Probably Approximate Safety Verification of Hybrid Dynamical Systems

Bai Xue[1,2,3(✉)], Martin Fränzle[4], Hengjun Zhao[5], Naijun Zhan[1,2], and Arvind Easwaran[6]

[1] State Key Laboratory of Computer Science, Institute of Software, CAS, Beijing, China
{xuebai,znj}@ios.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Beijing Institute of Control Engineering, Beijing, China
[4] Carl von Ossietzky Universität Oldenburg, Oldenburg, Germany
fraenzle@informatik.uni-oldenburg.de
[5] Southwest University, Chongqing, China
zhaohj2016@swu.edu.cn
[6] Nanyang Technological University, Singapore, Singapore
arvinde@ntu.edu.sg

Abstract. In this paper we present a method based on linear programming that facilitates reliable safety verification of hybrid dynamical systems subject to perturbation inputs over the infinite time horizon. The verification algorithm applies the probably approximately correct (PAC) learning framework and consequently can be regarded as statistically formal verification in the sense that it provides formal safety guarantees expressed using error probabilities and confidences. The safety of hybrid systems in this framework is verified via the computation of so-called PAC barrier certificates, which can be computed by solving a linear programming problem. Based on scenario approaches, the linear program is constructed by a family of independent and identically distributed state samples. In this way we can conduct verification of hybrid dynamical systems that existing methods are not capable of dealing with. Some preliminary experiments demonstrate the performance of our approach.

Keywords: Hybrid systems · Probably approximately safe · Linear program

1 Introduction

The complexity of today's technological applications induces a quest for automation, leading to autonomous cyber-physical systems [9]. Many of these systems operate in safety-critical contexts and hence become safety-critical systems themselves. Being safety-critical, they have to reliably sustain safety despite perturbations. The propagation of these perturbations however tends to be highly nonlinear and combine continuous and discrete dynamics. Such combined dynamics yield a hybrid dynamical system involving interacting discrete-event and

Y. Ait-Ameur and S. Qin (Eds.): ICFEM 2019, LNCS 11852, pp. 236–252, 2019.
https://doi.org/10.1007/978-3-030-32409-4_15

continuous-variable dynamics. Hybrid dynamical systems are important in applications such as robotics, manufacturing systems and bio-molecular networks, and have been at the center of intense research activity in computer-aided verification, control theory, and applied mathematics [2].

The process of verifying with mathematical rigor that a hybrid dynamical system behaves correctly is a well-established branch of formal methods in computer science [1]. Unfortunately, many decision problems underlying formal verification of hybrid systems are undecidable [17]. Even surprisingly simple dynamical systems combining discrete and continuous dynamics feature undecidable state-reachability problems, like multi-priced timed automata with stopwatch prices [13] or three-dimensional piecewise constant derivative systems [3]. General undecidability renders sound yet incomplete automatic verification methods as well as methods providing a controlled approximation error attractive, e.g. [10,14,26], which nevertheless are computationally expensive. Although sophisticated heuristics have been developed to improve scalability of the techniques, automatic key-press formal verification of real-world systems is still considered to be impractical [30]. Techniques for simulation-based verification can prove fruitful in this regard for systems over finite time horizons, as they combine the scalability of simulation with rigorous coverage criteria supporting either a complete or a statistical verification through generalization from samples [23,38].

In this paper we propose a linear programming based method that facilitates reliable, in the sense of featuring a rigorously quantified confidence in the verification verdict, safety verification of hybrid systems subject to perturbations over the infinite time horizon. Akin to [11], the verification algorithm applies the framework of PAC learning theory [15] to adjust the effort invested in generating samples to a desired confidence in the verification verdict. Given a confidence $\beta \in (0,1)$, the objective is to compute a probability $\epsilon \in (0,1)$ such that the probability of initial continuous states leading to the satisfiability of safety properties is larger than $1-\epsilon$, with at least $1-\beta$ confidence. Such verification in our method is studied by learning a so-called PAC barrier certificate with respect to ϵ and β, which withat least $1-\beta$ confidence is indeed a barrier certificate with probability larger than $1-\epsilon$. The computation is based on scenario approaches [6] and linear interval inequalities [28], which encodes as a linear programming problem. The linear program is constructed using linear interval inequalities and a family of independent and identically distributed state samples extracted from the initial set. Based on the computed solution to this linear program, confidence level $\beta \in (0,1)$ and number of samples, we compute a probability measure ϵ based on scenario approaches such that the computed solution to the linear program forms a PAC barrier certificate with respect to ϵ and β. Consequently we conclude that the probability of initial continuous states leading to the satisfiability of safety properties is larger than $1-\epsilon$, with confidence higher than $1-\beta$. Some examples demonstrate the performance and merits of our approach.

2 Preliminaries

In this section we introduce hybrid systems, the safety verification problem, scenario approaches and linear interval inequalities. The following notations are used throughout this paper: $\mathcal{C}^1(\mathbb{R}^n)$ is the set of continuously differentiable functions from $\mathbb{R}^n$ to $\mathbb{R}$. $\mathbb{R}_{\geq 0}$ denotes the set of nonnegative real values and $\mathbb{R}_{>0}$ denotes the set of positive reals. Vectors are denoted by boldface letters.

2.1 Hybrid Systems

A hybrid system is a tuple $H = (\mathcal{X}, L, X, X_0, \texttt{Inv}, \boldsymbol{F}, T)$ [24]:

- $\mathcal{X} \subseteq \mathbb{R}^n$ is the continuous state space;
- L is a finite set of locations and we will in the sequel denote its cardinality by $M = |L|$ with $L = \{1, \ldots, M\}$;
- The overall state space of the system is $X = L \times \mathcal{X}$, and a state of the system is denoted by $(l, \boldsymbol{x}) \in L \times \mathcal{X}$;
- $X_0 \subseteq X$ is the set of initial states;
- $\texttt{Inv} : L \to 2^{\mathcal{X}}$ is the invariant, which assigns to each location l a set $\texttt{Inv}(l) \subseteq \mathcal{X}$ that contains all possible continuous states while at location l;
- $\boldsymbol{F} : X \to 2^{\mathbb{R}^n}$ is a set of vector fields. $\boldsymbol{F}$ assigns to each $(l, \boldsymbol{x}) \in X$ a set $\boldsymbol{F}(l, \boldsymbol{x}) \subseteq \mathbb{R}^n$ which constrains the evolution of the continuous state according to the differential inclusion $\dot{\boldsymbol{x}} \in \boldsymbol{F}(l, \boldsymbol{x})$;
- $T \subseteq X \times X$ is a relation capturing discrete transitions between two locations. Here a transition $((l', \boldsymbol{x}'), (l, \boldsymbol{x})) \in T$ indicates that from the state $(l', \boldsymbol{x}')$ the system can undergo a discrete jump to the state $(l, \boldsymbol{x})$.

We assume that the uncertainty in the continuous flow is caused by some perturbation inputs in the manner: $\boldsymbol{F}(l, \boldsymbol{x}) = \{\dot{\boldsymbol{x}} \in \mathbb{R}^n \mid \dot{\boldsymbol{x}} = \boldsymbol{f}_l(\boldsymbol{x}, \boldsymbol{d}),$ for some $\boldsymbol{d} \in D(l)\}$, where $\boldsymbol{f}_l(\boldsymbol{x}, \boldsymbol{d})$ is a vector field that governs the flow of the system at location l, and $\boldsymbol{d}$ is a vector of perturbation inputs that takes value in $D(l) \subset \mathbb{R}^r$.

