistic guarantee of the candidate. Hence, the process repeats with the updated witnesses until $\beta_i \leq \beta^*$ for all $i \in \{1, \ldots, K\}$.

Algorithm 2 pwbSynthLP$(X_0, N, X, \{\tilde{P}_i\}_{i=1}^K)$

Input: Initial set X_0, partition $X = \{X_i\}_{i=1}^K$, time horizon N, and finite set of distributions $\{\tilde{\mathcal{P}}_i\}_{i=1}^K$.
Output: Optimal vector $b^* = (b_1, \ldots, b_K)^*$ and scalars η^* and β^* that minimize unsafety probability w.r.t. the given finite distribution sets $\tilde{P}_1, \ldots, \tilde{P}_K$.

```
$b^*, \beta^*, \eta^* \leftarrow \arg\min_{b,\beta,\eta} \eta + N \cdot \beta$

  subject to:
      $0 \leq b_i \leq 1$                              $\forall i \in \{1, \ldots, K\}$
      $b_i \leq \eta$                                  $\forall i : X_i \cap X_0 \neq \emptyset$
      $\sum_{j=1}^{K} b_j \cdot p_{ij} + p_{i\mathrm{u}} \leq b_i + \beta$        $\forall i$ and $\forall p_i \in \tilde{\mathcal{P}}_i$

return $b^*, \eta^*, \beta^*$
```

Algorithm 3 counterExDistLP$(b, \mathcal{P}_i, i)$

Input: Vector of constants $b = (b_1, \ldots, b_k)$, a set of feasible distributions $\mathcal{P}_i$ as defined in Equation (13), and region index i.
Output: Counterexample distribution $p_i = (p_{i1}, \ldots, p_{iK}, p_{i\mathrm{u}})$ that maximizes martingale gap β with respect to b.

```
$p_i, \beta \leftarrow \arg\max_{p_i,\beta} \beta$

  subject to:
      $\sum_{j=1}^{K} b_j \cdot p_{ij} + p_{i\mathrm{u}} = b_i + \beta$
      $\underline{p}_{ij} \leq p_{ij} \leq \overline{p}_{ij}$              $\forall j \in \{1, \ldots, K, \mathrm{u}\}$
      $\sum_{j=1}^{K} p_{ij} + p_{i\mathrm{u}} = 1$

return $p_i, \beta$
```

The following theorem guarantees that the CEGS algorithm terminates in finite time with an optimal solution.

Theorem 4 (CEGS for PWC-SBF) *Algorithm 1 terminates in finite time with PWC-SBF B^* that is an optimal solution to the problem in Theorem 2.*

Proof.

Denote β^* and β_{opt} as the candidate and true optimal β, respectively. From Theorem 3 it follows that β_{opt} exists and is unique. At each iteration $l \in \mathbb{N}_{\geq 1}$ 0f CEGS, the Alg. 1 synthesizes candidate optimal vector $b^\star$ along with associated β^* through subroutine Alg. 2. If at iteration $l : \beta^* \neq \beta_{\text{opt}}$, $\exists \beta_i$ such that $\beta^* > \max_i \beta_i$. This implies that there exists a set of witnesses p_i that further maximize the martingale gap, added at iteration $l+1$. These counterexamples are found through the optimization of Alg. 3, which is more constrained than subroutine Alg. 2. Assume by that no witnesses p_i are added at iteration $l+1$, and that $\beta^* \neq \beta_{\text{opt}}$. That would contradict the uniqueness of β_{opt}. The algorithm converges when $\beta^* = \beta_{\text{opt}} \leq \max_i \beta_i$. Given the finite vertices of the simplex $\mathcal{P}_i$, for all $i \in \{1, ..., K\}$, β_{opt} is converged to in finite iterations.

□

Computational Complexity We note that the LP in subroutine counterExDist (Agl. 3) can be solved more efficiently using a procedure called O-maximization as proposed in [23]. The time complexity of this procedure is $\mathcal{O}(K \log K)$, whereas a standard LP algorithm runs in $\mathcal{O}(K^3)$ time. In the worst-case, all the vertices of the simplex $\mathcal{P}_i$ are explored for all $i \in \{1, ..., K\}$, which has time complexiy $\mathcal{O}(K^2)$. It is further noted that the Alg. 1 keeps track of the history of counterexamples p_is added at each iteration, making the approach potentially memory-intensive.

5.2.3 Projected Gradient Descent

While CEGS is faster than the dual approach, it can be memory intensive. To alleviate this and allow better scalability, we present a third method for computing a PWC-SBF, namely, a gradient descent-based approach.

With an abuse of notation, let $\beta_i(b)$ be martingale gap of region X_i for a given PWC-SBF defined by vector $b = (b_1, \ldots, b_K)$. Specifically,

$$\beta_i(b) = \sup_{p_i \in \mathcal{P}_i} \max \Big\{0, \sum_{j=1}^{K} b_j \cdot p_{ij} + p_{i\text{u}} - b_i\Big\}, \qquad (21)$$

and $\beta(b) = \max_i \beta_i(b)$. Similarly, we denote

$$\eta(b) = \max_{i : X_i \cap X_0 \neq \emptyset} b_i. \qquad (22)$$

Then, we define loss (objective) function

$$\mathcal{L}(b) = \eta(b) + N \cdot \beta(b). \qquad (23)$$

This loss function indeed describes the objective function of the minimax problem in Theorem 2. Hence, by minimizing $\mathcal{L}(b)$, we solve the minimax problem. Below, we show that $\mathcal{L}(b)$ is convex, and hence, we can use a gradient descent-based method for its optimization. More precisely, our approach is projected subgradient descent since the elements of b are constrained to $[0, 1]$ and the maximization and supremum in Equation (21) can be discontinuous, admitting only subgradients.

Theorem 5 (Convexity of $\mathcal{L}(b)$) *The objective function $\mathcal{L}(b)$ in Equation* (23) *is convex in b.*

Proof. The proof follows a standard structure from disciplined convex programming, where functions are composed under convexity-preserving operations. We start by proving that $\beta_i(b)$ is convex in b. To this end, observe that $\sum_{j=1}^{K} b_j \cdot p_{ij} + p_{i\text{u}} - b_j$ is convex in b, invariant to the value p_i. Next, $\max(0, \cdot)$ is a convexity-preserving function, thus $\beta_i(b)$ is convex in b. The finite maximization for both η and β are once again convexity-preserving operations and finally, addition is convexity-preserving. This concludes the proof that $\mathcal{L}(b)$ is convex in b. □

Notice that the computation of $\beta_i(b)$ in Equation (21) is equivalent to the inner optimization problem of CEGS. Hence, the counterExDistLP routine in Alg. 3 (based on O-maximization method) can be used for efficient computation of subgradients of β_i. One subgradient for $\mathcal{L}(b)$ is the following

$$\nabla_b \mathcal{L}(b) = \vec{1}_\eta(b) + N \cdot \vec{1}_\beta(b), \qquad (24)$$

where $\vec{1}_\eta(b) = \vec{1}_i$ is a one-hot vector with the 1 being in element $i = \text{argmax}_{i:X_i \cap X_0 \neq \emptyset} b_i$, and

$$\vec{1}_\beta(b) = \nabla_b \beta_l(b) = p_l^* - \vec{1}_l, \qquad (25)$$

where $l = \arg\max_l \beta_l(b)$, and $p_l^* = \arg\max_{p_l \in \mathcal{P}_l} \sum_{j=1}^{K} b_j \cdot p_{lj} + p_{l\text{u}} - b_l$.

