Verification and Control for Finite-Time Safety of Stochastic Systems via Barrier Functions

Cesar Santoyo, Maxence Dutreix, and Samuel Coogan

Abstract— **This paper studies the problem of enforcing safety of a stochastic dynamical system over a finite time horizon. We use stochastic barrier functions as a means to quantify the probability that a system exits a given safe region of the state space in finite time. A barrier certificate condition that bounds the infinitesimal generator of the system, and hence bounds the expected value of the barrier function over the time horizon, is recast as a sum-of-squares optimization problem for efficient numerical computation. Unlike prior works, the proposed certificate condition includes a state-dependent bound on the infinitesimal generator, allowing for tighter probability bounds. Moreover, for stochastic systems for which the drift dynamics are affine-in-control, we propose a method for synthesizing polynomial state feedback controllers that achieve a specified probability of safety. Two case studies are presented that benchmark and illustrate the performance of our method.**

I. Introduction

Reliance on complex, safety-critical systems is increasing, which has made safety verification of such systems of utmost importance. For example, environments populated by both humans and autonomous systems (e.g. fulfillment centers, autonomous vehicles, and healthcare) require rigorous safety verification to ensure desired behavior is achieved. From a practical standpoint, safety verification can translate directly to ensuring qualitative guidelines such as collision avoidance are maintained. Safety-critical systems are often analyzed in a purely deterministic framework, however, many real-world applications are subject to stochastic disturbances and are better modeled as stochastic systems.

A common approach to safety verification in deterministic systems is via *barrier functions* which provide Lyapunov-like guarantees regarding system behavior. The existence of a barrier function which satisfies a *barrier certificate* can often be enough to certify the safe operation of a system [1]. Recent work has modified and improved the deterministic form of barrier functions and expanded their application. In particular, control barrier functions have been introduced to guaranteed safety in control affine systems [2], [3]. This is demonstrated in applications for cruise control [3], [4], collision avoidance in robotic swarms [5], and walking robots [6], and has recently been extended to allow for input-to-state safe control barrier functions [7] and to guarantee finite-time convergence to a safe region [8].

In the stochastic setting, safety verification via barrier certificates for infinite time horizons was introduced in [1] alongside the deterministic counterpart. The work presented in [1] provides a framework for bounding the probability a system will exit a safe region based on a non-negative barrier function defined on the system state space. In this approach, the probability is directly correlated with the set of initial conditions. However, this approach can be overly restrictive because it requires the infinitesimal generator, which dictates the expected value evolution of a stochastic process, to be non-positive; i.e., the barrier function is restricted to be a *supermartingale*.

The paper [9] relaxes this condition and instead provides a barrier certificate that only requires the infinitesimal generator of the barrier process to be upper-bounded by a constant. Such processes are called *c-martingales* and allow the expected value of the barrier function to increase over time. This approach results in a safety probability bound for finite time horizons. Recent work in [10] leverages c-martingales for temporal logic verification of discrete-time systems.

The present paper also studies the problem of verifying safety of stochastic systems on finite time horizons, and the contributions are as follows. First, we build on the approaches proposed in [1], [9] and propose a barrier certificate constraint that imposes a state-dependent bound on the infinitesimal generator. This bound was originally proposed and studied by Kushner in [11], [12]. The proposed barrier certificate allows the expected value of the barrier to increase and covers the c-martingale condition of [9] as a special case. However, our formulation also accounts for the system dynamics in the infinitesimal generator constraint. This allows for probability bounds that are no worse than the c-martingale condition, and in many cases, especially with high noise levels, provides better probability bounds.

Second, as in [1], [9], we compute barrier functions using *sum-of-squares* (SOS) optimization. Like in [1], but unlike [9], we utilize polynomial barrier functions. This provides a simpler formulation of the probability of failure on a finite time horizon when compared to the approach in [9] which uses exponential barrier functions and, empirically, provides tighter probability bounds.

Third, we extend our formulation to allow for control inputs and provide a method for synthesizing a safe controller. In particular, we consider affine-in-control systems

C. Santoyo (csantoyo@gatech.edu) and M. Dutreix (maxdutreix@gatech.edu) are with the School of Electrical & Computer Engineering, Georgia Institute of Technology, Atlanta, GA., 30318, USA.

S. Coogan (sam.coogan@gatech.edu) is with the School of Electrical & Computer Engineering and the School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, GA., 30318, USA.

This work was partially supported by NSF under Grant #1749357. C. Santoyo was supported by the NSF Graduate Research Fellowship Program under Grant No. DGE-1650044.

and the proposed approach searches for a polynomial state feedback controller which ensures a system's failure probability achieves a predetermined criterion via a *stochastic control barrier function*.