Trajectories of the hybrid system H starting from some initial state $(l_0, \boldsymbol{x}_0) \in X_0$ are concatenations of steps, with each step either being a continuous flow or a discrete transition, with the endpoint of each step matching the startpoint of the next step, and with the first step starting in $(l_0, \boldsymbol{x}_0) \in X_0$. During a continuous flow, the discrete location l is maintained and the continuous state evolves according to the differential inclusion $\dot{\boldsymbol{x}} \in \boldsymbol{F}(l, \boldsymbol{x})$, as long as $\boldsymbol{x}$ remains inside the invariant set $\texttt{Inv}(l)$. At a state $(l_1, \boldsymbol{x}_1)$ a discrete transition to $(l_2, \boldsymbol{x}_2)$ can occur iff $((l_1, \boldsymbol{x}_1), (l_2, \boldsymbol{x}_2)) \in T$. We then say that $\boldsymbol{x}_1 \in \texttt{G}_{l_1,l_2} = \{\boldsymbol{x}_1 \in \mathcal{X} \mid ((l_1, \boldsymbol{x}_1), (l_2, \boldsymbol{x})) \in T$ for some $\boldsymbol{x} \in \mathcal{X}\}$ and $\boldsymbol{x}_2 \in \texttt{R}_{l_1,l_2}(\boldsymbol{x}_1)$, where $\texttt{R}_{l_1,l_2} : \boldsymbol{x}_1 \to \{\boldsymbol{x} \in \mathcal{X} \mid ((l_1, \boldsymbol{x}_1), (l_2, \boldsymbol{x})) \in T\}$. If $\texttt{G}_{l',l}$ is empty then no discrete transition from location l' to location l is possible and the associated reset map undefined. Although not explicitly stated, it is assumed that the description of the hybrid system given above is well-posed. For example, $(l, \boldsymbol{x}) \in X_0$ automatically implies that $\boldsymbol{x} \in \texttt{Inv}(l)$, and $((l', \boldsymbol{x}'), (l, \boldsymbol{x})) \in T$ implies that $\boldsymbol{x}' \in \texttt{Inv}(l')$ and $\boldsymbol{x} \in \texttt{Inv}(l)$.

Given a system H and a set of unsafe states $X_u \subseteq X$, the classical *safety verification problem* is concerned with proving that no trajectory of the hybrid

system H originating from the set X_0 of initial states can ever enter the unsafe region X_u. If this property holds, the hybrid system H is safe. Unfortunately, such safety verification problem is undecidable generally and consequently is challenging, even for systems with simple dynamics. In this paper we relax the notion of safety, replacing qualitative safety (no trajectory may ever reach an unsafe state) by quantitative safety (the probability of unsafe behaviors stays below a quantitative safety target with some specified confidence). We call a system satisfying the latter property *probably approximately safe*. Its concept is formally introduced in Definition 1. The probably approximate safety verification applies the PAC learning framework [15] and consequently can be regarded as statistically formal verification in the sense that it provides formal safety guarantees expressed using error probabilities and confidence.

Suppose $\texttt{Ini}(l) = \{\boldsymbol{x} \mid (l, \boldsymbol{x}) \in X_0\}$ is endowed with a $\sigma-$algebra $\mathcal{D}_l$ and that a probability $\texttt{Pr}_l$ over $\texttt{Ini}(l)$ is assigned, where $l \in L$. In addition, we assume $\texttt{Ini} = \texttt{Ini}(1) \times \ldots \times \texttt{Ini}(M)$ is endowed with a $\sigma-$algebra $\mathcal{D}'$ and that a probability $\texttt{Pr}$ over $\mathcal{D}'$ is assigned. Obviously, $\texttt{Pr} = \texttt{Pr}_1 \times \ldots \times \texttt{Pr}_M$.

Definition 1. *A hybrid system H is probably approximately safe with respect to the set $\texttt{Ini}$, $\epsilon \in (0,1)$ and $\beta \in (0,1)$ (or, $\texttt{PAS}(\epsilon, \beta)$) if with at least $1-\beta$ confidence, $\texttt{Pr}(C) \geq 1-\epsilon$, where $C = \texttt{Ini}'(1) \times \ldots \times \texttt{Ini}'(M)$ is a subset of the set $\texttt{Ini}$ and $\texttt{Ini}'(l) \subseteq \texttt{Ini}(l)$ is a set of continuous states $\boldsymbol{x}$s in the location $l \in L$ such that trajectories of H starting from $(l, \boldsymbol{x})$ never enter the unsafe region X_u.*

Besides, we in this paper restrict the invariant set $\texttt{Inv}(l)$, disturbance set $D(l)$, unsafe set $\texttt{Uns}(l)$, guard set $\texttt{G}_{l',l}$ and initial set $\texttt{Ini}(l)$ to the interval form for $l \in L$, where $\texttt{Uns}(l) = \{\boldsymbol{x} \mid (l, \boldsymbol{x}) \in X_u\}$. The probability distribution $\texttt{Pr}_l$ is assumed to be uniform distribution over $\texttt{Ini}(l)$ for $l \in L$. We need to point out here that our method is not limited to this particular probability distribution. This feature is reflected in scenario approaches, which will be introduced in Subsect. 2.2. To some extent, the assumption of uniform distribution over $\texttt{Ini}(l)$ for $l \in L$ is reasonable since every continuous state in $\texttt{Ini}(l)$ is of equal importance especially for safety-critical systems. Any state leading to a violation of the safety property will result in a systems failure. Ideally, we wish that the hybrid system is safe for every initial state. As mentioned before, this is challenging to verify with mathematical rigor. Inspired by machine learning theory, we attempt to use a family of random finite states in $\texttt{Ini}(l)$ to learn safety information of hybrid systems in the PAC framework and would expect to verify systems that existing verification methods are not capable of dealing with.

2.2 Scenario Approaches

The scenario optimization has been shown as an intuitive and effective way to deal with chance-constrained optimization [4,5] based on finite randomization of the constraints at the expense of giving probabilistic guarantees on the robustness of the solution. Concretely, consider the chance-constrained optimization:

$$\begin{aligned} &\min_{\boldsymbol{x}\in\mathbb{R}^m} J(\boldsymbol{x}) \\ &\text{s.t. } P\big(\{\boldsymbol{\delta}\in\Delta \mid \max_{j=1,\ldots,n_m} g_j(\boldsymbol{x},\boldsymbol{\delta})\le 0\}\big)\ge 1-\epsilon, \end{aligned} \tag{1}$$

where $\boldsymbol{\delta}\in\Delta\subseteq\mathbb{R}^r$, $J:\mathbb{R}^m\to\mathbb{R}$ is a convex function and $g_j:\mathbb{R}^m\times\mathbb{R}^r\to\mathbb{R}$ for $j=1,\ldots,n_m$. Besides, $\{\boldsymbol{x}\in\mathbb{R}^m \mid \max_{j=1,\ldots,n_m} g_j(\boldsymbol{x},\boldsymbol{\delta})\le 0\}$ is convex and closed for fixed $\boldsymbol{\delta}$. Any $\boldsymbol{x}$ satisfying the chance constraint of (1) is referred to as an $\epsilon-$level feasible solution. It is assumed that Δ is endowed with a $\sigma-$algebra $\mathcal{D}$ and that P is a probability measure defined over $\mathcal{D}$.

