Finite-Time Regional Verification of Stochastic Nonlinear Systems

Jacob Steinhardt
Department of Mathematics
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139-4307
Email: jsteinha@mit.edu

Russ Tedrake
Department of Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139-4307
Email: russt@mit.edu

Abstract—**Recent trends pushing robots into unstructured environments with limited sensors have motivated considerable work on planning under uncertainty and stochastic optimal control, but these methods typically do not provide guaranteed performance. Here we consider the problem of bounding the probability of failure (defined as leaving a finite region of state space) over a finite time for stochastic nonlinear systems with continuous state. Our approach searches for exponential barrier functions that provide bounds using a variant of the classical supermartingale result. We provide a relaxation of this search to a semidefinite program, yielding an efficient algorithm that provides rigorous upper bounds on the probability of failure for the original nonlinear system. We give a number of numerical examples in both discrete and continuous time that demonstrate the effectiveness of the approach.**

I. Introduction

Consider the problem of a legged robot quickly traversing unknown rough terrain, a vision-based autonomous vehicle flying through a dense forest at high speeds, or a mobile manipulator fetching a beer out of the refrigerator. Each of these robots will be subject to many sources of uncertainty — including uncertainty from imperfect perception, imperfect models of robot and environment, and any unexpected disturbances. At the same time, we hope that our robots are able to accomplish their tasks by executing high-speed dynamic maneuevers, which demands that a high-performance control system will have to reason about the nonlinear dynamics of the machine. While there has been considerable progress recently in designing impressive control systems for this class of machine (e.g., [2, 6, 16, 27, 23, 22]), there is relatively little work on guaranteeing that these systems can achieve their goals in the presence of significant uncertainty.

In particular, we are motivated by the case where the effects of the uncertainty are large when compared to the control authority and passive stability of the robot (e.g. the small vision-based UAV flying through a cluttered environment in a strong wind). Consider, for instance, a closed-loop maneuver that is only locally stable for the deterministic plant, subjected to an unbounded uncertainty (for instance, Gaussian) from a perceptual system. In this case, the system will eventually be unstable with probability 1; hence, robust control design and verification methods that consider worst-case performance are not appropriate. Instead, here we attempt to analyze the stochastic stability of the nonlinear system over a finite time horizon — a framework considered in [12], which is also a special case of planning with chance constraints as formulated in [5]. In particular, we would like to verify an *activation set* — a set of intial conditions from which the control policy will provably achieve its goal with a desired probability. In addition to certifying performance, efficient algorithms for verifying this stochastic stability will lend themselves naturally to improved methods for feedback design and planning algorithms under chance constraints.

A common approach to nonlinear systems is via a finite-dimensional interpolation of the state-space — either by direct discretization, or through some more sophisticated technique like volumetric interpolation. However, this approach has multiple shortcomings — first, such approximations can end up having large effects on the result, even for relatively large numbers of interpolating functions and for well-behaved systems. Second, for more than a few dimensions there will not be enough memory on the computer to store even coarse approximations to the continuous-state dynamics (for instance, a discretization-based approach in a recent paper hits computation limits around 5 to 8 dimensions [1]; we can solve a similar problem in 10 dimensions). For both of these reasons we have been led to consider continuous-state verification. In other words, we would like to perform verification directly on the original system instead of first making a finite-dimensional approximation.

Unfortunately, it appears that so far little progress has been made on the problem of continuous-state, nonlinear, stochastic verification, although many special cases have been studied. If we eliminate stochasticity, then we can perform sum-of-squares verification on Lyapunov functions [21, 25, 26, 15, 17]. If we eliminate continuous-state, then exact solutions can be found by taking a matrix exponential (in continuous-time) or matrix power (in discrete-time) of the transition matrix for the Markov process, after adding an appropriate absorbing state to capture all of the failed states. If we assume linearity of the system then the problem falls into the risk-sensitive control framework [10], which handles not only verification but control design. Risk-sensitive control also deals with nonlinear systems, but in the nonlinear case typically requires a discretization of the state space, which is problematic for the reasons discussed above.

There has been some progress on dealing with the general case. The main approach is to find supermartingales of the system, which bound the probability of leaving a region [3]. These supermartingales can be thought of as stochastic analogues of Lyapunov functions, and are called barrier functions in [19]. They can alternately be thought of as upper bounds on a certain cost-to-go function.

However, unless the noise goes to zero near the desired point of stability, no supermartingale exists (assuming the dynamics of the system are sufficiently differentiable). This requires a slight variation on the supermartingale criterion, as in Kushner [12], which gives bounds very similar to our Theorems II.1 and II.2. In fact, the mathematical theory of [12] is more complete than that presented here, as Kushner derives a version of Theorem II.2 with the correct asymptotic form (exponential, rather than linear, decay of the success probability as time increases). Kushner also works through many nice examples of applying the derived bounds to different types of noise. The drawback of Kushner's work is that it does not provide any general algorithms for finding good supermartingales. We hope to remedy this with our work.

Much of the continuous-state verification research has focused on nonlinear systems with Gaussian and switching noise [19, 28]. In this paper we will focus on just Gaussian noise, although we believe that extending the techniques to include switching systems should not be too difficult. This is because the theory presented here holds for general Markov processes, with the computational results we provide tailored to Gaussian noise.

The results in [19] are as far as we know the first to provide an algorithm for finding supermartingales. However, their approach has a few shortcomings that we address. The first is that their method requires their barrier function to be a true supermartingale, which for a time-invariant barrier function requires them to pre-suppose stochastic stability for sufficiently small initial conditions, a condition which is difficult to check and not always true. A second issue is that they search over polynomially growing barrier functions, which will not give as strong of guarantees as exponential barrier functions. At the same time, while it is tractable to search over relatively high-degree barrier functions in the CT case, we believe that such a search becomes quickly infeasible in the DT case because the Lyapunov function composed with the dynamics leads to a polynomial whose degree is the product of the degrees of the dynamics and Lyapunov functions; this belief is based mainly on our own efforts to apply the methods of [19] to the DT case, as [19] only considers the CT case.