A challenge with applying subgradient descent in practice is that small step sizes are required due to the non-smoothness, slowing down the convergence. We propose to ameliorate this by substituting the maximization in $\mathcal{L}(b)$ with an L_p-norm where $1 < p < \infty$. Specifically, let $\tilde{\eta}(b) = (b_{i_1}, \ldots, b_{i_m})$ be a vector of PWC-SBF values corresponding to the regions that overlap with X_0, and $\tilde{\beta}(b) = (\beta_1(b), \ldots, \beta_K(b))$ be a vector of $\beta(b)_i$ values. Then, the proposed loss function is

$$\tilde{\mathcal{L}}(b) = \|\tilde{\eta}(b)\|_p + N\|\tilde{\beta}(b)\|_p, \qquad (26)$$

Algorithm 4 GD for PWC-SBF

Input: Initial set X_0, partition $X = \{X_i\}_{i=1}^K$, time horizon N, and feasible transition kernel sets $\{\mathcal{P}_i\}_{i=1}^K$.
Output: Optimal PWC-SBF B^* and safety probability bound $\underline{P}_\mathrm{s}^*$

```
$b^{(1)} \leftarrow (p_{1\mathrm{u}}, \dots, p_{K\mathrm{u}})$                  ▷ Initialize $b$
$b_\mathrm{best} \leftarrow \mathrm{COPY}(b)$                         ▷ Initialize $b_\mathrm{best}$
$\mathcal{L}_\mathrm{best} \leftarrow \infty$
$k \leftarrow 1$                                        ▷ Iteration number
while not converged do
    $g \leftarrow \nabla_{b^{(k-1)}} \tilde{\mathcal{L}}(b^{(k-1)})$            ▷ According to (27)
    $\alpha_k \leftarrow \textsc{StepSize}(k, g)$
    $b^{(k)} \leftarrow b^{(k-1)} - \alpha_k g$             ▷ Gradient descent step
    for $i \leftarrow 1$ to $K$ do                      ▷ Project onto $[0, 1]$
        $b_i^{(k)} \leftarrow \max(0, \min(1, b_i^{(k)}))$
    $\eta_\mathrm{best}, \beta_\mathrm{best} \leftarrow \|\tilde{\eta}(b^{(k)})\|_\infty, \|\tilde{\beta}(b^{(k)})\|_\infty$
    if $\eta_\mathrm{best} + N\beta_\mathrm{best} < \mathcal{L}_\mathrm{best}$ then
        $b_\mathrm{best} \leftarrow \mathrm{COPY}(b^{(k)})$
        $\mathcal{L}_\mathrm{best} \leftarrow \eta_\mathrm{best} + N\beta_\mathrm{best}$
    $k \leftarrow k + 1$
return $B^* \leftarrow (b_\mathrm{best}, 1)$ and $\underline{P}_\mathrm{s}^* \leftarrow 1 - \mathcal{L}_\mathrm{best}$
```

and the corresponding gradient is

$$\begin{aligned}\nabla_b \tilde{\mathcal{L}}(b) = \sum_{k=1}^{m} \left(\frac{\tilde{\eta}(b)_k}{\|\tilde{\eta}(b)\|_p} \right)^{p-1} \vec{1}_{i_k} + \\ \sum_{i=1}^{K} \left(\frac{\tilde{\beta}_i(b)}{\|\tilde{\beta}(b)\|_p} \right)^{p-1} \nabla_b \beta_i(b)\end{aligned} \tag{27}$$

The benefit of this loss is that L_p-norms are smooth, and that $\tilde{\mathcal{L}}(b) \geq \mathcal{L}(b)$ for every $p_i \in \mathcal{P}_i$ and all b. Furthermore, the relative magnitude is bounded, $\|y\|_p \,/\, \|y\|_\infty \leq \sqrt[p]{r}$ for any $y \in \mathbb{R}^r$. With the L_p-norm, we may tune the over-approximation to a trade-off between smoothness and tighter approximating the L_∞-norm. The modified loss, through smoothness, also addresses the issue that each $\mathcal{P}_i$ often is sparse, i.e., $\overline{p}_{ij} \approx 0$ for many j. Therefore, $\nabla_b \mathcal{L}(b)$ is often 0 for most regions. We remark that the modified loss $\tilde{\mathcal{L}}(b)$ is not smooth due to the maximization and supremum in the computation of $\beta_i(b)$.

The projected subgradient descent procedure is described in Alg. 4. For each iteration, the algorithm proceeds by calculating the gradient according to Eq. (27) and an appropriate, decreasing step size α followed by the gradient step. Since the PWC-SBF must reside in $[0, 1]^K$ and the gradient step may violate this, it is projected back onto the admissible space. Finally, since subgradient methods are not a steepest descent method, the algorithm keeps track of the best observed loss.

To guarantee convergence of gradient descent (Alg. 4) towards the optimum, it is paramount that the objective is convex. Theorem 5 shows that $\mathcal{L}(b)$ is convex. From this, it immediately follows that $\tilde{\mathcal{L}}(b)$ is also convex.

Corollary 2 *The modified objective $\tilde{\mathcal{L}}(b)$ is convex in b.*

The proof follows from the fact that η and β are convex in b. Further, L_p-norms are convexity-preserving operations [22].

Convergence and Computational Complexity We may prove that Alg. 4 finds an ϵ-optimal b within a finite number of steps. Specifically, we rely on a result for convergence with strictly decreasing and non-summable step sizes [24]. To this end, let ϵ denote the desired error, the gradient bound $G \geq \sup_{b \in [0,1]^K} \|\nabla_b \tilde{\mathcal{L}}(b)\|_2$, and the upper bound of the initial distance to the optimal barrier b^* by $R \geq \|b^{(1)} - b^*\|_2$. Then there exists an integer N_1, N_2 such that $\alpha_k \leq \epsilon/G$ for all $k > N_1$ and

$$\sum_{k=1}^{N_2} \alpha_k \geq \frac{1}{\epsilon} \left(R^2 + G^2 \sum_{k=1}^{N_1} \alpha_k^2 \right).$$

Then for any $k > N = \max(N_1, N_2)$ it holds that

$$b_\mathrm{best} - b^* \leq \epsilon. \tag{28}$$

A conservative bound on R is K and on G is $m + NK\sqrt{2}$. To conclude, Alg. 4 converges to a desired precision in a finite number of steps [24].