The scenario approach substitutes the chance constraint in (1) with a finite number of hard constraints, each corresponding to a different realization $\boldsymbol{\delta}^{(k)}$, $k=1,\ldots,N$ of the uncertain parameter $\boldsymbol{\delta}$, extracted independently according to the probability distribution P. This leads to the convex optimization:

$$\begin{aligned} &\min_{\boldsymbol{x}\in\mathbb{R}^m} J(\boldsymbol{x}) \\ &\texttt{s.t.} \max_{j=1,\ldots,n_m} g_j(\boldsymbol{x},\boldsymbol{\delta}^{(i)})\le 0, i=1,\ldots,N. \end{aligned} \tag{2}$$

Assumption 1. *The convex optimization* (2) *is feasible for all possible multi-sample extractions* $(\boldsymbol{\delta}^{(1)},\ldots,\boldsymbol{\delta}^{(N)})\in\Delta^N$ *and its feasibility region has a non-empty interior. Moreover, the solution* $\boldsymbol{x}^*$ *of* (2) *exists and is unique.*

One can allow for violating part of the sample constraints to improve the optimal value by removing some sample constraints. Any removal algorithm $\mathcal{A}$ can be used when removing constraints in (2) [4]. The randomized program (2) where k constraints are removed by $\mathcal{A}$ is expressed as

$$\begin{aligned} &\min_{\boldsymbol{x}\in\mathbb{R}^m} J(\boldsymbol{x}) \\ &\texttt{s.t.} \max_{j=1,\ldots,n_m} g_j(\boldsymbol{x},\boldsymbol{\delta}^{(i)})\le 0, i\in\{1,\ldots,N\}\setminus\mathcal{A}(\boldsymbol{\delta}^{(1)},\ldots,\boldsymbol{\delta}^{(N)}) \end{aligned} \tag{3}$$

and its solution is indicated as $\boldsymbol{x}^{**}$. We assume the following:

Assumption 2. $\boldsymbol{x}^{**}$ *almost surely violates all the* k *removed constraints.*

Theorem 1 [4,5]**.** *Let* $\beta\in(0,1)$ *be any small confidence value. If* N *and* k *are such that* $\binom{k+m-1}{k}\sum_{i=0}^{k+m-1}\binom{N}{i}\epsilon^i(1-\epsilon)^{N-i}\le\beta$, *where* m *is the number of optimization variables, then with probability at least* $1-\beta$, *we have that* $P\big(\{\boldsymbol{\delta}\in\Delta \mid \max_{j=1,\ldots,n_m} g_j(\boldsymbol{x}^{**},\boldsymbol{\delta})\le 0\}\big)\ge 1-\epsilon$.

In Theorem 1, $1-\beta$ is the N-fold probability P^N in $\Delta^N=\Delta\times\Delta\times\cdots\times\Delta$, i.e., the set to which the extracted sample $\big(\boldsymbol{\delta}^{(1)},\ldots,\boldsymbol{\delta}^{(N)}\big)$ belongs.

2.3 Linear Interval Inequalities

A system of linear interval inequalities is formulated as $A^I\boldsymbol{y}\le\boldsymbol{b}^I$, where $A^I=\{A:\underline{A}\le A\le\overline{A}\}$ (component-wise inequalities) is an $m\times n$ interval matrix and $\boldsymbol{b}^I=\{\boldsymbol{b}:\underline{b}\le b\le\overline{b}\}$(component-wise inequalities) is an $m-$dimensional

interval vector. A $\boldsymbol{y}_0$ is called a strong solution to the system of linear interval inequalities if it satisfies $A\boldsymbol{y}_0 \leq \boldsymbol{b}$ for each $A \in A^I$ and $\boldsymbol{b} \in \boldsymbol{b}^I$. We denote the set of all strong solutions by Y, and Y is given as follows.

Theorem 2 [28]. $Y = \{\boldsymbol{y}_1 - \boldsymbol{y}_2 : \overline{A}\boldsymbol{y}_1 - \underline{A}\boldsymbol{y}_2 \leq \underline{b}, \boldsymbol{y}_1 \geq \boldsymbol{0}, \boldsymbol{y}_2 \geq \boldsymbol{0}\}$.

A strong solution can be computed by solving a linear programming problem based on Theorem 2. Based on this, for a parametric polynomial of the form $B(\boldsymbol{x}, \boldsymbol{c}) = \sum_{\alpha \in \mathcal{M}} c_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}}$, where $c_{\boldsymbol{\alpha}}$'s are parametric coefficients making $B(\boldsymbol{x}, \boldsymbol{c})$ non-positive over an interval $\boldsymbol{x} \in I$, can be obtained in the way: (1) For each monomial $\boldsymbol{x}^{\boldsymbol{\alpha}} (\boldsymbol{\alpha} \in \mathcal{M})$, we use interval arithmetic to obtain its lower and upper bounds $I^{\alpha-}$ and $I^{\alpha+}$ respectively over the interval I, and yield a linear interval inequality $\sum_{\boldsymbol{\alpha} \in \mathcal{M}} [I^{\alpha-}, I^{\alpha+}] c_{\boldsymbol{\alpha}} \leq 0$. (2) According to Theorem 2, by replacing each variable $c_{\boldsymbol{\alpha}}$ with a difference of two variables $c_{\boldsymbol{\alpha}1}$ and $c_{\boldsymbol{\alpha}2}$, where $c_{\boldsymbol{\alpha}1} \geq 0$ and $c_{\boldsymbol{\alpha}2} \geq 0$, we can replace $[I^{\alpha-}, I^{\alpha+}] c_{\boldsymbol{\alpha}}$ by $I^{\alpha+} c_{\boldsymbol{\alpha}1} - I^{\alpha-} c_{\boldsymbol{\alpha}2}$ and arrive at a linear inequality $\sum_{\boldsymbol{\alpha} \in \mathcal{M}} [I^{\alpha+} c_{\boldsymbol{\alpha}1} - I^{\alpha-} c_{\boldsymbol{\alpha}2}] \leq 0$, denoted as $\psi[\boldsymbol{c}_{\boldsymbol{\alpha}1}, \boldsymbol{c}_{\boldsymbol{\alpha}2}]$. We denote the above procedure as `linear_interval_inequalities` $(B(\boldsymbol{x}, \boldsymbol{c}), I)$. For more details, please refer to [27,29,35]. If $(c_{\boldsymbol{\alpha}1}, c_{\boldsymbol{\alpha}2})_{\boldsymbol{\alpha} \in \mathcal{M}}$, where there exists an $\boldsymbol{\alpha} \in \mathcal{M}$ such that $c_{\boldsymbol{\alpha}1} - c_{\boldsymbol{\alpha}2} \neq 0$, is found, the polynomial B is obtained by substituting $c_{\boldsymbol{\alpha}}$ with $c_{\boldsymbol{\alpha}1} - c_{\boldsymbol{\alpha}2}$.

3 Probably Approximate Safety Verification

In this section we detail our approach for conducting probably approximate safety verification of hybrid systems via the computation of so-called PAC barrier certificates. The concept of PAC barrier certificates is introduced in Subsect. 3.1. The computation method is formulated in Subsect. 3.2.