This paper is organized as follows: Section II covers the background information of stochastic differential equations, barrier functions and SOS optimization. Section III covers the problem which we are solving in detail. Section IV highlights the methodology we utilize to solve the SOS optimization and stochastic control problem. Section V and Section VI present numerical case studies which illustrate our results and conclusions, respectively.

II. Preliminaries

In this section we first introduce our state space definitions as well as background information regarding stochastic processes, barrier functions, and SOS polynomials.

A. Stochastic Process

Consider a complete probability space $(\Omega, \mathcal{F}, P)$ and a standard Wiener process, $w(t)$ in $\mathbb{R}^m$. We consider stochastic processes $x(t)$ satisfying a stochastic differential equation of the form

$$dx = F(x)dt + \sigma(x)dw. \tag{1}$$

The compact set $\mathcal{X} \subset \mathbb{R}^n$ is the system state space, $F : \mathcal{X} \to \mathbb{R}^n$ is the drift rate and $\sigma : \mathcal{X} \to \mathbb{R}^{n\times m}$ is the diffusion term. We assume the functions $F(x)$ and $\sigma(x)$ are Lipschitz continuous. The stochastic process x is a right continuous strong Markov process [13]. We now introduce the infinitesimal generator, which extends the usual definition of a time derivative to instead consider the expectation of a function of a random process.

Definition 1. *Let x be a stochastic process in $\mathbb{R}^n$. The infinitesimal generator $\mathcal{A}$ of x acts on functions of the state space and is defined as*

$$\mathcal{A}B(x) = \lim_{t\downarrow 0} \frac{\mathbb{E}[B(x)|x_0] - B(x_0)}{t}$$

where $B : \mathcal{X} \to \mathbb{R}$ such that the limit exists for all x_0.

In particular, the infinitesimal generator for any process as in (1) is of the form shown in Fact 1.

Fact 1 (Ch. 7, Theorem 7.3.3 of [13]). *Let x be a stochastic process satisfying (1), then the infinitesimal generator $\mathcal{A}$ of some twice differentiable function $B(x)$ is given by*

$$\mathcal{A}B(x) = \sum_{i=1}^{n} F_i(x)\frac{\partial B}{\partial x_i} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\Big(\sigma(x)\sigma^T(x)\Big)_{i,j}\frac{\partial^2 B}{\partial x_i \partial x_j}.$$

The stochastic process x is not guaranteed to lie in $\mathcal{X}$ at all times which leads us to define the stopped process $\tilde{x}$.

Definition 2. *Suppose that τ is the first time of exit of x from the open set Int($\mathcal{X}$). Then the stopped process $\tilde{x}$ is defined by [1]*

$$\tilde{x}(t) = \begin{cases} x(t) & \text{for } t \le \tau \\ x(\tau) & \text{for } t \ge \tau. \end{cases}$$

It is worth noting that the stopped process inherits the same strong Markovian property of x and shares the same infinitesimal generator [11].

B. Barrier Functions

Consider an unsafe region of the state space $\mathcal{X}_u \subseteq \mathcal{X}$ and a set of initial conditions $\mathcal{X}_0 \subseteq \mathcal{X} \setminus \mathcal{X}_u$. In a similar spirit to Lyapunov functions, barrier functions are utilized as a means of guaranteeing a desired behavior on some region of a system's domain defined as a sub-level set (or super-level set) of the barriers. In that regard, stochastic barrier functions have been introduced to upper bound the probability of exiting a safe region over an infinite time-horizon.

Proposition 1 (Theorem 15 from [1]). *Given a stochastic differential equation of the form of (1) and the sets $\mathcal{X}$, $\mathcal{X}_0$, and $\mathcal{X}_u$ with $f(x)$ and $\sigma(x)$ locally Lipschitz continuous, consider the stopped process $\tilde{x}$. Suppose there exists a twice differentiable function B such that*

$$B(x) \le \gamma \ \forall x \in \mathcal{X}_0 \tag{2}$$

$$B(x) \ge 1 \ \forall x \in \mathcal{X}_u \tag{3}$$

$$B(x) \ge 0 \ \forall x \in \mathcal{X} \tag{4}$$

$$\frac{\partial B}{\partial x} f(x) + \frac{1}{2}\mathit{Trace}\Big(\sigma^T(x)\frac{\partial^2 B}{\partial x^2}\sigma(x)\Big) \le 0 \ \ \forall x \in \mathcal{X}. \tag{5}$$

Then, the probability of the system entering the unsafe region of the state space is bounded by

$$P\{\tilde{x}(t) \in \mathcal{X}_u \text{ for some } t \ge 0\} \le B(x_0) \le \gamma \tag{6}$$

where $x_0 \in \mathcal{X}_0$ is the initial state of the system.