The complexity per iteration of the projected subgradient descent (Alg. 4), utilizing the O-maximization procedure to compute β_i, is $\mathcal{O}(K \log K)$. We emphasize that much of the computation can be parallelized.

6 Evaluations

To show the power of PWC-SBFs and efficacy of our proposed synthesis methods, we study their performance on a set of seven benchmark problems, consisting of stochastic systems with linear and nonlinear dynamics and varying dimensionality, from 2D to 8D. We also compare the performance of PWC-SBFs against state-of-the-art continuous SBF techniques, namely SOS and NBF.

Our implementations of all methods, except NBF, including the three proposed PWC-SBF synthesis methods, are in `Julia`. The code is publically available for download on GitHub [1]. All the computations for the benchmarks were performed on a computer with 3.9 GHz 8-Core CPU and 128 GB of memory.

The stochastic systems we considered are:

[1] `https://github.com/aria-systems-group/StochasticBarrier.jl`.

Table 1: Benchmark results for the *piecewise constant* and *continuous* SBF methods. The three PWC-SBF synthesis methods are: dual LP, CEGS and GD. These are compared against state-of-the-art methods: SOS and NBF. K denotes the size of partition of X_s, τ_p is the computation time for the bounds on the transition kernel, $\underline{P}_\mathrm{s}$ is the obtained lower bound on the safety probability for $N = 10$ time steps, τ_o is the computation time for the SBF synthesis, and Deg is the degree of the SOS polynomial for SBF.

			Piecewise Constant Stochastic Barriers						Continuous Stochastic Barriers*				
			Dual LP		CEGS		GD		NBF		SOS		
Model	K	$\tau_p(s)$	$\underline{P}_\mathrm{s}$	$\tau_0(s)$	$\underline{P}_\mathrm{s}$	$\tau_o(s)$	$\underline{P}_\mathrm{s}$	$\tau_o(s)$	$\underline{P}_\mathrm{s}$	$\tau_o(s)$	Deg	$\underline{P}_\mathrm{s}$	$\tau_o(s)$
Linear	64	0.02	0.992	0.52	0.992	0.04	0.952	0.04	0.585	3850.93	4	0.582	0.014
2D	225	0.31	0.998	164.60	0.998	0.44	0.973	0.20	0.940	3991.47	8	0.582	0.265
Convex	900	8.85	0.999	1087.78	0.999	17.93	0.990	7.22	0.961	4025.67	30	0.978	151.16
	2500	41.44	0.999	2897.77	0.999	88.45	0.998	52.78	0.976	4085.65	36	0.992	458.21
Linear	900	5.04	0.494	1197.99	0.494	3.79	0.494	3.52	0.792	3546.69	12	0.010	0.02
2D	1225	8.20	0.800	1389.78	0.800	7.22	0.800	6.64	0.844	3579.58	20	0.010	11.08
Non-Convex	1444	9.18	0.921	1545.45	0.921	11.65	0.921	10.12	0.855	3589.13	24	0.023	37.89
	2926	47.98	0.927	3161.56	0.927	98.36	0.927	20.86	0.928	3599.85	26	0.034	62.88
	5890	179.94	0.929	8191.65	0.929	458.44	0.929	133.84	0.929	3675.77	30	0.075	196.85
	11818	477.45	-	TO	0.936	1875.49	0.936	842.67	0.931	3744.23	34	-	OM
	24336	987.65	-	TO	0.938	4441.55	0.938	3099.30	0.936	4234.56	36	-	OM
Pendulum	120	6.37	1.00	0.51	0.99	5.84	0.989	3.75	0.999	4242.89	4	0.999	7.71
2D	240	18.33	1.00	6.08	1.00	14.88	0.990	9.99	0.999	4457.82	4	0.999	34.96
	480	37.84	1.00	29.39	1.00	43.42	0.999	17.88	0.999	4675.12	4	0.999	187.60
Unicycle	1250	1103.42	0.750	1000.19	0.750	26.37	0.750	5.68			2	0.00	3110.21
4D	1800	1756.25	0.975	1719.58	0.975	92.26	0.975	25.78			4	0.00	5451.19
	2400	2001.11	0.998	2548.56	0.998	145.45	0.998	55.59			6	-	OM
Quadrotor	7865	80.80	0.770	9906.67	0.770	1174.49	0.770	2589.56			2	0.584	0.20
6D	15625	160.61	-	TO	0.901	3788.98	0.901	3258.87			8	0.900	8628.58
Convex	46656	458.59	-	TO	-	TO	0.912	9542.75			12	-	OM
Quadrotor	15625	188.10	-	TO	0.670	3845.25	0.670	3478.31			4	0.00	4.91
6D	31250	395.59	-	TO	0.810	9548.78	0.810	5878.28			8	0.00	8715.54
Non-Convex	46656	506.99	-	TO	-	TO	0.900	9789.54			12	-	OM
Quadrotor	65536	845.44	-	TO	-	TO	0.500	19377.90			2	0.00	14830.23
8D	128000	2530.74	-	TO	-	TO	0.560	39132.59			4	-	OM

* For linear systems that use SOS optimization, the state space is not partitioned. Further, synthesis methods for SOS and NBF do not use bounds on the transition kernel.

- 2D linear stochastic system with a (1) convex X_s, and (2) non-convex X_s,
- 2D pendulum Neural Network Dynamic Model (NNDM) adapted from [5],
- 4D nonlinear unicycle model under a hybrid feedback control law,
- 6D linearized lateral and vertical quadrotor guidance model with a hybrid controller with a (1) convex X_s, and (2) non-convex X_s,
- 8D linearized lateral and longitudinal quadrotor guidance model with a hybrid controller.

A summary of the results is shown in Table 1. More details on the results are presented in Table A.1 in Appendix A. For all case studies, $N = 10$ time steps. `TO` and `OM` denote time-out ($\tau > 45 \times 10^3 s$) and out-of-memory, respectively. Due to limitations pertaining to memory or hybrid control, the NBF is only run for the 2D linear system and the pendulum NNDM. It is further denoted that all NBF experiments are run for 150 epochs, with 400 iterations per epoch. Each NBF architecture consists of 2 hidden layers with 32 ReLu activated neurons per layer. For a relevant comparison, it is noted that NBF uses the same initial partitioning as PWC. Finally, it is noted that the GD approach uses a decay rate of 0.999, and a termination condition when the difference in $\max \beta$ is less than $1e^{-6}$ between 50 iterations.