To summarize, we are interested in bounding the probability that a nonlinear, possibly time-varying, system with Gaussian noise leaves a region (either pre-specified or computed as part of the optimization) in a certain time interval. We will do this by using the supermartingale approach discussed in [12], searching over a family of exponentially growing barrier functions. We will use sum-of-squares programming to identify a member of this class that provides a good bound on the failure probability.

We start in Section II by presenting Kushner's bounds on failure probability. In Section II we also give an overview of sum-of-squares programming, an optimization technique that will be important for finding a good barrier function. Next, in Section III, we will define the family of barrier functions that we intend to search over, and provide semidefinite constraints that allow us to bound the failure probability. In Section IV, we go over specific practical details of how we search for a certificate that provides a good upper bound. We conclude in Section V by providing examples of our approach on the simple pendulum, cart-and-pole, and rimless wheel systems, as well as for the heating system described in [1].

II. Background

A. A Bound on Markov Chains

We begin with an extension of the classical result about stability of supermartingales. Recall that a *supermartingale* is a function $B(x,t)$ of a Markov process such that $\mathbb{E}[B(x(t+\Delta t),t+\Delta t) \mid x(t)] \leq B(x(t),t)$ for all $\Delta t \geq 0$. We will instead consider functions that are almost supermartingales, in the sense that $\mathbb{E}[B(x(t+\Delta t),t+\Delta t)] \mid x(t)] \leq B(x(t),t) + \int_t^{t+\Delta t} c(s)ds$ for some function c that depends only on time. We will call such functions *c-martingales*. In discrete time, we instead consider the condition $\mathbb{E}[B(x(t+1),t+1)] \mid x(t)] \leq B(x(t),t) + c(n)$. In continuous time, a sufficient condition for being a c-martingale is that $\mathcal{A}\,B(x(t),t) \leq c(t)$, where $\mathcal{A}$ is the infinitesimal operator:

$$\mathcal{A}\,B(x(t),t) = \lim_{t' \downarrow t} \frac{\mathbb{E}[B(x(t'),t') \mid x(t)] - B(x(t),t)}{t'-t}. \quad (1)$$

We require this limit to converge uniformly across all $x(t)$ and t, which implies that B must be a continuous function of both x and t. For a more detailed treatment of the technical issues surrounding statistics of Markov processes, see Dynkin's book on the subject [7]; we refer the reader in particular to equations (1.2) and (5.8) and the surrounding exposition.

The relaxation of the supermartingale condition to the c-martingale condition allows us to consider systems that are only locally stable and have non-zero noise at the origin. It is similar to the approach taken in [18] for contracting systems.

We can draw an analogy between c-martingales and amortized analysis in computer science — if there is some function of our state that increases slowly, then it will be a long time before it can reach a large value. If we can find a function B of our state that increases slowly in expectation (such as a c-martingale), and B is large outside of a region of state space, then it will take a long time for a trajectory of the system to escape that region. We more formally define the *escape probability* from a region R at time T as the probability that a trajectory of the system leaves R by time T (this includes leaving R before time T, even if it later re-enters; we also allow for the possibility that R is time-varying).

Our main theorems are given below. We only prove the continuous-time version, as the discrete-time proof is essentially the same but without the extra analytical technicalities.

For another exposition of these same ideas, see Theorem 1 of [11].

Theorem II.1. *Let $\mathcal{M}$ be a Markov chain over a space X with initial condition $x(0)$, let R be an open subset of X, and let B be a non-negative real-valued function on $X \times [0,T]$. Suppose that B is a c-martingale inside R, and that $B(x,t) \geq B_0$ for all $x \notin R$, $0 \leq t < T$. Then the escape probability at time T is at most $\frac{B(x(0),0)+\sum_{n=0}^{T-1} c(n)}{B_0}$.*

Theorem II.2. *Let $\mathcal{M}$ be a strong Markov process over a space X whose trajectories are almost surely right-continuous. Let $x(0)$ be the initial condition of the Markov process, let R be an open subset of X, and let B be a non-negative real-valued function on $X \times [0,T]$. Suppose that B is a c-martingale inside R, and that $B(x,t) \geq B_0$ for all $x \notin R$, $0 \leq t < T$. Then the escape probability at time T is at most $\frac{B(x(0),0)+\int_0^T c(t)dt}{B_0}$.*

Proof of Theorem II.2.: Modify $\mathcal{M}$ to a new Markov process $\mathcal{M}'$ that stops as soon as a trajectory leaves R. More formally, if $x(t_0) \notin R$, then $x(t) = x(t_0)$ for all $t \geq t_0$. Also add a state variable τ that is equal to t up until the time t_0 and that is equal to t_0 for all $t \geq t_0$.

With this new Markov process defined, $\mathcal{A}\,B(x(t),\tau(t)) \leq c(t)$ holds across all of X. Consequently, we have $\mathbb{E}[B(x(T),\tau(T)) \mid x(0)] \leq B(x(0),0) + \int_0^T c(t)dt$. Since B is non-negative, by Markov's inequality we must have $\mathbb{P}[B(x(T),\tau(T)) \geq B_0 \mid x(0)] \leq \frac{\mathbb{E}[B(x(T),\tau(T)) \mid x(0)]}{B_0} \leq \frac{B(x(0),0)+\int_0^T c(t)dt}{B_0}$. Since $B(x,\tau) \geq B_0$ for all $x \notin R$, $0 \leq \tau \leq T$, we also have $\mathbb{P}[x(T) \notin R \mid x(0)] \leq \mathbb{P}[B(x(T),\tau(T)) \geq B_0 \mid x(0)]$. On the other hand, since $\mathcal{M}'$ stops upon leaving R, the escape probability of $\mathcal{M}$ at time T is exactly the probability that $x(T) \notin R$ for $\mathcal{M}'$, which proves the theorem. ■

In the following sections, we will discuss how to usefully apply this bound to dynamical systems with Gaussian noise.