This theorem provides a powerful means of bounding the probability of failure of a stochastic process on an infinite time horizon. However, we note that the inequality condition (5), also referred to as the barrier certificate, imposes that $B(x)$ is a supermartingale. This inequality enforces that the expectation of the barrier function decreases at all points of $\mathcal{X}$. In practice, this is often overly restrictive on the system dynamics. For example, it has been shown that no supermartingale exists on a bounded set where the system's noise does not vanish [11]. In Section III, we present a relaxed version of this theorem with its respective probability bounds for finite-time horizons.

C. Sum-of-Squares

Definition 3. *Define $\mathbb{R}[x]$ as the set of all polynomials in $x \in \mathbb{R}^n$. Then*

$$\Sigma[x] \triangleq \left\{ s(x) \in \mathbb{R}[x] : s(x) = \sum_{i=1}^{m} g_i(x)^2, g_i(x) \in \mathbb{R}[x] \right\}$$

is the set of SOS polynomials. It is noted that if $s(x) \in \Sigma[x]$ then $s(x) \ge 0 \ \forall \ x$.

Definition 4. *Given $p_i(x) \in \mathbb{R}[x]$ for $i = 0, \ldots, m$, the problem of finding $q_i(x) \in \Sigma[x]$ for $i = 1, \ldots, \hat{m}$ and*

$q_i(x) \in \mathbb{R}[x]$ for $i = \hat{m}+1, \ldots, m$ such that

$$p_0(x) + \sum_{i=1}^{m} p_i(x) q_i(x) \in \Sigma[x]$$

is a sum-of-squares program (SOSP). SOSPs can be efficiently converted to semidefinite programs using tools such as SOSTOOLS [14].

III. Problem Formulation

The problem we address is: how do we create a bound on the probability a stochastic system of form (1) exits a safe region during a finite-time horizon?

Objectives: First, our goal in this paper is to relax the supermartingale condition on the barrier certificate in (5) similar to what is shown in [9]. Second, based on that relaxation, we aim to derive a state-feedback controller ensuring a user-specified upper bound on the probability of exiting a safe region in the state space.

Consider the stochastic process x which satisfies the stochastic differential equation

$$dx = (f(x) + g(x)u(x))dt + \sigma(x)dw \tag{7}$$

where $f : \mathcal{X} \to \mathbb{R}^n$, $g : \mathcal{X} \to \mathbb{R}^{n\times k}$, $\sigma : \mathcal{X} \to \mathbb{R}^{n\times m}$ and w is a m-dimensional Wiener process. Additionally, $u : \mathcal{X} \to \mathbb{R}^k$ where u is a state dependent control input. We define $F(x) = f(x) + g(x)u(x)$.

Now, we relax the supermartingale condition shown in (5). The following theorem is an immediate corollary of Chapter 3, Theorem 1 in [11].

Theorem 1. *Given the stochastic differential equation shown in (7) and the sets $\mathcal{X} \subset \mathbb{R}^n$, $\mathcal{X}_u \subseteq \mathcal{X}, \mathcal{X}_0 \subseteq \mathcal{X} \setminus \mathcal{X}_u$ with $F(x) = f(x) + g(x)u(x)$ and $\sigma(x)$ locally Lipschitz continuous, where $u(x)$ is some feedback control strategy. Consider the stopped process $\tilde{x}$. Suppose there exists a twice differentiable function B such that*

$$B(x) \le \gamma \ \forall x \in \mathcal{X}_0 \tag{8}$$

$$B(x) \ge 1 \ \forall x \in \mathcal{X}_u \tag{9}$$

$$B(x) \ge 0 \ \forall x \in \mathcal{X} \tag{10}$$

$$\frac{\partial B}{\partial x}F(x) + \frac{1}{2}Trace\left(\sigma^T(x)\frac{\partial^2 B}{\partial x^2}\sigma(x)\right) \le -\alpha B(x) + \beta \quad \forall x \in \mathcal{X}\setminus\mathcal{X}_u \tag{11}$$

for some $\alpha \ge 0$, $\beta \ge 0$ and $\gamma \in [0, 1)$. Define

$$\rho_u := P\{\tilde{x}(t) \in \mathcal{X}_u \text{ for some } 0 \le t \le T\}. \tag{12}$$