We provide detailed discussions on the experimental setup and obtained results below. For case studies with non-convex X_s, we use the notion of *obstacle* to refer to the unsafe sets. All the obstacles are hyper-rectangles defined by a center point $c = (c_1, \dots, c_n) \in \mathbb{R}^n$ and the half-length along each dimension denoted by ϵ_i. Formally, obstacle

$$\mathbb{O}(c, \epsilon) = \big\{(x_1, ..., x_n) \in \mathbb{R}^n : |x_i - c_i| \leq \epsilon_i \ \ \forall 1 \leq i \leq n\big\}.$$

6.1 2D Linear System

We considered a linear stochastic system in $\mathbb{R}^2$ with dynamics

$$\mathbf{x}(k+1) = 0.5I \cdot \mathbf{x}(k) + \mathbf{v},$$

where I is the identity matrix, and $\mathbf{v} \sim \mathcal{N}(0, 1e^{-2}I_{2\times 2})$. The initial set $X_0 = [-0.8, -0.6] \times [-0.2, 0.0]$. We consider two cases for X_s, both defined based on the set $X_D = [-1, 0.5] \times [-0.5, 0.5]$ as shown in Fig. 1a:

(1) Convex safe set: $X_\text{s}^{conv} = X_D$, and
(2) Non-convex safe set: $X_\text{s}^{ncov} = X_D \setminus \bigcup_{i=1}^{2} \mathbb{O}_i(c_i, \epsilon_i)$, where

$$\begin{aligned} c_1 &= (-0.55, 0.30), \quad & \epsilon_1 &= (0.02, 0.02), \\ c_2 &= (+0.55, 0.15), \quad & \epsilon_2 &= (0.02, 0.02). \end{aligned}$$

Convex X_s Case Observe that all three PWC-SBF methods outperform the state-of-the-art continuous SBF techniques SOS and NBF. It takes a degree 36 polynomial SBF to obtain a result similar to that of the PWC methods using just $K = 64$, despite the absence of obstacles (convexity of X_s). This is mainly attributed to the fact that initial set is not configured in a symmetric fashion, making curve fitting difficult for even simple two-dimensional problems.

The main limitation of NBF in this experiment pertains to the training time. Note that it takes the continuous SBF methods 3-4 orders of magnitude larger in computation time to arrive to a similar value of $\underline{P}_\text{s}$ as the PWC methods. In addition, the continuous SBF methods never reached $\underline{P}_\text{s} = 0.999$.

Among the PWC-SBF methods, the Dual LP and CEGS always obtain the same $\underline{P}_\text{s}$ as they are both optimal and exact methods. However, it respectively takes CEGS at least an order of magnitude less time to arrive to this optimal value. That is due to the large number of the decision variables in the Dual LP, which is persist further for as K increases. The GD approach is the fastest method, however, its hyperparameters are tricky to design. For this case study, GD terminated before convergence to the true optimal $\underline{P}_\text{s}$.

Non-convex X_s Case For this case, the power of the PWC-SBFs are even more evident (see Fig. 1). Due to the fact that the obstacles are within the safe set, the SOS approach has difficulties fitting a function. This forces $B(x) \geq 1$ in the entire state space, as is evident from Fig. 1b. This phenomenon does not improve much for a significant increase in the degree of the SOS polynomial, as it is an inherent limitation of the appraoch. On the other hand, our methods effectively tackle this problem by assigning $b_i = 1$ to the regions that overlap with obstacles, and optimization over the remainder of the state space. This results in the PWC as depicted in Fig. 1c. The difference is likewise highly notable in terms of the safety probability $\underline{P}_\text{s}$, where SOS can only guarantee safety probability of 0.075 for the polynomial degree 30, where two of PWC-SBF methods provide up to 0.938 safety probability.

Note that for large values of K, the dual LP is unable to terminate, due to the scope of the required convex optimization. A similar fact is observed for high degree SOS barrier polynomials. The NBF method performs well in terms of safety probability, at lower number of partitions even better than PWC SBF. Nonetheless, the PWC SBFs are superior in terms of computation time.

6.2 2D Pendulum NNDMs

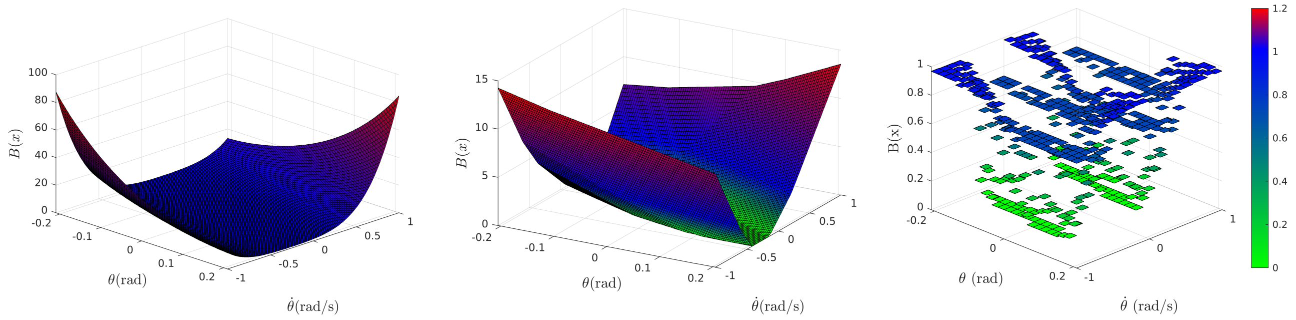
In this case-study, we consider the NNDM from [5] with dynamics $\mathbf{x} = f^{NN}(x) + \mathbf{v}$, where f^{NN} is a NN trained on a Pendulum agent with a fixed mass m and length l in close-loop with a NN controller. The actuator is limited by $u \in [-1, 1]$ and the dynamics of the pendulum are

$$\begin{aligned} \dot{\theta}_{k+1} &= \dot{\theta}_k + \frac{3g}{2l}\sin(\theta_k)\delta t^2 + \frac{3}{ml^2}u\delta t^2, \\ \theta_{k+1} &= \theta_k + \dot{\theta}_{k+1}\delta t \end{aligned}$$

For the details on the trained model we refer to [5]. For this system, noise $\mathbf{v} \sim \mathcal{N}(0, 1e^{-4}I)$. The safe set is $\theta \in [-\frac{\pi}{15}, \frac{\pi}{15}]$ and $\dot{\theta} \in [-1, 1]$ and the initial set is $\theta, \dot{\theta} \in [-0.01, 0.01]$.