3.1 PAC Barrier Certificates

A popular approach to safety verification for hybrid systems employs barrier certificates, which partition the state space X into two regions containing forward reachable states of the initial states and backward reachable states of the unsafe states, respectively. There are several variants of barrier certificates and accordingly diverse methods for computing them, e.g., [7,20–22,24,32,37]. In this paper we employ exponential-condition-based barrier certificates from [21] as an instance serving to illustrate our method, which however is not confined to this particular variant of barrier certificates. Exponential-condition-based barrier certificates form the core of Theorem 3 underneath.

Theorem 3 ([21]). *Let $H = (\mathcal{X}, L, X, X_0, \mathtt{Inv}, \boldsymbol{F}, T)$ be a hybrid system. Given $S_\lambda = \{\lambda_l \in \mathbb{R} \mid l \in L\}$ and $S_\sigma = \{\sigma_{l',l} \in \mathbb{R}_{\geq 0} \mid ((l, \cdot), (l', \cdot)) \in T\}$, if there exists a family of functions $(B_l(\boldsymbol{x}) \in \mathcal{C}^1(\mathbb{R}^n))_{l \in L}$ such that for all $l \in L$, the following constraints hold*

$$\begin{aligned} &(1)\ B_l(\boldsymbol{x}) > 0, \forall \boldsymbol{x} \in \mathtt{Uns}(l),\ (2)\ B_l(\boldsymbol{x}) \leq 0, \forall \boldsymbol{x} \in \mathtt{Ini}(l), \\ &(3)\ \frac{\partial B_l}{\partial \boldsymbol{x}}(\boldsymbol{x}) \boldsymbol{f}_l(\boldsymbol{x}, \boldsymbol{d}) + \lambda_l B_l(\boldsymbol{x}) \leq 0, \forall (\boldsymbol{x}, \boldsymbol{d}) \in \mathtt{Inv}(l) \times D(l), \\ &(4)\ B_l(\boldsymbol{x}) - \sigma_{l',l} B_{l'}(\boldsymbol{x}') \leq 0, \forall (\boldsymbol{x}', \boldsymbol{x}) \in \mathtt{G}_{l',l} \times \mathtt{R}_{l',l}(\boldsymbol{x}'), \end{aligned} \tag{4}$$

then the safety of the hybrid system H is guaranteed, i.e., no trajectories starting from $(l, \boldsymbol{x})$ for $l \in L$ and $\boldsymbol{x} \in \mathtt{Ini}(l)$ will enter the unsafe state set X_u.

Based on Theorem 3, semi-definite programming based methods were proposed in [21] to synthesize barrier certificates for polynomial hybrid systems. In order to be able to automatically compute similar certificates for a much wider class of systems, we verify probably approximate safety of hybrid systems and provide a proof of this property via the computation of *PAC barrier certificates*. The concept of PAC barrier certificates is formally presented in Definition 2.

Definition 2. *Let $H = (\mathcal{X}, L, X, X_0, \mathtt{Inv}, \boldsymbol{F}, T)$ be a hybrid system. Given $S_\lambda = \{\lambda_l \in \mathbb{R} \mid l \in L\}$ and $S_\sigma = \{\sigma_{l',l} \in \mathbb{R}_{\geq 0} \mid ((l,\cdot),(l',\cdot)) \in T\}$, a family of functions $(B_l(\boldsymbol{x}) \in \mathcal{C}^1(\mathbb{R}^n))_{l\in L}$ is a family of PAC barrier certificates with respect to $\epsilon \in (0,1)$ and $\beta \in (0,1)$ (or, $\mathtt{PACBC}(\epsilon,\beta)$), if they satisfy the following constraints:*

1. *for each $l \in L$ and $l' \in L$,*

$$\begin{aligned} &(1)\ B_l(\boldsymbol{x}) > 0, \forall \boldsymbol{x} \in \mathtt{Uns}(l),\ (2)\ B_l(\boldsymbol{x}) - \sigma_{l',l} B_{l'}(\boldsymbol{x}') \leq 0, \forall (\boldsymbol{x}', \boldsymbol{x}) \in \mathtt{G}_{l',l} \times \mathtt{R}_{l',l}(\boldsymbol{x}'), \\ &(3)\ \frac{\partial B_l}{\partial \boldsymbol{x}}(\boldsymbol{x}) \boldsymbol{f}_l(\boldsymbol{x}, \boldsymbol{d}) + \lambda_l B_l(\boldsymbol{x}) \leq 0, \forall (\boldsymbol{x}, \boldsymbol{d}) \in \mathtt{Inv}(l) \times D(l). \end{aligned} \tag{5}$$

2. *with confidence of at least $1-\beta$, $\mathtt{Pr}(C) \geq 1-\epsilon$, where $C = \{\boldsymbol{y} \in \mathtt{Ini} \mid B_l(\boldsymbol{x}_l) \leq 0, l \in L\}$ with $\boldsymbol{y} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M)$ and $\boldsymbol{x}_l \in \mathtt{Ini}(l)$.*

The $\mathtt{PACBC}(\epsilon,\beta)$ is an exact barrier certificate for the system H with the initial set $\cup_{l\in L}\{(l,\boldsymbol{x}) \mid \boldsymbol{x} \in \mathtt{Ini}(l) \wedge B_l(\boldsymbol{x}) \leq 0\}$. That is, no trajectories starting from $\cup_{l\in L}\{(l,\boldsymbol{x}) \mid \boldsymbol{x} \in \mathtt{Ini}(l) \wedge B_l(\boldsymbol{x}) \leq 0\}$ will enter X_u, and the set $\cup_{l\in L}\{(l,\boldsymbol{x}) \mid \boldsymbol{x} \in \mathtt{Ini}(l) \wedge B_l(\boldsymbol{x}) \leq 0\}$ is an under-approximation of the set of initial states rendering H safe, e.g.,[33,34,36]. However, it is just a PAC barrier certificate for the system H with the initial set X_0.

Theorem 4. *If $(B_l(\boldsymbol{x}) \in \mathcal{C}^1(\mathbb{R}^n))_{l\in L}$ is $\mathtt{PACBC}(\epsilon,\beta)$, the system H is $\mathtt{PAS}(\epsilon,\beta)$.*

Proof. Let $C = \{\boldsymbol{y} \in \mathtt{Ini} \mid B_l(\boldsymbol{x}_l) \leq 0, l \in L\}$, where $\boldsymbol{y} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M)$ with $\boldsymbol{x}_l \in \mathtt{Ini}(l)$. From constraint (5) in Definition 2, we have that trajectories starting from $\cup_{l\in L}\{(l,\boldsymbol{x}) \mid \boldsymbol{x} \in \mathtt{Ini}(l) \wedge B_l(\boldsymbol{x}) \leq 0\}$ cannot enter X_u. Also, since $\mathtt{Pr}(C) \geq 1-\epsilon$ with at least $1-\beta$ confidence, H is $\mathtt{PAS}(\epsilon,\beta)$ from Definition 1. □

Corollary 1 is an immediate consequence of Definition 2 and Theorem 3.

Corollary 1. *Suppose that $(B_l(\boldsymbol{x}) \in \mathcal{C}^1(\mathbb{R}^n))_{l\in L}$ is $\mathtt{PACBC}(\epsilon,\beta)$. If $\mathtt{Ini}(l) \subseteq \{\boldsymbol{x} \in \mathtt{Ini}(l) \mid B_l(\boldsymbol{x}) \leq 0\}$ for $l \in L$, the hybrid system H is safe.*

Another benefit of computing $\mathtt{PACBC}(\epsilon,\beta)$ is to conduct probabilistic safety verification of hybrid systems.