B. Sum-of-Squares Programming

Suppose that we want to compute the global minimum of a polynomial $p(x_1, \dots, x_n)$. We could formulate this as maximizing δ subject to the constraint $p(x) - \delta \geq 0$ for all x. This problem is NP-hard in general; however, if we could write $p(x) - \delta = h(x)^T Q h(x)$ for some matrix $Q \succeq 0$, then we would know that $p(x) \geq \delta$ for all x. We can more generally consider programs with linearly parameterized polynomials and several positivity constraints, e.g.

$$\begin{aligned} &\underset{\alpha}{\text{maximize}} && h^T \alpha \\ &\text{subject to} && \alpha^T p_i(x) \geq 0, \quad i = 1, \dots, m, \end{aligned} \tag{2}$$

which are then replaced with

$$\begin{aligned} &\underset{\alpha, Q}{\text{maximize}} && h^T \alpha \\ &\text{subject to} && \alpha^T p_i(x) = h_i(x)^T Q_i h_i(x), && i = 1, \dots, m \\ & && Q_i \succeq 0, && i = 1, \dots, m. \end{aligned} \tag{3}$$

Note that (3) is a semidefinite program, and can thus be solved efficiently. The α are referred to as decision variables and the x are referred to as free variables.

We may also wish to only enforce a constraint $p_i(x) \geq 0$ in some region described by $q_i(x) \leq 0$. In this case, we can introduce a *Lagrange multiplier* $\lambda(x)$ and impose the constraints $p_i(x) + \lambda(x) q_i(x) \geq 0$ and $\lambda(x) \geq 0$. If q_i is not fixed then the constraint is no longer linear, an issue we deal with in Section IV.

Sum-of-squares programs can be formulated using the MATLAB package yalmip [13]. Yalmip is a modeling language for optimization problems that has built-in support for several optimizers; we used SeDuMi [24] for our work. While the final version of our code uses yalmip, we also used CVX [8, 9] and SOSTOOLS [20] during development. All of the software mentioned here is freely available online.

III. Certificates of Stability

Theorems II.1 and II.2 show us how to obtain true certificates of stability from approximate certificates. In order to usefully apply these theorems, we need to pick a suitable barrier function for a given noise model. For now, we will consider systems with polynomial dynamics and (possibly state-dependent) Gaussian noise. In the DT case, this means systems of the form $x_{n+1} = f(x_n) + g(x_n) w_n$, where w_n is unit covariance white noise. In the CT case, this means systems of the form $dx(t) = f(x)dt + g(x)dw(t)$, where w is a vector of independent Wiener processes. All of the following results also hold for time-varying f and g, but we will omit the possible dependence on t to keep the equations more readable.

We will consider barrier functions of the form $B_S(x,t) = e^{\frac{1}{2}x^T S(t) x} - 1$. Note that including cubic or higher terms in the exponent would make the expected value of B_S infinite with respect to Gaussian noise.

A. Discrete-Time

In discrete-time, we can compute

$$\begin{aligned} &\mathbb{E}[B_S(x(t+1)) \mid x(t)] = \\ &\qquad \det(I - g^T S g)^{-\frac{1}{2}} e^{\frac{1}{2} f(x)^T S \left(S - S g g^T S\right)^{-1} S f(x)} - 1. \end{aligned}$$

Applying Theorem II.1 to B_S lets us bound the failure probability by

$$\frac{e^{\frac{1}{2}x(0)^T S(0) x(0)} - 1 + \sum_{n=1}^{N} C(n)}{e^{\frac{1}{2}\rho} - 1} \tag{4}$$

as long as $x^T S(n) x \geq \rho$ for all $x \notin R_n$ and

$$\begin{aligned} &C(n) \geq -e^{\frac{1}{2}x^T S(n-1) x} + \\ &\qquad \det(I - g^T S(n) g)^{-\frac{1}{2}} e^{\frac{1}{2} f^T S(n) (S(n) - S(n) g g^T S(n))^{-1} S(n) f} \end{aligned}$$

whenever $x^T S(n) x < \rho$. The expression for $C(n)$ is cumbersome, as it involves a determinant as well as the difference of two exponential functions. The following two lemmas let us relax the expression to a condition on polynomials.

Lemma III.1. $\det(I - M) \geq 1 - \mathrm{Tr}(M)$ *when* $0 \preceq M \preceq I$.

Proof: This is the same as showing that $\prod_{i=1}^n(1-\lambda_i) \geq 1-\sum_{i=1}^n \lambda_i$ whenever $0 \leq \lambda_i \leq 1$. Since $A(1-\lambda) = A - A\lambda \geq A-\lambda \geq B-\lambda$ whenever $B \leq A \leq 1$, the lemma follows by induction on n (with $A = \prod_{i=1}^{n-1}(1-\lambda_i)$, $B = 1-\sum_{i=1}^{n-1}\lambda_i$, and $\lambda = \lambda_n$). ■

Lemma III.2. *Suppose that* $0 < r_0 < 1$ *and*

$$(1-r_0)^{-\frac{1}{2}}e^{p_0}(p-p_0) - e^{q_0}(q-q_0) \leq \delta \qquad (5)$$

$$r \leq r_0. \qquad (6)$$

Then

$$(1-r)^{-\frac{1}{2}}e^{p} - e^{q} \leq Me^{\frac{\delta}{M}}, \qquad (7)$$

with $M = (1-r_0)^{-\frac{1}{2}}e^{p_0} - e^{q_0}$.