We note that since for this particular case study the dynamics are given by a NN, the SOS approach also uses the partitioning of size K, where local linear relaxations of f^{NN} are performed. For more information on this SOS formulation of the SBF see [5].

As can be observed from Table 1, the PWC, SOS and NBF methods perform similarly in terms of safety probability. It is noted that SOS performs well due to the fact that the initial set is centered in the state space. In Fig. 2, we show the plots of the SOS, NBF and PWC (dual approach) SBFs. From Fig. 2a, we observe that the obtained SOS polynomial is an over-fit, as the values near the boundary of the safe set by-far exceed one. We observe a similar yet less excessive pattern for the NBF in Fig. 2b. For the PWC SBF the barrier in the domain is less or equal to 1. Among the PWC SBF methods, a general trend is observed: the dual LP is the slowest algorithm and the GD the fastest. It is also worth noting that that the GD method terminates with a $\underline{P}_\text{s}$ that is slightly below the optimal value, indicating the difficulty of designing the termination criterion.



(a) Degree-8 SOS-polynomial (b) NBF with $|K| = 480$. (c) PWC (dual LP) with $|K| = 480$.

Fig. 2. SBF plots for the pendulum NNDM case study.

6.3 4D Unicycle

In this case study, we consider a wheeled mobile robot with the dynamics of a unicycle

$$\dot{x} = v\cos\theta, \quad \dot{y} = v\sin\theta, \quad \dot{\theta} = \omega, \quad \dot{v} = a,$$

where $x \in [-1.0, 0.5]$ and $y \in [-0.5, 1.0]$ are the Cartesian position, $\theta \in [-1.75, 0.5]$ is the orientation with respect to the $x-$axis, and $v \in [-0.5, 1.0]$ is the speed. The inputs are steering rate ω and acceleration a. We design a feedback linearization controller according to [25] coupled with an LQR stabilizing controller, making this a hybrid model. We use $\Delta t = 0.01$ to obtain a discrete time dynamics, using the Euler method. We add noise $\mathbf{v}\ \mathcal{N}(0, 1e^{-4}I_{4\times 4})$ to capture the time-discretization error that is inherent to the Euler method.

Fig. 3 shows 10^5 Monte Carlo simulations of the trajectories of the unicycle from the initial set defined by a ball centered at $(-0.5, -0.5, 0, 0)$ with radius 0.01. These simulated trajectories suggest that the system is stable under the hybrid control law. This is in-line with the results in Table 1 for the PWC methods, especially for $K = 2400$, for which $\underline{P}_\mathrm{s} = 0.998$. Note that the SOS approach highly suffers for this nonlinear case study, with $\underline{P}_\mathrm{s} = 0$, despite very long computation times. This further emphasizes the efficacy of our methods. Observe that the GD approach terminates with the optimal $\underline{P}_\mathrm{s}$ value in this case study. We finally note that since the controller is hybrid, current NBF formulations cannot be employed for this problem.

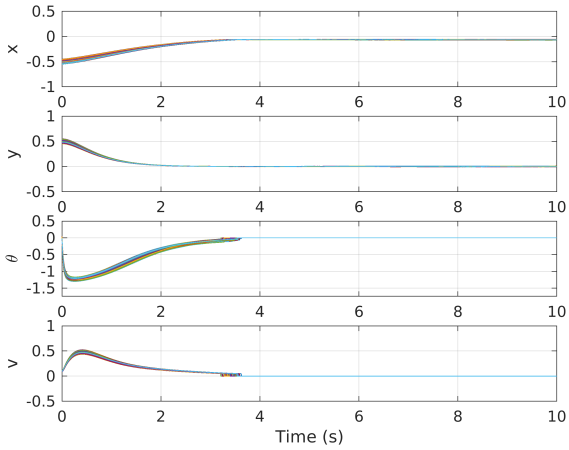


Fig. 3. Monte Carlo simulation (10^5 instances) for the unicycle model with $t = 10s$.

6.4 6D and 8D Quadcopter Systems

To evaluate scalability of the proposed PWC-SBF methods, we considered a set of high-dimensional systems. Specifically, we considered a quadrotor guidance model where the dynamics consist of lateral and vertical dynamics and the longitudinal and spin variables are constrained, and then lateral and longitudinal motion model by fixing the vertical and spin variables.

6.4.1 6D Lateral-Vertical Model

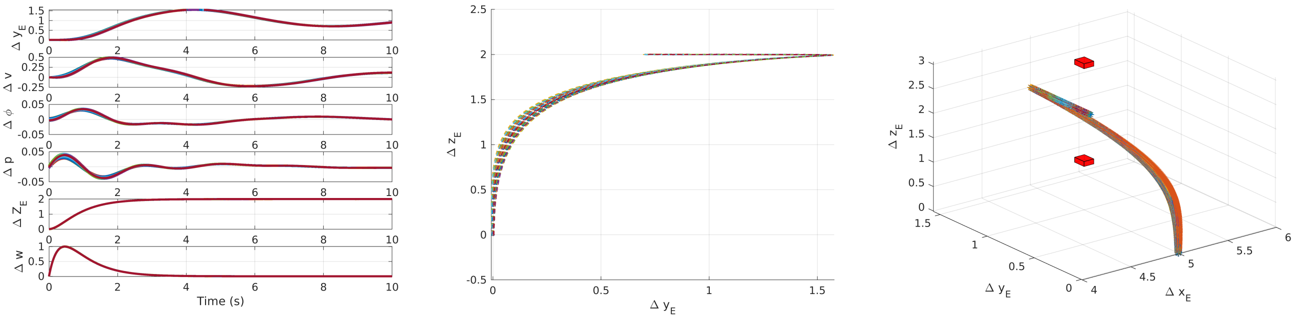
The lateral closed-loop state space model is defined as

$$\Delta\dot{y}_E = \Delta v, \qquad \Delta\dot{v} = g\Delta\phi,$$
$$\Delta\dot{\phi} = \Delta p, \qquad \Delta\dot{p} = \frac{1}{I_x}\Delta L_c,$$

where g is gravity, and I_x is the inertia about the $x-$axis. State $y_E \in [-0.5, 2.0]$ is the $y-$position, and $v \in [-1.0, 1.0]$ is the corresponding velocity. The roll angle $\phi \in [-0.1, 0.1]$, the roll rate $p \in [-0.1, 0.1]$, and L_c is the roll control moment. The control law follows directly from

$$\Delta L_c = -k_1\Delta p - k_2\Delta\phi - k_3\Delta v - k_3k_4\Delta y_E + k_3k_4\Delta y_{E,r},$$

where k_i denotes the gain for $i \in \{1, \dots, 4\}$, and $\Delta y_{E,r}$ is the reference position.