Corollary 2. *Suppose that $(B_l(\boldsymbol{x}) \in \mathcal{C}^1(\mathbb{R}^n))_{l\in L}$ is $\mathtt{PACBC}(\epsilon,\beta)$. If $\mathtt{Pr}(C) \geq 1-\epsilon$, where $C = \{\boldsymbol{y} \in \mathtt{Ini} \mid B_l(\boldsymbol{x}_l) \leq 0$ for $l \in L\}$ with $\boldsymbol{y} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M)$ and $\boldsymbol{x}_l \in \mathtt{Ini}(l)$, then $\mathtt{Pr}_l(C_l) \geq 1-\epsilon$ for $l \in L$, where C_l is a set of states $\boldsymbol{x}$s in $\mathtt{Ini}(l)$ such that trajectories starting from $(l,\boldsymbol{x})$ never enter X_u.*

Proof. Since $\mathtt{Pr} = \mathtt{Pr}_1 \times \ldots \times \mathtt{Pr}_l$, we have that $\mathtt{Pr}_l(C_l') \geq 1-\epsilon$, where $C_l' = \{\boldsymbol{x} \in \mathtt{Ini}(l) \mid B_l(\boldsymbol{x}) \leq 0\}$. Also, since $C_l' \subseteq C_l$, we have the conclusion. □

[31] developed a tool ProbReach to address the probabilistic safety verification problem in Corollary 2 for hybrid systems. Since reachable set computation based techniques are used in [31], it is limited to safety verification of hybrid systems over finite time horizons. [19] proposed a bilinear semidefinite programming based method to compute probabilistic barrier certificates for polynomial hybrid systems. Unfortunately, the bilinear semidefinite program falls within nonlinear programming framework and is notoriously hard to solve.

3.2 Probably Approximate Safety Verification

In this section we present our linear programming based method for synthesizing $\mathtt{PACBC}(\epsilon, \beta)$ and thus conducting probably approximate safety verification of hybrid systems. The linear program is constructed based on linear interval inequalities and scenario approaches.

We first select barrier certificate templates $(B_l(c_{l,1}, \ldots, c_{l,i_l}, \boldsymbol{x}))_{l \in L}$ such that (1) $B_l(c_{l,1}, \ldots, c_{l,i_l}, \boldsymbol{x}) \in \mathcal{C}^1(\mathbb{R}^n)$ is a linear function in $c_{l,1}, \ldots, c_{l,i_l}$ for $\boldsymbol{x} \in \mathbb{R}^n$, where $(c_{l,j})_{j=1}^{i_l}$ are unknown parameters and $i_l \geq 1$ is a positive integer. For convenience $\boldsymbol{c}_l$ is used to denote $(c_{l,1}, \ldots, c_{l,i_l})$ in the rest of this paper. (2) Let $C_r = \{\boldsymbol{x} \in \mathtt{Ini}(l) \mid B_l(\boldsymbol{c}_l, \boldsymbol{x}) = r\}$ for $r \in \mathbb{R}$,

$$\mathtt{Pr}_l(C_r) = 0, \forall l \in L, \forall r \in \mathbb{R}. \tag{6}$$

This requirement is to ensure that the solution computed by scenario approaches satisfies Assumption 2, which will be reflected in Lemma 1. Generally, polynomial functions satisfy the requirement (6).

Under the assumption that ϵ is given (later, we will introduce how to give an appropriate ϵ), we try to compute $(\boldsymbol{c}_l)_{l \in L}$ by solving the following chance-constrained optimization:

$$\min_{\boldsymbol{c}_l, l \in L, \theta} \theta + \sum_{l=1}^{M} w_l \int_{\mathtt{Ini}(l)} B(\boldsymbol{c}_l, \boldsymbol{x}) d\boldsymbol{x}, \tag{7}$$

$$\mathtt{s.t.Pr}(\{\boldsymbol{y} \in \mathtt{Ini} \mid \max_{l \in L} B(\boldsymbol{c}_l, \boldsymbol{x}_l) \leq \theta\}) \geq 1-\epsilon, \tag{8}$$

$$0 \leq \theta \leq U_\theta, \tag{9}$$

$$\mathtt{for\ each}\ l \in L \text{ and } l \in L' : \tag{10}$$

$$B_l(\boldsymbol{x}) - \zeta_l \geq 0, \forall \boldsymbol{x} \in \mathtt{Uns}(l), \tag{11}$$

$$\frac{\partial B_l}{\partial \boldsymbol{x}}(\boldsymbol{x}) \boldsymbol{f}_l(\boldsymbol{x}, \boldsymbol{d}) + \lambda_l B_l(\boldsymbol{x}) \leq 0, \forall (\boldsymbol{x}, \boldsymbol{d}) \in \mathtt{Inv}(l) \times D(l), \tag{12}$$

$$B_l(\boldsymbol{x}) - \sigma_{l',l} B_{l'}(\boldsymbol{x}') \leq 0, \forall (\boldsymbol{x}', \boldsymbol{x}) \in \mathtt{G}_{l',l} \times \mathtt{R}_{l',l}(\boldsymbol{x}'), \tag{13}$$

where $\boldsymbol{y} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M)$ with $\boldsymbol{x}_l \in \mathtt{Ini}(l)$, $\sigma_{l',l} \in \mathbb{R}_{\geq 0}$, $\zeta_l \in \mathbb{R}_{>0}$ and $\lambda_l \in \mathbb{R}$ are given, and U_θ is a user-defined positive bound for θ. w_ls, $l = 1, \ldots, M$, are given

positive values such that $\sum_{l=1}^{M} w_l = 1$. In (7), w_l for $l \in L$ represents the relative significance of the l_{th} set $\texttt{Ini}(l)$. The minimum operator on $\int_{\texttt{Ini}(l)} B(\boldsymbol{c}_l, \boldsymbol{x})d\boldsymbol{x}$ aims to find $\boldsymbol{c}_l$ such that $\{\boldsymbol{x} \in \texttt{Ini}(l) \mid B(\boldsymbol{c}_l, \boldsymbol{x}) \leq 0\}$, which is a set of states $\boldsymbol{x}$s such that trajectories starting from $(l, \boldsymbol{x})$ never enter X_u, is as large as possible.

Solving the chance-constrained optimization (7)–(13) directly is notoriously hard. It generally falls within the nonlinear programming framework and is an NP-hard problem. Below we show the use of scenario approaches and linear interval inequalities to encode (7)–(13) as a linear programming problem, whose solution provides a family of PAC barrier certificates with respect to ϵ and β.