Proof: Since the left-hand side of (7) is increasing with r, by condition (6) it suffices to consider the case $r = r_0$. We can then maximize $(1-r_0)^{-\frac{1}{2}}e^p - e^q$ against (5) using Lagrange multipliers, and obtain a unique maximum at $p = p_0 + \frac{\delta}{M}$, $q = q_0 + \frac{\delta}{M}$ as long as $0 < r_0 < 1$. Substituting back in yields (7). ■

Setting $p_0 = q_0 = 0$ and letting b equal $1-(1-r_0)^{\frac{1}{2}}$, Lemmas III.1 and III.2 imply that we can set $C(n)$ to $(1-b)^{-1}-1$ as long as $(1-b)^{-1}f^TS(n)(S(n)-S(n)gg^TS(n))^{-1}S(n)f \leq x^TS(n-1)x$ and $\mathrm{Tr}(g^TS(n)g) \leq 2b-b^2$. We handle this last part by introducing a Lagrange multiplier, so that we end up with the three constraints

$$\begin{aligned}(1-b)x^TS(n-1)x + \lambda(x)(x^TS(n-1)x-\rho) \qquad (8)\\ - f^TS(n)(S(n)-S(n)gg^TS(n))^{-1}S(n)f \geq 0\end{aligned}$$

$$\begin{bmatrix} 1 & b \\ b & 2b - \mathrm{Tr}(g(x,n-1)^TS(n)g(x,n-1)) \end{bmatrix} \succeq 0 \qquad (9)$$

$$\lambda(x) \geq 0. \qquad (10)$$

for all x. Note that (9) is equivalent to $\mathrm{Tr}(g^TS(n)g) \leq 2b-b^2$ by Schur complements.

Remark As the noise goes to 0, we can set b to 0. It is easy to check that the constraints then reduce to the Lyapunov equation $f(x)^TS(t)f(x) \leq x^TS(t-1)x$ with a Lagrange multiplier added to check regional stability.

B. Continuous-Time

We now turn to the continuous-time case. Recall that we are interested in the infinitesimal operator $\mathcal{A}B(x,t)$ defined in Equation 1. For systems of the form $dx(t) = f(x)dt + g(x)dw(t)$, we can compute [28]

$$\mathcal{A}B(x,t) = \frac{\partial B}{\partial t} + \frac{\partial B}{\partial x}f(x) + \frac{1}{2}\mathrm{Tr}\left(g(x)^T\frac{\partial^2 B}{\partial x^2}g(x)\right). \qquad (11)$$

For functions of the form $B_S(x) = e^{\frac{1}{2}x^TS(t)x}$, (11) becomes

$$\begin{aligned}\mathcal{A}B_S(x,t) &= e^{\frac{1}{2}x^TSx}\\ &\times\left[\frac{1}{2}x^T\dot{S}x + x^TSf + \frac{1}{2}\mathrm{Tr}\left(g^TSg\right) + \frac{1}{2}x^TSgg^TSx\right].\end{aligned}$$

Then Theorem II.2 implies that the failure probability is bounded by

$$\frac{e^{\frac{1}{2}x(0)^TS(0)x(0)} - 1 + \int_0^T C(t)dt}{e^{\frac{1}{2}\rho}-1} \qquad (12)$$

as long as (i) $x^TSx \geq \rho$ for all $x \notin R_t$ and (ii) $C(t) \geq e^{\frac{1}{2}x^TSx}\left[\frac{1}{2}x^T\dot{S}x + x^TSf + \frac{1}{2}\mathrm{Tr}(g^TSg) + \frac{1}{2}x^TSgg^TSx\right]$ whenever $x^TSx < \rho$. We would therefore like an analog of Lemma III.2 for functions of the form $p(x)e^{q(x)}$. The following will suffice:

Lemma III.3. *Suppose that* $p(x) \leq p_0(1+q_0-q(x))$ *and* $p_0 \geq 0$. *Then* $p(x)e^{q(x)} \leq p_0e^{q_0}$.

Proof: Since $1-x \leq e^{-x}$, $1+q_0-q(x) \leq e^{q_0-q(x)}$, so $p(x) \leq p_0(1+q_0-q(x)) \leq p_0e^{q_0-q(x)}$. Multiplying both sides by $e^{q(x)}$ yields $p(x)e^{q(x)} \leq p_0e^{q_0}$. ■

Applying Lemma III.3 with $p_0 = b$ and $q_0 = 0$ allows us to upper-bound $\mathcal{A}B_S(x,t)$ by b as long as

$$\begin{aligned}b(2-x^TSx) - 2x^TSf - \mathrm{Tr}(g^TSg) - x^T\dot{S}x \qquad (13)\\ - x^TSgg^TSx + 2\lambda(x,t)(x^TSx-\rho) \geq 0\end{aligned}$$

for some non-negative function $\lambda(x,t)$.

IV. Optimizing over S

Assuming that we can find suitable values of S, b, λ, and ρ, conditions (8-10) and (13) allow us to upper-bound the failure probability. However, while checking these constraints for a fixed tuple (S,b,λ,ρ) is a sum-of-squares program, optimizing against them is not, because the decision variables S, b, λ, and ρ appear nonlinearly.

We now describe one approach for finding good values of these variables. Our approach is to first find values of S and b (S_0 and b_0, say) that work for the system linearized about some fixed point. We then restrict our consideration to multiples cS_0 of S_0, and binary search over c for several values of ρ, setting b equal to $(1-\mathrm{Tr}(g^TSg))^{-\frac{1}{2}}$ in the DT case and $\frac{1}{2}\mathrm{Tr}(g^TSg)$ in the CT case; these values are optimal for satisfying (8) and (13).

This approach will work well for time-invariant systems; it is similar to using the S matrix from LQR as a local Lyapunov function for a deterministic system. However, it will not work well for time-varying systems because it introduces extra conservatism that compounds over the course of a trajectory. Another approach based on alternating maximization between (S,ρ) and (b,λ) will work better for trajectories, but it is more difficult to implement and more often runs into numerical issues; we intend to describe this approach in a later paper.