(a) State Evolution

(b) y-z Plane

(c) 3D Workspace

Fig. 4. Monte Carlo simulations (10^5 instances) for the 6D quadcopter model with lateral-vertical dynamics for $10s$. Trajectories are visualized in (a) state vs. time plot, (b) y-z Plane, and (c) 3D workspace. The red boxes in (c) are the obstacles. The control laws stabilize the quadrotor around the equilibrium point $x_{\text{eq}} = (1, 0, 0, 0, 2, 0)$.

The lateral model is appended with a guidance model for vertical motion given by

$$\Delta \dot{z}_E = \Delta w, \qquad \Delta \dot{w} = \frac{1}{m}\Delta Z_c,$$

where m denotes the quadrotor mass, $z_E \in [-0.5, 3.0]$ denotes the z-position and $w \in [-0.5, 1.5]$ is the yaw rate. The control law ΔZ_c is defined as

$$\Delta Z_c = -k_1 \Delta z_E - k_2 \Delta w - Fr$$

where Fr denotes the reference signal for the vertical dynamics. The full dynamics constitutes a 6D linear model. Using $\Delta t = 0.01$, we obtained a discrete-time dynamics via the Euler method, to which we added noise $\mathbf{v} \sim \mathcal{N}(0, 1e^{-4} I_{6\times 6})$.

The designed control laws stabilize the quadrotor around the equilibrium point $x_{\text{eq}} = (1, 0, 0, 0, 2, 0)$. Monte Carlo simulations of 10^5 trajectories of this model is shown in Fig. 4. The 3D workspace (environment) is shown in Fig. 4c.

We considered two safe sets:

(1) Convex safe set: X_{s}^{conv} is defined by the given ranges of of the variables above,
(2) Non-convex safe set: $X_{\text{s}}^{nconv} = X_{\text{s}}^{conv} \setminus \bigcup_{i=1}^{2} \mathbb{O}(c_i, \epsilon_i)$, where

$$\begin{aligned} c_1 &= (1, 0, 0, 0, 1, 0), & \epsilon_1 &= 0.01 \cdot \mathbb{1}_{6\times 1} \\ c_2 &= (1, 0, 0, 0, 2.75, 0), & \epsilon_2 &= 0.01 \cdot \mathbb{1}_{6\times 1}. \end{aligned}$$

Fig. 4c depicts the obstacles.

Convex X_{s}^{conv} case Table 1 shows that the two PWC methods (namely, CEGS and GD) outperform SOS, both in terms of computation time, as well as safety probability certification. It is important to note that, systems with linear dynamics and convex X_{s} are ideal setup for SOS optimization; however, it has a lot of difficulty with the dimensionality of the system as the results suggest. For this case study, an SOS polynomial of degree greater than 8 runs out of memory. Finally, we did not run NBF on this problem since it also suffers from scalability.

Non-convex X_{s} case In this case, it is even more evident that PWC-SBFs can be more powerfully than continuous SBFs. The SOS method provides $\underline{P}_{\text{s}} = 0.00$, whereas the GD method gets to $\underline{P}_{\text{s}} = 0.900$. Observe that the dual LP method times out, which is also true for the CEGS method when $K > 4.5 \times 10^4$. The GD approach is able to handle this, the capability of PWC barriers for scalability purposes.

6.5 8D Lateral-Longitudinal Dynamics

Here, we aim to test the performance boundary of PWC-SBFs in terms of scalability. We consider an 8D model by combining the lateral dynamics of the quadcopter with its longitudinal and fixing vertical and spin variables. In a similar fashion as lateral, the longitudinal guidance is constructed under the dynamics

$$\begin{aligned} \Delta \dot{x}_E &= \Delta u, & \Delta \dot{u} &= -g\Delta\delta, \\ \Delta \dot{\theta} &= \Delta q & \Delta \dot{q} &= \frac{1}{I_y}\Delta M_c, \end{aligned}$$

where g is gravity, and I_y is the inertia about the $y-$axis. State $x_E \in [-0.5, 4.0]$ is the $x-$position, and $u \in [-0.5, 1.5]$ is the corresponding velocity. The pitch angle $\theta \in [-0.1, 0.1]$, the pitch rate $q \in [-0.1, 0.1]$, and M_c is the pitch control moment. The control law follows directly from

$$\Delta M_c = -l_1 \Delta q - l_2 \Delta\theta - l_3 \Delta u - \\ l_3 l_4 \Delta x_E + l_3 l_4 \Delta x_{E,r},$$

where l_i denotes the gain for $i \in \{1, \dots, 4\}$, and $\Delta x_{E,r}$ is the reference position.

The results are shown in the bottom 2 rows of Table 1. For this system, the SOS approach is unable to provide any safety guarantees, i.e., $\underline{P}_\mathrm{s} = 0$. The dimensionality naturally has a major effect on the performance of the SOS optimizer. The dual LP and CEGS likewise suffer, pertaining to the dimensionality and the partitioning of the state space. Nonetheless, the GD approach is still able to handle the large scale optimization, and can provide a lower safety probability of $\underline{P}_\mathrm{s} = 0.560$. While this is not near the statistical safety threshold obtained through simulation ($\tilde{P}_{\mathrm{s},sim} \approx 0.95$), it is clearly show that the proposed PWC-SBF method outperforms the current state-of-the-art SBF methods.

7 Conclusion

In this work, we introduce a formulation for piecewise (PW) stochastic barrier functions (SBFs). We discuss the flexibility they provide at the cost of increased computational cost. To lay the groundwork, we focus on their simplest form, namely, PW constant (PWC) functions. We show that the synthesis of optimal PWC SBFs reduces to a constrained minimax optimization problem that includes bilinear terms. We further show that PWC SBFs can obtain the same or even higher probabilistic safety guarantees than continuous SBFs.

We also propose three efficient computational methods for solving the minimax problem by relaxing bilinear terms. This is achieved by (1) dual formulation in form of a linear program (LP), (2) splitting the problem into two separate iterative LPs, where one synthesizes candidate function and the other provide counter examples, and finally (3) designing a convex loss function that precisely captures the objective function of the minimax problem, enabling the use of gradient descent method. To show the expressivity of PWC SBFs and scalability of the proposed computational methods, we provide extensive evaluations on a range of benchmark problems. We also compare performance of the proposed methods against state-of-the-art SOS optimization and NBF learning. The results clearly show the efficacy and superiority of PWC SBFs. The PWC SBFs provide a new proposing path to ensuring safety for stochastic systems.