Then

- *If $\alpha > 0$ and $\frac{\beta}{\alpha} \le 1$,*

$$\rho_u \le P\left\{\sup_{0\le t\le T} B(\tilde{x}) \ge 1\right\} \le 1 - \Big(1 - B(x_0)\Big)e^{-\beta T}. \tag{13}$$

- *If $\alpha > 0$ and $\frac{\beta}{\alpha} \ge 1$,*

$$\rho_u \le P\left\{\sup_{0\le t\le T} B(\tilde{x}) \ge 1\right\} \le \frac{B(x_0) + (e^{\beta T} - 1)\frac{\beta}{\alpha}}{e^{\beta T}}. \tag{14}$$

- *If $\alpha = 0$,*

$$\rho_u \le P\left\{\sup_{0\le t\le T} B(\tilde{x}) \ge 1\right\} \le B(x_0) + \beta T. \tag{15}$$

The bound shown in (15) is characterized in [10] and [9] as the upper bound on the probability of being unsafe for a c-martingale.

If $B(x)$ satisfies the conditions of Theorem 1, then $B(x)$ is called a *stochastic control barrier function* for a given control policy $u(x)$. Relaxing the supermartingale condition on the infinitesimal generator in the fashion of Theorem 1 gives three case-dependent finite time probability bounds on a system's likelihood of entering an unsafe region in the form of (13), (14), and (15).

Remark 1. *If the initial state x_0 is not known exactly but only known to lie within $\mathcal{X}_0$, then γ can be substituted for $B(x_0)$ in the probability bounds in Theorem 1. This provides an upper bound on the probability of failure over the entire set of initial conditions rather than on a particular initial point in $\mathcal{X}_0$.*

IV. Methodology

In this section we present our approach to construct the stochastic control barrier functions based on the problem formulation of Section II. First, we adapt the constraints given in Theorem 1 to be formulated as an SOSP. Second, we cover the algorithms which construct barrier functions and present our method for computing a low-energy control policy $u(x)$.

A. SOS Formulation for Safety Verification

Theorem 2. *Consider a system of the form of (7) and the sets $\mathcal{X}$, $\mathcal{X}_0$, and $\mathcal{X}_u$ and assume these sets can be described as $\mathcal{X} = \{x \in \mathbb{R}^n : s_{\mathcal{X}}(x) \ge 0\}$, $\mathcal{X}_0 = \{x \in \mathbb{R}^n : s_{\mathcal{X}_o}(x) \ge 0\}$, and $\mathcal{X}_u = \{x \in \mathbb{R}^n : s_{\mathcal{X}_u}(x) \ge 0\}$ for some polynomials $s_{\mathcal{X}}$, $s_{\mathcal{X}_o}$, and $s_{\mathcal{X}_u}$. Suppose there exists a polynomial $B(x)$, a polynomial $u(x)$, and SOS polynomials $\lambda_{\mathcal{X}}(x)$, $\lambda_{\mathcal{X}_o}(x)$, and $\lambda_{\mathcal{X}_u}(x)$ that satisfy the following*

$$B(x) - \lambda_{\mathcal{X}}(x)s_{\mathcal{X}}(x) \in \Sigma[x]$$

$$B(x) - \lambda_{\mathcal{X}_u}(x)s_{\mathcal{X}_u}(x) - 1 \in \Sigma[x]$$

$$-B(x) - \lambda_{\mathcal{X}_o}(x)s_{\mathcal{X}_o}(x) + \gamma \in \Sigma[x]$$

$$-\frac{\partial B(x)}{\partial x}F(x) - \alpha B(x) + \beta + \lambda_{\mathcal{X}_u}(x)s_{\mathcal{X}_u}(x) - \lambda_{\mathcal{X}}(x)s_{\mathcal{X}}(x) \in \Sigma[x]$$

where $F(x) = f(x) + g(x)u(x)$. Then, the probability of failure, depending on the values of α and β, is defined by (13), (14) or (15).

We omit the proof due to space constraints, but the proof follows the general approach for relaxing set constraints to SOS programs using the *Positivstellensatz* condition; see the documentation of [14] for details.