For the linearization-based approach, we still need a way to find good values of S and b for the linearized system. We consider the DT constraints first. If we apply Schur complements to (8) and linearize, we obtain the matrix constraint

$$\begin{bmatrix} S_0 - S_0g(0)g(0)^TS_0 & S_0F \\ (S_0F)^T & (1-b)S_0 \end{bmatrix} \succeq 0, \qquad (14)$$

where $F = \nabla f(0)$. If we introduce a dummy variable P and require that $S_0 - S_0 g(0) g(0)^T S_0 \succeq P$, then we can re-write (14) as the pair of constraints

$$\left[\begin{array}{cc} P & S_0 F \\ (S_0 F)^T & (1-b) S_0 \end{array} \right] \tag{15}$$

$$\left[\begin{array}{cc} I & (S_0 g(0))^T \\ S_0 g(0) & S_0 - P \end{array} \right] \succeq 0. \tag{16}$$

The CT case can be dealt with similarly, leaving us with

$$\left[\begin{array}{cc} I & (S_0 g(0))^T \\ S_0 g(0) & -bS_0 - S_0 F - (S_0 F)^T \end{array} \right] \succeq 0. \tag{17}$$

In both cases, once we fix b we are left with a semi-definite constraint, so we can just perform a line search on b and then solve a semidefinite program over S_0. However, we need a good objective to optimize against. We choose to maximize α such that $S_0 \succeq \alpha M$, for some well-chosen matrix M (this is equivalent to requiring that $x^T S_0 x \geq \alpha$ whenever $x^T M x \geq 1$). There are two reasons to use this objective. First, if $x^T M x$ gives an indication of how nonlinear the system is at x, then we want $x^T S_0 x$ to be large whenever $x^T M x$ is large; this makes it more likely that S_0 will work well for the nonlinear system. Second, if M defines some safety constraint (i.e. the system is safe if $x^T M x < 1$), then we would also like S_0 to be large relative to M in order to minimize the failure probability.

V. Examples

Now that we have covered the theoretical underpinnings of our method, we will demonstrate its effectiveness with several examples. For each example, we first describe the system, then indicate which M matrix we used (see Section IV), the values of S_0 and b_0, the values of c and ρ, and the final probability bound.

A. Example 1: Simple Pendulum, Discrete Time

Our first example is a pendulum stabilized about the upright with a time step of $\Delta t = 0.01$. We use the following equations for the pendulum dynamics (the $\sin$ term has been Taylor expanded to third order):

$$\left[\begin{array}{c} \theta_{n+1} \\ \dot{\theta}_{n+1} \end{array} \right] = \left[\begin{array}{c} \theta_n + 0.01\dot{\theta}_n \\ -0.0167\theta_n^3 - 0.3\theta_n + 0.97\dot{\theta}_n \end{array} \right] + \left[\begin{array}{c} 0.01 w_{1,n} \\ 0.05 w_{2,n} \end{array} \right]$$

We want to bound the probability that θ leaves the region $\left(-\frac{\pi}{6}, \frac{\pi}{6}\right)$ after 3600 seconds. We thus set M to $\left[\begin{array}{cc} \left(\frac{6}{\pi}\right)^2 & 0 \\ 0 & 0 \end{array} \right]$, as then $x^T M x > 1 \iff |\theta| > \frac{\pi}{6}$.

For $b_0 = 0.0136$, we get $S_0 = \left[\begin{array}{cc} 142.71 & 7.49 \\ 7.49 & 5.09 \end{array} \right]$ with $\alpha = 36.10$. When we verify on the nonlinear system, we get $c = 0.955$, $\rho = 34.48$ (ρ is equal to $c\alpha$ because the constraint $S \succeq \rho M$ was the first to become tight). Figure 1 shows the log of the failure probability plotted against initial conditions.

Note that we get strong bounds (failure probabilities less than 10^{-3}) for a large region around the origin. For the sake of comparison, we estimated the actual failure probability using a Kalman filter for the linearized system, also included in Figure 1. While the true probabilities are much smaller than verified (10^{-10} vs. 10^{-3}), the verified region of stability is not much smaller than the actual region of stability. For most robotics applications we are more interested in the operating region where we have a high success probability than in how small the failure probability is for zero initial conditions. In this respect our verification method is close to the true answer.

B. Example 2: Simple Pendulum, Continuous Time

We perform the same optimization as before, checking against the continuous-time version of the constraints. For $b_0 = 1.51$, we get $S_0 = \left[\begin{array}{cc} 156.90 & 8.34 \\ 8.34 & 5.64 \end{array} \right]$ with a corresponding α value of 39.63. When we verify on the nonlinear system, we get $c = 1.0$, $\rho = 39.63$. The failure probability is plotted in Figure 2.

C. Example 3: Cart-Pole Balancing, Continuous Time

The next example demonstrates that our approach is scalable to more complicated systems. It is also an example of including observation noise in the model. The cart and pole system is a pendulum with length L and mass m_p attached to a cart with mass m_c. The system is actuated by a force u on the center of mass of the cart. Letting $\theta = 0$ when the pendulum is pointing straight up, the equations of motion are

$$\ddot{x} = \frac{u - m_p \sin(\theta)(L\dot{\theta}^2 - g\cos(\theta))}{m_c + m_p \sin(\theta)^2},$$

$$\ddot{\theta} = \frac{u\cos(\theta) - m_p L\dot{\theta}^2 \cos(\theta)\sin(\theta) + (m_c + m_p) g \sin(\theta)}{L(m_c + m_p \sin(\theta)^2)}.$$

We set $m_p = 1.0$, $m_c = 10.0$, $L = 0.5$, $g = 9.8$, and take a third-order Taylor expansion to get the following dynamics:

$$\left[\begin{array}{c} dx \\ d\theta \\ d\dot{x} \\ d\dot{\theta} \end{array} \right] = \left[\begin{array}{c} \dot{x} \\ \dot{\theta} \\ -.75\theta^3 - .01\theta^2 u - .05\theta\dot{\theta}^2 + .98\theta + .1u \\ -5.75\theta^3 - .12\theta^2 u - .1\theta\dot{\theta}^2 + 21.56\theta + .2u \end{array} \right] dt$$
$$+ \operatorname{diag}\left(\left[\begin{array}{cccc} 0.03 & 0.03 & 0.1 & 0.1 \end{array} \right]\right) dw(t).$$

To stabilize this system, we apply LQR control to the linear system with cost matrices $Q = \operatorname{diag}([10, 10, 1, 1])$, $R = 0.1$ to get a gain matrix of $K = \left[\begin{array}{cccc} -10.0 & 289.83 & -19.53 & 63.25 \end{array} \right]$.