This study raises several interesting research questions for future investigations. One is how to perform adaptive refinement of the pieces of the PWC SBF to reduce computational overhead. Another interesting question is on the complexity of PW linear (and nonlinear) SBFs and whether they provide more expressivity (better safety guarantees) than PWC SBFs.

References

[1] J. Guiochet, M. Machin, and H. Waeselynck, “Safety-critical Advanced Robots: A survey,” *Robotics and Autonomous Systems*, vol. 94, pp. 43–52, 2017.

[2] S. Shalev-Shwartz, S. Shammah, and A. Shashua, “On a Formal Model of Safe and Scalable Self-driving Cars,” *arXiv preprint arXiv:1708.06374*, 2017.

[3] H. J. Kushner, “Stochastic stability and control,” tech. rep., Brown Univ Providence RI, 1967.

[4] A. Clark, “Control Barrier Functions for Stochastic Systems,” *Automatica*, vol. 130, p. 109688, 2021.

[5] R. Mazouz, K. Muvvala, A. Ratheesh Babu, L. Laurenti, and M. Lahijanian, “Safety Guarantees for Neural Network Dynamic Systems via Stochastic Barrier Functions,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 9672–9686, 2022.

[6] F. B. Mathiesen, S. C. Calvert, and L. Laurenti, “Safety Certification for Stochastic Systems via Neural Barrier Functions,” *IEEE Control Systems Letters*, vol. 7, pp. 973–978, 2022.

[7] S. Prajna, A. Jadbabaie, and G. J. Pappas, “A Framework for Worst-Case and Stochastic Safety Verification using Barrier Certificates,” *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1415–1428, 2007.

[8] C. Santoyo, M. Dutreix, and S. Coogan, “A Barrier Function Approach to Finite-Time Stochastic System Verification and Control,” *Automatica*, vol. 125, p. 109439, 2021.

[9] C. Dawson, Z. Qin, S. Gao, and C. Fan, “Safe nonlinear control using robust neural lyapunov-barrier functions,” in *Conference on Robot Learning*, pp. 1724–1735, PMLR, 2022.

[10] A. A. Ahmadi and A. Majumdar, “DSOS and SDSOS Optimization: LP and SOCP-based Alternatives to Sum of Squares Optimization,” in *2014 48th annual conference on information sciences and systems (CISS)*, pp. 1–5, IEEE, 2014.

[11] H. J. Kushner and D. S. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems*, vol. 26. Springer Science & Business Media, 2012.

[12] A. Lavaei, S. Soudjani, A. Abate, and M. Zamani, “Automated Verification and Synthesis of Stochastic Hybrid Systems: a Survey,” *Automatica*, vol. 146, p. 110617, 2022.

[13] M. Lahijanian, S. B. Andersson, and C. Belta, “Approximate Markovian Abstractions for Linear Stochastic Systems,” in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pp. 5966–5971, IEEE, 2012.

[14] N. Cauchi, L. Laurenti, M. Lahijanian, A. Abate, M. Kwiatkowska, and L. Cardelli, “Efficiency through Uncertainty: Scalable Formal Synthesis for Stochastic Hybrid Systems,” in *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, pp. 240–251, 2019.

[15] F. B. Mathiesen, L. Romao, S. C. Calvert, A. Abate, and L. Laurenti, “Inner Approximations of Stochastic Programs for Data-driven Stochastic Barrier Function Design,” *arXiv preprint arXiv:2304.04505*, 2023.

[16] D. P. Bertsekas and S. E. Shreve, *Stochastic Optimal Control: the Discrete-time Case*, vol. 5. Athena Scientific, 2004.

[17] L. Laurenti and M. Lahijanian, “Unifying Safety Approaches for Stochastic Systems: From Barrier Functions to Uncertain Abstractions via Dynamic Programming,” *arXiv preprint arXiv:2310.01802*, 2023.

[18] 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.

[19] J. Skovbekk, L. Laurenti, E. Frew, and M. Lahijanian, “Formal abstraction of general stochastic systems via noise partitioning,” *IEEE Control Systems Letters*, vol. 7, pp. 3711–3716, 2023.

[20] L. Laurenti, M. Lahijanian, A. Abate, L. Cardelli, and M. Kwiatkowska, “Formal and efficient synthesis for continuous-time linear stochastic hybrid processes,” *IEEE Transactions on Automatic Control*, vol. 66, no. 1, pp. 17–32, 2020.

[21] S. Adams, M. Lahijanian, and L. Laurenti, “Formal Control Synthesis for Stochastic Neural Network Dynamic Models,” *IEEE Control Systems Letters*, vol. 6, pp. 2858–2863, 2022.

[22] S. P. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004.

[23] R. Givan, S. Leach, and T. Dean, “Bounded-Parameter Markov Decision Processes,” *Artificial Intelligence*, vol. 122, no. 1-2, pp. 71–109, 2000.

[24] S. Boyd, L. Xiao, and A. Mutapcic, “Subgradient methods,” *lecture notes of EE392o, Stanford University, Autumn Quarter*, 2003.

[25] A. De Luca, G. Oriolo, and M. Vendittelli, “Stabilization of the Unicycle via Dynamic Feedback Linearization,” *IFAC Proceedings Volumes*, vol. 33, no. 27, pp. 687–692, 2000.

A Extensive Results

The extensive results pertaining to the case studies are presented in Table (A.1).

Table A.1: This table provides extensive results pertaining to the case studies for safety verification based on *piecewise constant* and *continuous* stochastic barriers, as briefly established in Table 1.