Algorithm 1 Compute $B(x)$

```
procedure COMPUTE-$B(l_\alpha, u_\alpha, \sigma, u(x), n_B)$
    $\alpha \leftarrow Range(l_\alpha, u_\alpha, d)$    ▷ Assign $\alpha$ values $d$ apart
    $P^* \leftarrow 1$
    $P \leftarrow \emptyset$
    for $\alpha_0 \in \alpha$ do
        min $B(x_0) + \beta$
        subject to    $B(x) - \lambda_{\mathcal{X}} s_{\mathcal{X}}(x) \geq 0$
                      $-\mathcal{A}B(x) + \alpha_0 B(x) - \beta$
                      $+\lambda_{\mathcal{X}_u} s_{\mathcal{X}_u}(x) - \lambda_{\mathcal{X}}(x) s_{\mathcal{X}}(x) \geq 0$
                      $-B(x) - \lambda_{\mathcal{X}_o} s_{\mathcal{X}_o}(x) + \gamma \geq 0$
                      $B(x) - 1 - \lambda_{\mathcal{X}_u} s_{\mathcal{X}_u}(x) \geq 0$

        Compute $P$, using (13), (14) or (15)
        if $P < P^*$ then
            $\alpha^* = \alpha_0$
            $\beta^* = \beta$
            $P^* = P$
        end if
    end for
    return $\alpha^*, \beta^*, P^*$
end procedure
```

B. Barrier Function Numerical Procedure

Next, we present an algorithmic solution to this problem. Algorithm 1 computes the barrier function $B(x)$ used to quantify an upper bound on the failure probability. The input values $l_\alpha, u_\alpha, \sigma, u(x), n_B$ are the lower α range value, upper α range value, diffusion term, control polynomial, and the order of the $B(x)$ polynomial, respectively. Our algorithm performs a grid search over a range of scalar α with value spacing d,which are design parameters. Next, the SOSP is encoded using the constraints shown in Theorem 2. Lastly, as the SOSP is run, the algorithm returns a function, $B(x)$, that is evaluated at any $x_0 \in \mathcal{X}_0$ and utilized to compute the probability, P, using (13), (14) or (15). The degree of $B(x)$ is a design parameter; however, higher order polynomials tend to produce tighter bounds. Well refined bounds (i.e. higher order polynomials) present themselves with the trade-off of longer computational times versus probability of failure refinement.

The objective of the SOSP in Algorithm 1 is set to minimize the value, $B(x_0) + \beta$. Minimizing $B(x_0) + \beta$ is a consensus objective which may not be the best one but provides a means of avoiding bi-linear programs.

C. Controller Synthesis Procedure

In general, when searching for a control policy, we are aiming for a polynomial of the same or lower order of $B(x)$ such that the upper bound on the probability of failure reduces to a designer specified value. First, we write the polynomial $u(x)$ in quadratic form as

$$u(x) = z^T Q z \tag{16}$$

where z is a vector of monomials in x of a specified order and Q is a coefficient matrix of appropriate dimensions. Because there likely exist many feasible controllers ensuring the desired probability of failure, we introduce a cost criterion to choose among them. We approximate the energy of a particular control policy via a proxy measure. In this case, the proxy is the non-negative scalar, c, such that the following vector element-wise constraints

Algorithm 2 Initialize $u(x)$

```
procedure COMPUTE-$u(B(x), \alpha, \beta, n_u)$
    $u(x) = z^T Q z$    ▷ $u(x)$ is an $n_u$ power polynomial
                        ▷ $z$ is a vector of state monomials
    min $c$
    subject to    $c\mathbb{1} - \mathrm{vec}(Q) \geq 0$
                  $\mathrm{vec}(Q) + c\mathbb{1} \geq 0$
                  $-\mathcal{A}B(x) + \alpha B(x) - \beta$
                  $+\lambda_{\mathcal{X}_u}(x) s_{\mathcal{X}_u}(x) - \lambda_{\mathcal{X}}(x) s_{\mathcal{X}}(x) \geq 0$
    return $u(x), c, Q$
end procedure
```

$$c\mathbb{1} - \mathrm{vec}(Q) \geq 0$$

$$\mathrm{vec}(Q) + c\mathbb{1} \geq 0$$

hold where $\mathrm{vec}(Q)$ is the vector form of matrix Q and $\mathbb{1}$ is the vector of ones of appropriate dimension. We choose the cost $\min c$ to minimize the coefficients appearing in the polynomial controller to encourage lower control effort. This objective and procedure are highlighted in Algorithm 2.

Algorithm 3 takes $P_{goal}, \sigma, \alpha, n_B, n_u$ and ϵ as arguments. These variables are the goal probability, diffusion term, α multiplier on $B(x)$, barrier polynomial order, control polynomial order and a small offset, respectively. Once the procedure begins, it runs until the probability of failure is within ϵ of the predefined goal probability. It is possible to use other conditions to determine whether the algorithm should continue to run such as computing the change in optimal scalar c value, c^*, between iterations. Additionally, the algorithm may be sped up by using a floor value for c. Initially, a polynomial barrier of a specified polynomial power is computed given no control policy (i.e. $u(x) = 0$). Generally speaking, as in our case studies, we are interested in systems where the probability of failure with no control action is above the goal probability.