Let us suppose that we also have independent measurement noise on x, θ, $\dot{x}$, and $\dot{\theta}$, with standard deviations of 0.01, 0.01, 0.03, and 0.03, respectively. Our feedback law will push this noise back into the dynamics, adding 4 extra noise channels that end up being functions of θ.

Because the major source of nonlinearity is θ, want $x^T S_0 x$ to grow quickly with θ. We will therefore set M to $\operatorname{diag}\left(\left[\begin{array}{cccc} 0 & 1 & 0 & 0 \end{array} \right]\right)$. For $b_0 = 0.728$, we get $S_0 = \left[\begin{array}{cccc} 2.45 & -12.01 & 2.57 & -1.85 \\ -12.01 & 231.42 & -21.72 & 22.72 \\ 2.57 & -21.72 & 5.95 & -4.46 \\ -1.85 & 22.72 & -4.46 & 5.26 \end{array} \right]$, with $\alpha = 124.36$. When we verify on the nonlinear system, we get $c = 0.9023$,

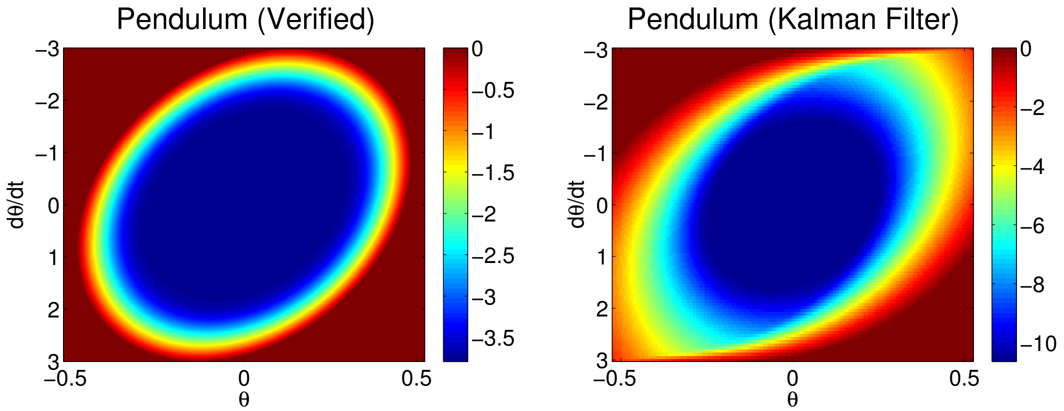

Fig. 1: The log-base-10 of the failure probability for the discrete-time pendulum after one hour. Left: failure probability plotted against initial conditions, verified with our algorithm. Right: estimated failure probability for the linearized discrete-time pendulum, computed with a Kalman filter.

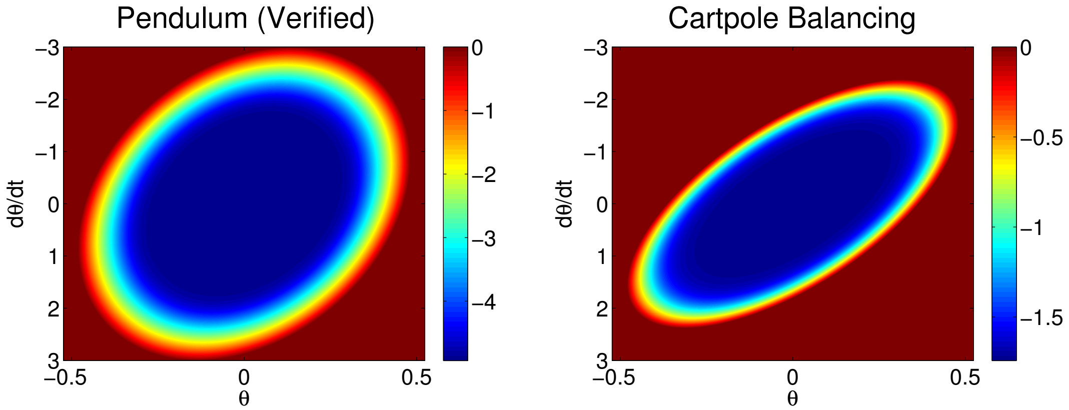

Fig. 2: The log-base-10 failure probabilities for two continuous-time systems. Left: balancing for the pendulum, as a function of initial conditions. Right: balancing for the cartpole, as a function of initial conditions in x and θ (the initial conditions for $\dot{x}$ and $\dot{\theta}$ are fixed to 0).

$\rho = 20.75$. Figure 2 contains a visualization of the failure probability after one hour.