			Piecewise Constant Stochastic Barriers											Continuous Stochastic Barriers								
			Dual Linear Program			Counter-Example Guided				Gradient Descent				Neural Barrier Function				Sum-of-Squares				
Model	$\|K\|$	$\mathcal{T}_p(s)$	β	P_s	$\mathcal{T}_0(s)$	β	P_s	$\mathcal{I}$	$\mathcal{T}_o(s)$	β	P_s	$\mathcal{I}$	$\mathcal{T}_o(s)$	η	β	P_s	$\mathcal{T}_o$	Deg	η	β	P_s	$\mathcal{T}_o(s)$
Linear	64	0.02	$8.4e^{-4}$	0.992	0.52	$8.4e^{-4}$	0.992	2	0.04	$4.7e^{-3}$	0.952	50	0.04	$7.1e^{-5}$	$4.1e^{-2}$	0.585	3850.93	4	$2.4e^{-1}$	$1.8e^{-2}$	0.582	0.014
2D	225	0.31	$1.4e^{-4}$	0.998	164.60	$1.4e^{-4}$	0.998	2	0.44	$2.6e^{-3}$	0.973	50	0.20	$2.0e^{-3}$	$5.8e^{-3}$	0.940	3991.47	8	$2.4e^{-1}$	$1.7e^{-2}$	0.582	0.265
Convex	900	8.85	$1.1e^{-4}$	0.999	1087.78	$1.1e^{-4}$	0.999	2	17.93	$1.0e^{-3}$	0.990	150	7.22	$1.9e^{-3}$	$3.7e^{-3}$	0.961	4025.67	30	$1.4e^{-2}$	$7.6e^{-4}$	0.978	151.16
	2500	41.44	$8.1e^{-5}$	0.999	2897.77	$8.1e^{-5}$	0.999	2	88.45	$9.1e^{-4}$	0.998	175	52.78	$1.7e^{-3}$	$2.2e^{-3}$	0.976	4085.65	36	$7.0e^{-3}$	$5.9e^{-5}$	0.992	458.21
Linear	900	5.04	$5.1e^{-2}$	0.494	1197.99	$5.1e^{-2}$	0.494	1	3.79	$5.1e^{-2}$	0.494	50	3.52	$1.5e^{-1}$	$6.1e^{-3}$	0.792	3546.69	12	$9.9e^{-1}$	$1.0e^{-6}$	0.010	0.02
2D	1225	8.20	$2.0e^{-2}$	0.800	1389.78	$2.0e^{-2}$	0.800	1	7.22	$2.0e^{-2}$	0.800	50	6.64	$1.5e^{-1}$	$5.4e^{-3}$	0.844	3579.58	20	$9.9e^{-1}$	$2.3e^{-6}$	0.010	11.08
Non-Convex	1444	9.18	$7.9e^{-3}$	0.921	1545.45	$7.9e^{-3}$	0.921	1	11.65	$7.9e^{-3}$	0.921	55	10.12	$1.4e^{-1}$	$5.2e^{-3}$	0.855	3589.13	24	$9.2e^{-1}$	$5.3e^{-3}$	0.023	37.89
	2926	47.98	$7.2e^{-3}$	0.927	3161.56	$7.2e^{-3}$	0.927	2	98.36	$7.2e^{-3}$	0.927	115	20.86	$1.9e^{-2}$	$5.3e^{-1}$	0.928	3599.85	26	$9.1e^{-1}$	$5.6e^{-3}$	0.034	62.88
	5890	179.94	$7.1e^{-3}$	0.929	8191.65	$7.1e^{-3}$	0.929	3	458.44	$7.1e^{-3}$	0.929	285	133.84	$1.8e^{-2}$	$5.3e^{-1}$	0.929	3675.77	30	$8.7e^{-1}$	$4.7e^{-3}$	0.075	196.85
	11818	477.45	-	-	TO	$6.4e^{-3}$	0.936	3	1875.49	$6.4e^{-3}$	0.936	445	842.67	$1.7e^{-2}$	$5.2e^{-1}$	0.931	3744.23	34	-	-	-	OM
	24336	987.65	-	-	TO	$6.2e^{-3}$	0.938	2	4441.55	$6.2e^{-3}$	0.938	595	3099.30	$1.6e^{-2}$	$4.8e^{-3}$	0.936	4234.56	36	-	-	-	OM
Pendulum	120	6.37	$1.0e^{-6}$	1.00	0.51	$1.1e^{-4}$	0.99	50	5.84	$1.1e^{-3}$	0.989	15000	3.75	$7.7e^{-4}$	$2.1e^{-5}$	0.999	4242.89	4	$5.9e^{-5}$	$2.1e^{-4}$	0.999	7.71
2D	240	18.33	$1.0e^{-6}$	1.00	6.08	$2.2e^{-5}$	1.00	200	14.88	$1.0e^{-3}$	0.990	20000	9.99	$7.6e^{-4}$	$2.1e^{-5}$	0.999	4457.82	4	$4.9e^{-5}$	$2.1e^{-4}$	0.999	34.96
	480	37.84	$1.0e^{-6}$	1.00	29.39	$2.0e^{-6}$	1.00	300	43.42	$8.1e^{-5}$	0.999	40000	17.88	$5.8e^{-4}$	$1.9e^{-5}$	0.999	4675.12	4	$4.8e^{-5}$	$2.0e^{-4}$	0.999	187.60
Unicycle	1250	1103.42	$2.5e^{-2}$	0.750	1000.19	$2.5e^{-2}$	0.750	200	26.37	$2.5e^{-2}$	0.750	300	5.68					2	$9.9e^{-1}$	$1.0e^{-6}$	0.00	3110.21
4D	1800	1756.25	$2.5e^{-2}$	0.975	1719.58	$2.5e^{-2}$	0.975	200	92.26	$2.5e^{-2}$	0.975	20000	25.78					4	$9.9e^{-1}$	$1.0e^{-6}$	0.00	5451.19
	2400	2001.11	$2.1e^{-3}$	0.998	2548.56	$2.1e^{-3}$	0.998	200	145.45	$2.1e^{-3}$	0.998	40000	55.59					6	-	-	-	OM
Quadrotor	7865	80.80	$2.3e^{-2}$	0.770	9906.67	$2.3e^{-2}$	0.770	450	1174.49	$2.3e^{-2}$	0.770	40000	2589.56					2	$8.8e^{-2}$	$3.3e^{-2}$	0.584	0.20
6D	15625	160.61	-	-	TO	$9.9e^{-3}$	0.901	575	3788.98	$9.9e^{-3}$	0.901	50000	3258.87					8	$1.1e^{-3}$	$9.9e^{-3}$	0.900	8628.58
Convex	46656	458.59	-	-	TO	-	-	-	TO	$8.8e^{-3}$	0.912	120000	9542.75					12	-	-	-	OM
Quadrotor	15625	188.10	-	-	TO	$3.3e^{-2}$	0.670	595	3845.25	$3.3e^{-2}$	0.670	55550	3478.31					4	$9.9e^{-1}$	$1.0e^{-6}$	0.00	4.91
6D	31250	395.59	-	-	TO	$1.9e^{-2}$	0.810	615	9548.78	$1.9e^{-2}$	0.810	89850	5878.28					8	$9.9e^{-1}$	$1.0e^{-6}$	0.00	8715.54
Non-Convex	46656	506.99	-	-	TO	-	-	-	TO	$1.0e^{-2}$	0.900	121500	9789.54					12	-	-	-	OM
Quadrotor	65536	845.44	-	-	TO	-	-	-	TO	$5.0e^{-2}$	0.500	198550	19377.90					2	$9.9e^{-1}$	$1.0e^{-6}$	0.00	14830.23
8D	128000	2530.74	-	-	TO	-	-	-	TO	$4.4e^{-2}$	0.560	256700	39132.59					4	-	-	-	OM