Next, if the probability of failure is greater than P_{goal} then we compute a scaled down β value multiplied by a_{dec}. If the failure probability is less than P_{goal} then we scale up the β by a_{inc}. The intuition behind this comes from analyzing the probability bounds (13), (14) or (15). In general, a lower β reduces our failure probability, thus when searching for $u(x)$ a scaled version of β can be used. The values of a_{inc} and a_{dec} are also design parameters.

V. CASE STUDIES

In this section, we first present a simple academic example to illustrate the our technique. Second, we present a nonlinear example to demonstrate the versatility of our approach. Both case studies are compared to a Monte Carlo simulation which is considered ground truth. We utilize SOSTOOLS [14] which converts our SOSP into semidefinite programs. Our choice of solver is the semidefinite program solver SDPT3 [15], [16]. These case studies were conducted on a 2.3 GHz Intel Core i5 computer with 8GB of memory.[1]

[1] The MATLAB source code for the two case studies is contained at `https://github.com/gtfactslab/stochasticbarrierfunctions`

Algorithm 3 Search for control polynomial $u(x)$

```
procedure COMPUTE-$u_{min}(P_{goal}, \sigma, \alpha, n_B, n_u, \epsilon)$
    $i_{count} = 1$                                   ▷ Initialize counting variable
    while $|P^* - P_{goal}| > \epsilon$ do
        if $i_{count} = 1$ then
            $\beta, P \leftarrow$ COMPUTE-$B(l_\alpha, u_\alpha, \sigma, u(x), n_B)$
                                                       ▷ Since $\alpha$ fixed, $l_\alpha = u_\alpha$
                                                       ▷ $u(x) = 0$
            $i_{count} = i_{count} + 1$
        else
            $u(x), c, Q \leftarrow$ COMPUTE-$u(B(x), \alpha, \beta, n_u)$
            $\beta, P \leftarrow$ COMPUTE-$B(l_\alpha, u_\alpha, \sigma, u(x), n_B)$
        end if

        if $P < P_{goal}$ and $c < c^*$ then
            $\beta^* = \beta$
            $P^* = P$
            $c^* = c$
        end if
                                                       ▷ $c^*$ is initialized as a large number
        if $P > P_{goal}$ then
            $\beta = a_{dec}\beta$
        else
            $\beta = a_{inc}\beta$
        end if
                                                       ▷ $a_{inc}$ and $a_{dec}$ are scaling factors

    end while
    return $u^*(x), c*, Q$
end procedure
```

A. 1-D Stochastic System

Consider a 1-D stochastic control affine system of the form

$$dx = \big(-x + u(x)\big)dt + \sigma dw. \tag{17}$$

This is of the same form as (7) where $f(x) = -x$ and $g(x) = 1$. We define our state space as $\mathcal{X} = \{x : -2 \leq x \leq 2\}$, $\mathcal{X}_u = \{x : x^2 \geq 1\}$, and $\mathcal{X}_0 = \{x : x^2 \leq .2^2\}$. First, we benchmark the probability of failure without a control input (i.e. $u(x) = 0$) for a finite time horizon of $T = 1$ s. Thus, to do so, we follow the procedure outlined in Algorithm 1. We grid search over a defined range of values for the constant α. In this particular example, our $\alpha \in [0, 5]$ with $d = .05$ in Algorithm 1.

We choose to search for $B(x)$ of the 16th degree. Additionally, we reproduce the c-martingale bound presented in [9, Algorithm 3]. Lastly, we benchmark against the true probability of failure created via a 5000 draw Monte Carlo simulation. The results are presented in Fig. The 1.

In Fig. 1, we see that our polynomial bound on the probability of failure performs better than the bound from [9] generated using the c-martingale condition that is not state dependent. The difference is particularly notable at higher noise levels where the exponential bound from [9] becomes trivial, i.e., greater than or equal to one.

Next, we address the control problem of achieving a particular bound on the probability of failure of this system. We choose a desired failure probability $P_{goal} = .30$. We restrict our attention to a linear controller of the form $u(x) = -kx$. Our search for a low-energy controller which successfully fulfills the design requirement follows a modified binary search version of Algorithm 3. Fig. 2 plots k^* achieving the desired failure probability bound for $\sigma \in [1, 2]$. Here, we note that the degree of barrier function for which we search greatly affects the control gain needed to achieve the control objective. In some sense, searching for a higher-order polynomial refines the probability of failure bound requiring lower control effort; however, these high order polynomials require more computation time. Eventually, the degree of the polynomial reaches a saturation point where it does not further decrease the k^* required.