We first relax constraints (11)–(13) to linear constraints over $\boldsymbol{c}_l$ using linear interval inequalities. For this sake, we first construct a family of interval boxes $(I^i_{\texttt{U}(l)})_{i=1}^{k_{1,l}}$, $(I^i_{\texttt{Inv}(l)})_{i=1}^{k_{2,l}}$ and $(I^i_{\texttt{G}_{l',l}})_{i=1}^{k_{3,l}}$ such that $\texttt{Uns}(l) \subseteq \cup_{i=1}^{k_{1,l}} I^i_{\texttt{U}(l)}$, $\texttt{Inv}(l) \times D(l) \subseteq \cup_{i=1}^{k_{2,l}} I^i_{\texttt{Inv}(l)}$ and $\texttt{G}_{l',l} \subseteq \cup_{i=1}^{k_{3,l}} I^i_{\texttt{G}_{l',l}}$, respectively. Then, for $i = 1, \ldots, k_{1,l}$, we obtain a linear relaxation $\psi_{1,i}[\boldsymbol{c}_{1,l}, \boldsymbol{c}_{2,l}]$ of the constraint $-B_l(\boldsymbol{c}_l, \boldsymbol{x}) + \zeta_l \leq 0$ for $\boldsymbol{x} \in \texttt{Uns}(l)$ based on $\texttt{linear\_interval\_inequalities}(-B_l(\boldsymbol{c}_l, \boldsymbol{x}) + \zeta_l, I^i_{\texttt{U}(l)})$, where $\zeta_l \in \mathbb{R}_{>0}$ is a user-defined small positive value. If $(\boldsymbol{c}_{1,l}, \boldsymbol{c}_{2,l})$ satisfies $\wedge_{i=1}^{k_{1,l}} \psi_{1,i}[\boldsymbol{c}_{1,l}, \boldsymbol{c}_{2,l}]$, $-B_l(\boldsymbol{c}_{1,l} - \boldsymbol{c}_{2,l}, \boldsymbol{x}) < 0$ for $\boldsymbol{x} \in \texttt{Uns}(l)$. Analogously, we obtain linear relaxations $\wedge_{i=1}^{k_{2,l}} \psi_{2,i}[\boldsymbol{c}_{1,l}, \boldsymbol{c}_{2,l}]$ and $\wedge_{i=1}^{k_{3,l}} \psi_{3,i}[\boldsymbol{c}_{1,l}, \boldsymbol{c}_{2,l}]$ of constraints (12) and (13), respectively. Therefore, if $(\boldsymbol{c}_{1,l}, \boldsymbol{c}_{2,l})_{l \in L}$ satisfies

$$\wedge_{i=1}^{k_{1,l}} \psi_{1,i}[\boldsymbol{c}_{1,l}, \boldsymbol{c}_{2,l}] \bigwedge \wedge_{i=1}^{k_{2,l}} \psi_{2,i}[\boldsymbol{c}_{1,l}, \boldsymbol{c}_{2,l}] \bigwedge \wedge_{i=1}^{k_{3,l}} \psi_{3,i}[\boldsymbol{c}_{1,l}, \boldsymbol{c}_{2,l}], \tag{14}$$

$(B_l(\boldsymbol{c}_{1,l} - \boldsymbol{c}_{2,l}, \boldsymbol{x}))_{l \in L}$ satisfies constraints (11), (12) and (13). For ease of exposition, we denote (14) by $\psi_l[\boldsymbol{c}_{1,l}, \boldsymbol{c}_{2,l}]$.

Next, we substitute the chance constraint (8) with N hard constraints, which are constructed based on N independent and identically distributed samples $(\boldsymbol{y}_i)_{i=1}^{N}$ with $\boldsymbol{y}_i = \left(\boldsymbol{x}_{1,i}, \ldots, \boldsymbol{x}_{M,i}\right)$ extracted from the set $\texttt{Ini}$ according to the probabilistic distribution $\texttt{Pr}$, where $\boldsymbol{x}_{l,i} \in \texttt{Ini}(l)$ for $l = 1, \ldots, M$. The N hard constraints over $\boldsymbol{c}_l$ and θ are $\max_{l \in L} B_l(\boldsymbol{c}_l, \boldsymbol{x}_{l,i}) \leq \theta, i = 1, \ldots, N$. Obviously, $B_l(\boldsymbol{c}_l, \boldsymbol{x}_{l,i}) \leq \theta$ is a linear function in $\boldsymbol{c}_l$ and θ.

Finally, we obtain a linear relaxation (15) over $(\boldsymbol{c}_{i,l})$ and θ for solving (7)–(13),

$$\begin{aligned} &\min_{\boldsymbol{c}_{i,l}, i=1,2, l \in L, \theta} \theta + \sum_{l=1}^{M} w_l \int_{\texttt{Ini}(l)} B(\boldsymbol{c}_{1,l} - \boldsymbol{c}_{2,l}, \boldsymbol{x})d\boldsymbol{x} \\ &\texttt{s.t. for each } i = 1, \ldots, N : \max_{l \in L} B_l(\boldsymbol{c}_{1,l} - \boldsymbol{c}_{2,l}, \boldsymbol{x}_{l,i}) \leq \theta, \\ &\quad \texttt{for each } l \in L : (1)\ \psi_l[\boldsymbol{c}_{1,l}, \boldsymbol{c}_{2,l}],\ (2)\ 0 \leq \theta \leq U_\theta, \boldsymbol{c}_{i,l} \leq U_{\boldsymbol{c}}, i = 1, 2, \end{aligned} \tag{15}$$

where $U_{\boldsymbol{c}} \in \mathbb{R}_{>0}$ is a pre-specified upper bound for $\boldsymbol{c}_{i,l}$ for $l \in L$ and $i = 1, 2$, and $U_\theta \in \mathbb{R}_{>0}$ is pre-specified upper bound for θ. Let $(\boldsymbol{c}^*_{1,1}, \boldsymbol{c}^*_{2,1}, \ldots, \boldsymbol{c}^*_{1,M}, \boldsymbol{c}^*_{2,M}, \theta^*)$ be an optimal solution to the linear program (15).

Remark 1. After obtaining $(\boldsymbol{c}^*_{1,1}, \boldsymbol{c}^*_{2,1}, \ldots, \boldsymbol{c}^*_{1,M}, \boldsymbol{c}^*_{2,M}, \theta^*)$, $\texttt{Pr}(C)$ can be estimated based on the Chernoff-Hoeffding Bound [18] in the statistical context.

The Chernoff-Hoeffding Bound formulates that with a confidence of at least $1-e^{-2N\epsilon'^2}$, $\mathtt{Pr}(C) \geq p-\epsilon'$ with $p=\frac{N'}{N}$, where C is defined in Definition 1 and N' is the number of sample states $\boldsymbol{y}_i$s such that $\max_{l\in L} B_l(\boldsymbol{c}^*_{1,l}-\boldsymbol{c}^*_{2,l},\boldsymbol{x}_{l,i}) \leq 0$. In the following we give a different estimation based on scenario approaches. The difference between these two estimations will be presented in examples.

Based on the computed solution $(\boldsymbol{c}^*_{1,1},\boldsymbol{c}^*_{2,1},\ldots,\boldsymbol{c}^*_{1,M},\boldsymbol{c}^*_{2,M})$, we further relax the linear program (15) as a new linear program over the single variable θ:

$$\begin{aligned}
&\min_{\theta}\ \theta+\sum_{l=1}^{M} w_l \int_{\mathtt{Ini}(l)} B(\boldsymbol{c}^*_{1,l}-\boldsymbol{c}^*_{2,l},\boldsymbol{x})d\boldsymbol{x}\\
&\mathtt{s.t.\ for\ \ each}\ \ i=1,\ldots,N: \max_{l\in L} B_l(\boldsymbol{c}^*_{1,l}-\boldsymbol{c}^*_{2,l},\boldsymbol{x}_{l,i}) \leq \theta,\\
&\qquad \mathtt{for\ \ each}\ \ l\in L: (1)\ \psi_l[\boldsymbol{c}^*_{1,l},\boldsymbol{c}^*_{2,l}],\ (2)\ 0\leq\theta\leq U_\theta, \boldsymbol{c}^*_{i,l}\leq U_{\boldsymbol{c}}, i=1,2.
\end{aligned} \tag{16}$$

Obviously, (16) is feasible. Also, the optimal value of θ is unique and equal to θ^*. Assumption 1 is satisfied.