D. Example 4: Rimless Wheel

The rimless wheel is a common model for walking first introduced in [14]. It is a wheel consisting of n_s spokes, each of length L, connected at a point. The angle between consecutive spokes is $\theta = \frac{2\pi}{n_s}$. The spokes are massless; the central point has a mass of M. The rimless wheel typically rolls down a hill, say with slope angle γ. When a spoke impacts the ground, the collision is inelastic, conserves angular momentum, and immediately transfers support to the next spoke. Because of the impacts, the rimless wheel is an inherently discrete-time system. One way to compute its dynamics across several collisions is via the *Poincaré return map*, which gives the angular velocity at the point where the stance leg is vertical. If we let ω_n denote this angular velocity between the nth and $(n+1)$st impacts, and let $x_n = \omega_n^2$, then [4]

$$x_{n+1} = \cos^2(\theta)\left(x_n + \frac{2g}{L}(1-\cos\beta_1)\right) - \frac{2g}{L}(1-\cos\beta_2),$$

where $\beta_1 = \frac{\theta}{2} + \gamma$ and $\beta_2 = \frac{\theta}{2} - \gamma$. As in [4], we model γ as Gaussian with mean $\gamma_0 = 8^\circ$ and standard deviation $\sigma = 1.5^\circ$. This means that the actual noise to the system is non-Gaussian since it is filtered through a cosine. The system is locally stable to some value $\bar{x} > 0$ as well as to the state where both stance legs are on the ground and the wheel stops moving. We will consider this second stable point a failure state, which corresponds to $x_n \leq 0$.

We will compare the following approaches to bounding the time until the wheel enters this failure state:

1) Find the smallest slope γ_s such that the rimless wheel would roll forever with a constant slope of γ_s. Then compute the probability that $\gamma < \gamma_s$. The expected time to failure is at least the reciprocal of this probability.

One-step slope bound (nonlinear)	313 impacts
One-step slope bound (linear)	428 impacts
Noise as state variable	50 impacts
Linearized noise	12647 impacts
Discrete-state	643600 impacts

TABLE I: Expected failure time/50% failure probability thresholds for the rimless wheel. The first, second, and last bounds compute expected failure times, while the second and third bounds compute the time with a 50% failure probability.

2) Let v_n denote $\gamma - \gamma_0$ for time $n+1$. Then v_n is Gaussian, and it is okay that it affects the dynamics in a nonlinear way because it is a state variable. We can then apply the techniques of this paper to find a time that has at most a 50% probability of failure.
3) Approximate the noise as an appropriate Gaussian by linearizing around γ_0, then apply the techniques of this paper.
4) Discretize the state space and compute the expected time to failure exactly (up to the discretization) by solving a system of equations, as in [4].

In order to make the point of stability the origin, we make the change of coordinates $x \mapsto x - \bar{x}$.

In the first approach, solving for γ_s yields 3.91° in the nonlinear noise case and 3.76° in the linearized case. The respective bounds on expected time to failure are 313.08 and 427.74 impacts, respectively.

In the second approach, we set M to $\begin{bmatrix} \frac{1}{\bar{x}^2} & 0 \\ 0 & 0 \end{bmatrix}$. On the nonlinear system, we obtain $c = 0.972$, $\rho = 7.45$, leading to a bound of $0.4057T$ for initial conditions at the origin. We thus hit 50% failure at $T = \frac{40.49}{2 \times 0.4057} = 49.90$ impacts. This compares poorly to the first approach, which may imply that dealing with non-Gaussian noise by filtering it through nonlinear dynamics does not work well in practice.

In the third approach, we set M to $\frac{1}{\bar{x}^2}$. We get $c = 1$, $\rho = 19.19$, and a 50% failure rate at $T = 12646.90$ impacts, a significant improvement on both of the first two approaches.

Finally, as computed in [4], the actual expected failure time is 643600. These results are summarized in Table I.

E. Example 5: Room Heating

Our final example evaluates the scalability of our approach. We compare our algorithm to the algorithm presented in Abate et al [1]. The experiment presented in [1] concerns bounding the probability that a heating system allows any of h rooms to leave given temperature ranges. For a heating system with h rooms, we represent the temperature of the h rooms as a vector $x = (x_1, x_2, \dots, x_h)$, and consider the discrete-time system $x_{n+1} = f(x_n) + g(x_n)w_n$ with

$$f(x)_i = \quad x_i + b(x_0 - x_i) + a\left(\sum_{j \neq i} x_j - x_i\right) + c\sigma\left(\frac{x_i}{\alpha} - 1\right) \tag{18}$$

$$g(x) = \quad \nu I_{h \times h}, \tag{19}$$

where σ is a sigmoidal function rising from 0 to 1, which we approximated as $\sigma(y) = 0.5 - 2.5y + 1.25y^2 + 20y^3$. For our experiment we took $a = 0.0625$, $b = 0.025$, $c = 0.6$, $x_0 = 6.0$, $\alpha = 19.5$, and $\nu = 0.25$. The goal was to bound the probability of leaving the temperature region defined by $[17, 22] \times [16, 23]^{h-1}$. The numbers given above are based on Abate et al.'s paper, although we make a few simplifying assumptions to the dynamics — first, we replace a certain Bernoulli noise source by its expectation; second, we assume symmetric between-room interactions so that there will be an easily identifiable fixed point about which to verify stability. We also remove a one-step lag on noise, which increases the discretization mesh of [1] by a factor of 2 per dimension.

We observe that Abate et al. are able to (using 5 bins per dimension) verify a 5-room heating system in 11 hours on a 3.4GHz PC with 1GB of RAM. Because of the factor of 2 per dimension that they incur, a fair comparison of runtime would be to test our SOS verification on a 7-room heating system (the mesh size in [1] would decrease by a factor of 32 by ignoring lag, then gain a factor of 25 when going from 5 to 7 dimensions, so that their 7-room times without lags would be 6-7 hours, as their runtime scales about quadratically with mesh size).

In this case a single SOS verification runs in an average of 17.2 seconds (our algorithm performs several such verifications). We used a 3.4GHz PC with 24GB of RAM; we note that our PC had 12 cores, with CPU diagnostics indicating that only 4 cores were actually utilized by our computation. We furthermore note that for a fixed degree of Taylor approximation our method scales polynomially with dimension, whereas discretization methods scale exponentially with dimension. Our method is therefore not only more scalable currently, it will also continue to scale well with increased computing power.