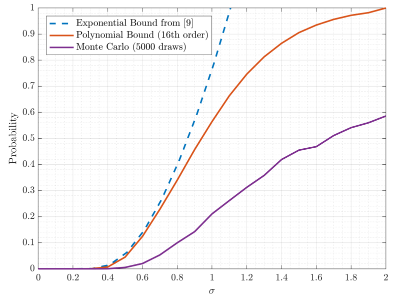

Fig. 1: The probability of failure bounds for the 1-D system are presented here. The polynomial barrier function, $B(x)$ considered here was of the 16th degree. The Monte Carlo simulation results illustrate the true probability of failure for this system.

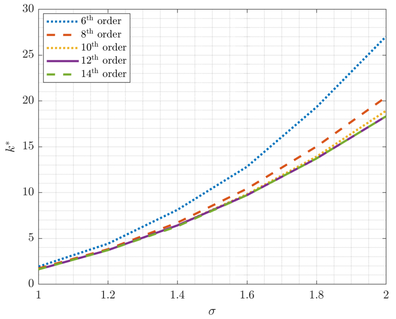

Fig. 2: An illustration of the trade-off between required control gain and the degree of the barrier function, $B(x)$ needed to successfully attain the desired probability of failure threshold. Using higher order polynomials allows us to guarantee that the desired probability bound is satisfied for a smaller control gain up until some point.

B. Nonlinear Dynamics

Consider the stochastic non-linear dynamics

$$dx_1 = x_2 dt \tag{18}$$

$$dx_2 = \Bigg(-x_1 - x_2 - 0.5x_1^3 + u(x)\Bigg)dt + \sigma dw. \tag{19}$$

This system is studied in [17] without the input term $u(x)$.

We define our state space as $\mathcal{X} = \{(x_1, x_2) \mid -3 \leq x_1 \leq 2, -2 \leq x_2 \leq 3\}$, $\mathcal{X}_u = \{x_2 \mid x_2 \geq 2.25\}$, and $\mathcal{X}_0 = \{(x_1, x_2) | (x_1 + 2)^2 + x_2^2 \leq 0.1^2\}$. A sample trajectory of (18)–(19) is illustrated in Fig. 3. Additionally, level sets of $B(x)$ are projected onto the state space. In this illustration, $B(x)$ is computed with $u(x) = 0$ solely using Algorithm 1. Here, we see that the values for the barrier function abide by the definitions of Theorem 1. In this particular trajectory illustration, the evolution of system noise is enough for the system to enter the predefined unsafe set; however, this is not always the case. To illustrate this, we compute a Monte Carlo

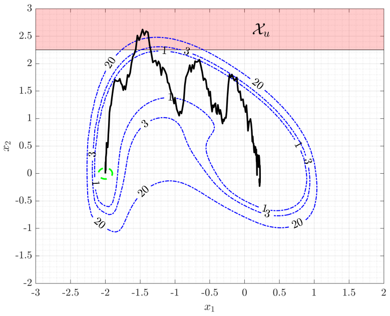

Fig. 3: Given the initial conditions $x_0 = [-2, 0]$, the single trajectory dynamics for time horizon of $T = 2$ and a $\sigma = 1.0$ is illustrated. We define the unsafe region as $\mathcal{X}_u = \{x_2 \mid x_2 \geq 2.25\}$. Additionally, the level sets of $B(x)$ and their respective values are labeled and given as dashed blue lines.

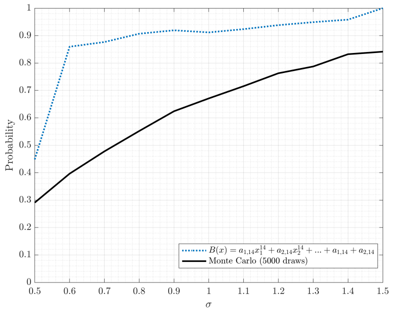

Fig. 4: Computing a 14th order polynomial barrier function for the nonlinear dynamics we are able to bound the probability of failure of the 5000 draw Monte Carlo dynamics for constant noise levels $\sigma \in [.5, 1.5]$.

simulation of the system dynamics shown. Additionally, an upper bound is computed on the probability of becoming unsafe given our initial condition and illustrated in Fig. 4. While we encode a set of initial conditions into the SOSP, we evaluate the probability bound at the same initial point, $x_0 \in \mathcal{X}_0$, as the Monte Carlo simulation.