Then we remove samples from $(\boldsymbol{x}_{1,i},\ldots,\boldsymbol{x}_{M,i})_{i=1}^N$ such that $\max_{l\in L} B_l(\boldsymbol{c}^*_{1,l}-\boldsymbol{c}^*_{2,l},\boldsymbol{x}_{l,i}) > 0$, and denote the indexes of removed constraints by $\{i_1,\ldots,i_k\}$, leading eventually to the following linear program,

$$\begin{aligned}
&\min_{\theta}\ \theta+\sum_{l=1}^{M} w_l \int_{\mathtt{Ini}(l)} B(\boldsymbol{c}^*_{1,l}-\boldsymbol{c}^*_{2,l},\boldsymbol{x})d\boldsymbol{x}\\
&\mathtt{s.t.\ \ for\ \ each}\ \ i=1,\ldots,N\setminus\{i_1,\ldots,i_k\}: \max_{l\in L} B_l(\boldsymbol{c}^*_{1,l}-\boldsymbol{c}^*_{2,l},\boldsymbol{x}_{l,i}) \leq \theta,\\
&\qquad \mathtt{for\ \ each}\ \ l\in L: (1)\ \psi_l[\boldsymbol{c}^*_{1,l},\boldsymbol{c}^*_{2,l}],\ (2)\ 0\leq\theta\leq U_\theta, \boldsymbol{c}^*_{i,l}\leq U_{\boldsymbol{c}}, i=1,2.
\end{aligned} \tag{17}$$

Let θ^{**} be an optimal solution to the linear program (17). Obviously, $\theta^{**}=0$.

Remark 2. Although the removed sample $(x_{1,j},\ldots,x_{M,j})$ satisfies $\max_{i\in L} B_l(\boldsymbol{c}^*_{1,i}-\boldsymbol{c}^*_{2,i},\boldsymbol{x}_{l,j}) > 0$, where $j\in\{i_1,\ldots,i_k\}$, it does not indicate that the hybrid system H starting from $(l,\boldsymbol{x}_{l,j})$ will enter X_u, since the existence of barrier certificates satisfying (4) is just a sufficient condition for justifying the safety of the system.

The constraint removal algorithm $\mathcal{A}$ for obtaining (17) can be chosen as $\mathcal{A}(\boldsymbol{y}_1,\ldots,\boldsymbol{y}_N)=\{i_1,\ldots,i_k\}$, where $\big(\max_{l\in L} B_l(\boldsymbol{c}^*_{1,l}-\boldsymbol{c}^*_{2,l},\boldsymbol{x}_{l,i_j})\big)_{j=1}^k$ are the first k largest values in $\big(\max_{l\in L} B_l(\boldsymbol{c}^*_{1,l}-\boldsymbol{c}^*_{2,l},\boldsymbol{x}_{l,i})\big)_{i=1}^N$. Let $\boldsymbol{z}=(\boldsymbol{y}_1,\ldots,\boldsymbol{y}_N)$. According to (6), $\mathtt{Pr}^N(\{\boldsymbol{z}\in\mathtt{Ini}^N|\theta^{**}(\boldsymbol{z})$ violates the k removed constraints$\})=1$, satisfying Assumption 2. This is formally stated in Lemma 1. Obviously, $\theta^{**}(\boldsymbol{z})=\max_{l\in L}\max_{i\in\{1,\ldots,N\}\setminus\{i_1,\ldots,i_k\}} B_l(\boldsymbol{c}^*_{1,i}-\boldsymbol{c}^*_{2,i},\boldsymbol{x}_{l,i})$. Herein, we shall write the optimal solutions to (17) as $\theta^{**}(\boldsymbol{z})$ to emphasize its stochastic nature.

Lemma 1. *Let* $\mathcal{A}(\boldsymbol{y}_1,\ldots,\boldsymbol{y}_N)=\{i_1,\ldots,i_k\}$ *in* (17) *and* $\big(\max_{l\in L} B_l(\boldsymbol{c}^*_{1,l}(\boldsymbol{z})-\boldsymbol{c}^*_{2,l}(\boldsymbol{z}),\boldsymbol{x}_{l,i_j})\big)_{j=1}^{k}$ *be the first* k *largest values in the family* $\big(\max_{l\in L} B_l(\boldsymbol{c}^*_{1,l}(\boldsymbol{z})-\boldsymbol{c}^*_{2,l}(\boldsymbol{z}),\boldsymbol{x}_{l,i})\big)_{i=1}^{N}$. *Then* $\mathtt{Pr}^N(S)=1$, *where*

$$S=\big\{\boldsymbol{z}\in \mathtt{Ini}^N \mid \theta^{**}(\boldsymbol{z}) \text{ violates the } k \text{ removed constraints}\big\}$$

and $\boldsymbol{z}=(\boldsymbol{y}_1,\ldots,\boldsymbol{y}_N)$, $\boldsymbol{y}_i=(\boldsymbol{x}_{1,i},\ldots,\boldsymbol{x}_{M,i})$ *with* $\boldsymbol{x}_{l,i}\in \mathtt{Ini}(l)$ *for* $l\in L$ *and* $i\in\{1,\ldots,N\}$.

Proof. Let $A=\big\{\boldsymbol{z}\in \mathtt{Ini}^N | \theta^{**}(\boldsymbol{z}) \text{ does not violate the } k \text{ removed constraints}\big\}$.

Let $\mathcal{M}=\{1,\ldots,N\}$, $\boldsymbol{z}_0=(\boldsymbol{y}_{1,0},\ldots,\boldsymbol{y}_{N,0})\in A$ with $\boldsymbol{y}_{i,0}=(\boldsymbol{x}_{1,i,0},\ldots,\boldsymbol{x}_{M,i,0})$ and $\boldsymbol{x}_{l,i,0}\in \mathtt{Ini}(l)$ for $l\in L$ and $i\in\mathcal{M}$, and $\mathcal{M}'=\{i_1,\ldots,i_k\}$. Consequently,

$$\max_{i\in\mathcal{M}\setminus\mathcal{M}'}\max_{l\in L} B_l(\boldsymbol{c}^*_{1,l}(\boldsymbol{z}_0)-\boldsymbol{c}^*_{2,l}(\boldsymbol{z}_0),\boldsymbol{x}_{l,i,0})=\min_{j\in\mathcal{M}'}\max_{l\in L} B_l(\boldsymbol{c}^*_{1,l}(\boldsymbol{z}_0)-\boldsymbol{c}^*_{2,l}(\boldsymbol{z}_0),\boldsymbol{x}_{l,j,0}).$$