VI. Conclusion

We have presented a method for verifying stochastic nonlinear systems. However, the results here are by no means a complete theory; there is much work left to be done. Our hope is that the successful examples in this paper will convince others that the methods first presented in [19] can extend usefully to complex systems for suitable choices of barrier functions. We chose exponentials of quadratic barrier functions because the systems we had in mind were locally well-approximated by linear systems and the noise model was Gaussian. Other applications will require different families of barrier functions; hopefully the convex relaxations given in (III.2) and (III.3) will provide inspiration for similar relaxations for those other families. It seems that usually one can obtain such relaxations from simple analytical properties of the expressions in question, but the authors do not yet have a way to make this observation rigorous.

Some interesting modifications to the dynamics would be to consider mixtures of Gaussians, as well as switching processes, in the noise model; also to consider verification about stabilized trajectories. A final case of interest is Gaussian noise passed through a nonlinear filter; as discussed in the Rimless Wheel section, our method handles this case in principal, but performs poorly in practice.

References

[1] A. Abate, J.-P. Katoen, J. Lygeros, and M. Prandini. Approximate model checking of stochastic hybrid systems. *European Journal of Control*, 16:624641, dec 2010.

[2] Pieter Abbeel, Adam Coates, Morgan Quigley, and Andrew Y. Ng. An application of reinforcement learning to aerobatic helicopter flight. In *Proceedings of the Neural Information Processing Systems (NIPS '07)*, volume 19, December 2006.

[3] Frederick J. Beutler. On two discrete-time system stability concepts and supermartingales. *Journal of Mathematical Analysis and Applications*, 44(2):464 – 471, 1973.

[4] Katie Byl and Russ Tedrake. Metastable walking machines. *International Journal of Robotics Research*, 28 (8):1040–1064, August 1 2009.

[5] A. Charnes and W. W. Cooper. Chance-constrained programming. *Management Science*, 6(1):pp. 73–79, 1959.

[6] Rick Cory and Russ Tedrake. Experiments in fixed-wing UAV perching. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*. AIAA, 2008.

[7] E. B. Dynkin. *Markov Processes*, volume 1. Academic Press, 1965.

[8] M Grant and S Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control (tribute to M. Vidyasagar), Lecture Notes in Control and Information Sciences*, pages 95–110. Springer, 2008.

[9] M Grant and S Boyd. CVX: Matlab software for disciplined convex programming - version 1.1beta. http://cvxr.com/cvx, January 2011.

[10] Matthew R. James. Asymptotic analysis of nonlinear stochastic risk-sensitive control and differential games. *Mathematics of Control, Signals, and Systems (MCSS)*, 5:401–417, 1992. 10.1007/BF02134013.

[11] H. J. Kushner. On the stability of stochastic dynamical systems. *PNAS*, 53(1):8–12, Jan. 15 1965.

[12] HJ Kushner. Finite time stochastic stability and the analysis of tracking systems. *IEEE Transactions on Automatic Control*, pages 219–227, April 1966.

[13] Johan Lofberg. Pre- and post-processing sum-of-squares programs in practice. *IEEE Transactions On Automatic Control*, 54(5):1007–, May 2009.

[14] Tad McGeer. Passive dynamic walking. *International Journal of Robotics Research*, 9(2):62–82, April 1990.

[15] A. Megretski. Positivity of trigonometric polynomials. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 4, pages 3814–3817 vol.4, Dec. 2003.

[16] D. Mellinger, N. Michael, and V. Kumar. Trajectory generation and control for precise aggressive maneuvers with quadrotors. In *Proceedings of the 12th International Symposium on Experimental Robotics (ISER 2010)*, 2010.

[17] Antonis Papachristodoulou and Stephen Prajna. Analysis of non-polynomial systems using the sum of squares decomposition. *Positive Polynomials in Control*, 312/2005: 23–43, 2005.

[18] Quang-Cuong Pham, Tabareau, N., Slotine, and J.-J. A contraction theory approach to stochastic incremental stability. *Automatic Control, IEEE Transactions on*, 54 (4):816 –820, Apr 2009.

[19] S Prajna, A. Jadbabaie, and GJ Pappas. Stochastic safety verification using barrier certificates. *43rd IEEE Conference on Decision and Control*, pages 929–934, 2004.

[20] Stephen Prajna, Antonis Papachristodoulou, Peter Seiler, and Pablo A. Parrilo. *SOSTOOLS: Sum of Squares Optimization Toolbox for MATLAB Users guide*, 2.00 edition, June 1 2004.

[21] Stephen Prajna, Antonis Papachristodoulou, and Fen Wu. Nonlinear control synthesis by sum of squares optimization: A Lyapunov-based approach. In *Proceedings of the ASCC 2004*, 2004.

[22] Marc Raibert, Kevin Blankespoor, Gabriel Nelson, Rob Playter, and the BigDog Team. Bigdog, the rough-terrain quadruped robot. *Proceedings of the 17th World Congress, The International Federation of Automatic Control*, 2008.

[23] Alexander Shkolnik, Michael Levashov, Ian R. Manchester, and Russ Tedrake. Bounding on rough terrain with the littledog robot. *Under review*, 2010.

[24] Jos F. Sturm. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11(1-4):625 – 653, 1999.

[25] W. Tan and A. Packard. Stability region analysis using polynomial and composite polynomial Lyapunov functions and sum-of-squares programming. *IEEE Transactions on Automatic Control*, 53(2):565–571, March 2008.

[26] Mark M. Tobenkin, Ian R. Manchester, and Russ Tedrake. Invariant funnels around trajectories using sum-of-squares programming. *arXiv:1010.3013 [math.DS]*, 2010.

[27] Eric R. Westervelt, Jessy W. Grizzle, Christine Chevallereau, Jun Ho Choi, and Benjamin Morris. *Feedback Control of Dynamic Bipedal Robot Locomotion*. CRC Press, Boca Raton, FL, 2007.

[28] Y Yang, J Li, and G Chen. Finite-time stability and stabilization of nonlinear stochastic hybrid systems. *Journal of Mathematical Analysis and Applications*, 356:338–345, 2009.