The design specification for this example is to reduce the probability of failure bound to $P_{goal} = .10$ for specified noise levels. For this example we create a 2nd order polynomial controller of the form of (16). We look to minimize the constant, c, highlighted in Algorithm 2. We run the $u(x)$ search algorithm for select noise levels, specified α values, and present the results in Table 1. The α values in this table originate from the initial (i.e. $u(x) = 0$) probability bound computation.

Noise Level, σ	$\mathbf{P_{u(x)=0}}$	α	min $\mathbf{c}$
0.6	0.860	1.4	2.1821
0.9	0.919	1.3	0.5251
1.0	0.912	1.3	0.6396
1.3	0.949	1.5	1.1488

TABLE 1: The results from the search for a control polynomial u(x) which reduces the probability of failure to $P_{goal} = .10$. The probability of failure without a given control input is presented here for comparison.

VI. Conclusion

We consider control barrier functions whose existence gives a means of quantifying an upper bound on a system's probability of failure. Additionally, we present a novel, state dependent approach to the problem of finite-time verification which further relaxes the constraint on the evolution of the expected value. Lastly, we synthesize a feedback control strategy $u(x)$ such that a certain probability of failure criteria is met. We illustrate our methods with two case studies which demonstrate our ability to quantify system failure probabilities. In these case studies, we solve for the barrier function polynomials using SOS optimization and demonstrate our proposed approach outperforms existing methods.

References

[1] S. Prajna, A. Jadbabaie, and G. Pappas, "A framework for worst-case and stochastic safety verification using barrier certificates," *Automatic Control, IEEE Transactions on*, vol. 52, no. 8, pp. 1415–1428, 2007.

[2] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 462–467, 2007.

[3] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *Automatic Control, IEEE Transactions on*, vol. 62, no. 8, pp. 3861–3876, 2017.

[4] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *IEEE Conference on Decision and Control (CDC)*. IEEE, 2014, pp. 6271–6278.

[5] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *Robotics, IEEE Transactions on*, vol. 33, no. 3, pp. 661–674, 2017.

[6] S.-C. Hsu, X. Xu, and A. D. Ames, "Control barrier function based quadratic programs with application to bipedal robotic walking," in *American Control Conference (ACC)*. IEEE, 2015, pp. 4542–4548.

[7] S. Kolathaya and A. D. Ames, "Input-to-state safety with control barrier functions," vol. 3, no. 1, 2018.

[8] A. Li, L. Wang, P. Pierpaoli, and M. Egerstedt, "Formally correct composition of coordinated behaviors using control barrier certificates," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018.

[9] J. Steinhardt and R. Tedrake, "Finite-time regional verification of stochastic non-linear systems.(special issue on robotics: science and systems 2011)(technical report)," *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 901–923, 2012.

[10] P. Jagtap, S. Soudjani, and M. Zamani, "Temporal logic verification of stochastic systems using barrier certificates," *CoRR*, vol. abs/1807.00064, 2018. [Online]. Available: http://arxiv.org/abs/1807.00064

[11] H. J. Kushner, *Stochastic stability and control*, ser. Mathematics in science and engineering, v.33. New York: Academic Press, 1967.

[12] H. Kushner, "Finite time stochastic stability and the analysis of tracking systems," *Automatic Control, IEEE Transactions on*, vol. 11, no. 2, pp. 219–227, 1966.

[13] B. Øksendal, *Stochastic differential equations : an introduction with applications*, 5th ed., ser. Universitext. Berlin ; New York: Springer, 1998.

[14] A. Papachristodoulou, J. Anderson, G. Valmorbida, S. Prajna, P. Seiler, and P. A. Parrilo, *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*, `http://arxiv.org/abs/1310.4716`, 2013, available from `http://www.eng.ox.ac.uk/control/sostools`, `http://www.cds.caltech.edu/sostools` and `http://www.mit.edu/~parrilo/sostools`.

[15] K. Toh, M. Todd, and R. Tutuncu, "Sdpt3 - a matlab software package for semidefinite programming, version 1.3," *Optimization Methods & Software*, vol. 11-2, no. 1-4, pp. 545–581, 1999.

[16] K. C. Toh, M. J. Todd, and R. Tutuncu, "Solving semidefinite-quadratic-linear programs using sdpt3," *Mathematical Programming*, vol. 95, no. 2, pp. 189–217, 2003.

[17] S. Prajna, A. Jadbabaie, and G. J. Pappas, "Stochastic safety verification using barrier certificates," in *IEEE Conference on Decision and Control, 2004*. IEEE, 2004, pp. 929–934.