Let $B=\left\{\boldsymbol{z}\in \mathtt{Ini}^N \;\middle|\; \begin{array}{l}\max\limits_{i\in\mathcal{M}\setminus\mathcal{M}'}\max\limits_{l\in L} B_l(\boldsymbol{c}^*_{1,l}(\boldsymbol{z})-\boldsymbol{c}^*_{2,l}(\boldsymbol{z}),\boldsymbol{x}_{l,i})\\ =\min\limits_{j\in\mathcal{M}'}\max\limits_{l\in L} B_l(\boldsymbol{c}^*_{1,l}(\boldsymbol{z})-\boldsymbol{c}^*_{2,l}(\boldsymbol{z}),\boldsymbol{x}_{l,j})\end{array}\right\}$. Obviously, $A=B$. Also, since $\mathtt{Pr}_l\big(\{\boldsymbol{x}\in \mathtt{Ini}(l) \mid B_l(\boldsymbol{c}^*_{1,l}-\boldsymbol{c}^*_{2,l},\boldsymbol{x})=r\}\big)=0$ for $r\in\mathbb{R}$ according to (6), we have that $\mathtt{Pr}\big(\{\boldsymbol{y}\in \mathtt{Ini} \mid \max_{l\in L} B_l(\boldsymbol{c}^*_{1,l}-\boldsymbol{c}^*_{2,l},\boldsymbol{x}_l)=r\}\big)=0$ for $r\in\mathbb{R}$. Therefore, $\mathtt{Pr}^N(B)=0$ and consequently $\mathtt{Pr}^N(A)=0$. □

Therefore, according to Theorem 1, if ϵ satisfies $\sum_{i=0}^{k}\binom{N}{i}\epsilon^i(1-\epsilon)^{N-i}\le\beta$, $(B_l(\boldsymbol{c}^*_{1,l}-\boldsymbol{c}^*_{2,l},\boldsymbol{x}))_{l\in L}$ is $\mathtt{PACBC}(\epsilon,\beta)$.

Theorem 5. *If* ϵ *satisfies* $\sum_{i=0}^{k}\binom{N}{i}\epsilon^i(1-\epsilon)^{N-i}\le\beta$, *the system* H *is* $\mathtt{PAS}(\epsilon,\beta)$.

Proof. We reformulate (16) equivalently as the following linear program over θ,

$$\begin{aligned}&\min_{\theta}\ \theta+\sum_{l=1}^{M} w_l\int_{\mathtt{Ini}(l)} B(\boldsymbol{c}^*_{1,l}-\boldsymbol{c}^*_{2,l},\boldsymbol{x})d\boldsymbol{x}\\ &\mathtt{s.t.}\quad \mathtt{for\ \ each}\ i=1,\ldots,N:\max_{l\in L} B_l(\boldsymbol{c}^*_{1,l}-\boldsymbol{c}^*_{2,l},\boldsymbol{y}_i)\le\theta,\\ &\qquad \mathtt{for\ \ each}\ l\in L:(1)\ \psi_l[\boldsymbol{c}^*_{1,l},\boldsymbol{c}^*_{2,l}],\ (2)\ 0\le\theta\le U_\theta,\boldsymbol{c}^*_{i,l}\le U_c, i=1,2,\end{aligned}\tag{18}$$

where $B_l(\boldsymbol{c}^*_{1,l}-\boldsymbol{c}^*_{2,l},\boldsymbol{y}_i)=B_l(\boldsymbol{c}^*_{1,l}-\boldsymbol{c}^*_{2,l},\boldsymbol{x}_{l,i})$ and $\boldsymbol{y}_i=(\boldsymbol{x}_{1,i},\ldots,\boldsymbol{x}_{M,i})$ with $\boldsymbol{x}_{l,i}\in \mathtt{Ini}(l)$ and $l\in L$. The number of variables in (18) is 1.

Optimal solutions to (18) are optimal solutions to (17), and vice versa. Obviously, (18) is feasible and has unique solution. Also, according to Lemma 1, Assumption 2 holds. Thus, according to Definition 2 and Theorem 1, $(B_l(\boldsymbol{c}^*_{1,l}-\boldsymbol{c}^*_{2,l},\boldsymbol{x}))_{l\in L}$ is $\mathtt{PACBC}(\epsilon,\beta)$. Thus, the system H is $\mathtt{PAS}(\epsilon,\beta)$ from Theorem 4. □

If $k>0$, ϵ satisfying Theorem 5 can be explicitly relaxed as the following constraint according to inequation (8) in [5]:

$$\epsilon\ge\min\{1,\frac{1}{N}[k+\ln\frac{1}{\beta}+\sqrt{\ln^2\frac{1}{\beta}+2k\ln\frac{1}{\beta}}]\}.\tag{19}$$

If $k = 0$, ϵ satisfying Theorem 5 can be explicitly relaxed as the following constraint according to inequation (4) in [6]:

$$\epsilon \geq 1 - \beta^{\frac{1}{N}}. \tag{20}$$

Remark 3. One may compute the probability of continuous states leading to the satisfiability of safety properties via calculating $\int_C d\mathtt{Pr}$, where $C = \{\boldsymbol{x} \in \mathtt{Ini}(1) \mid B_1(\boldsymbol{c}^*_{1,l} - \boldsymbol{c}^*_{2,l}, \boldsymbol{x}) \leq 0\} \times \ldots \times \{\boldsymbol{x} \in \mathtt{Ini}(M) \mid B_M(\boldsymbol{c}^*_{1,l} - \boldsymbol{c}^*_{2,l}, \boldsymbol{x}) \leq 0\}$. Although there are methods, e.g. [16], to compute $\int_C d\mathtt{Pr}$, we have to point out that this computation is nontrivial generally, especially for high-dimensional systems.

4 Experiments

In this section we evaluate our method on some examples. Parameters that determine the performance of our method are presented in Table 1. All computations were performed on MATLAB installed on an i7-7500U 2.70 GHz CPU with 32G RAM running Windows 10. In our following computations, we adopt uniform grid spacings when partitioning continuous state spaces.

Example 1. Consider a pendulum described by differential equations

$$\dot{x} = y, \dot{y} = -d_0 \sin(x) - d_1 y,$$

where $\mathtt{Inv}(1) = [-10, 10] \times [-10, 10]$, $\mathtt{Ini}(1) = [-10, 5] \times [8, 10]$, $\mathtt{Uns}(1) = [9, 10] \times [7, 8]$ and $D(1) = \{(d_0, d_1) \mid d_0 \in [0.9, 1.1], d_1 \in [0.9, 1.1]\}$.

The PAC barrier certificate template is $c_0 + c_1 x + c_2 y + c_3 x^2 + c_4 xy + c_5 y^2$. We first try to find a barrier certificate to verify whether this system is safe. The sets $\mathtt{Inv}(1)$, $\mathtt{Uns}(1)$ and $\mathtt{Ini}(1)$ are partitioned into 10^4, 1 and 10^4 interval boxes, respectively. The system of linear constraints constructed by using

Table 1. $\mathtt{dim}_v$: dimension of the state space; $\mathtt{dim}_p$: dimension of the perturbation space; k: number of removed samples; ϵ: error level; β: confidence level; N: number of extracted samples; m: number of variables in (15); ζ: ζ_ls in (7)–(13); σ : $\sigma_{l',l}$s in (7)–(13); γ : γ_ls in (7)–(13); w: weights w_l in (15); U: upper bounds U_c and U_θ in (15); T: computation times (seconds)

Benchmarks	Dimension		Parameter											Time
	$\mathtt{dim}_v$	$\mathtt{dim}_p$	M	ϵ	β	N	k	m	ζ	σ	γ	w	U	T
Ex.3	2	2	1	0.05	10^{-12}	10^4	180	9	10^{-3}	–	1	1	10	19.10
Ex.4	2	0	2	0.47	10^{-12}	10^4	3559	25	10^{-3}	1	1	$\frac{1}{2}$	10	140.73
Ex.4	2	0	2	0.02	10^{-12}	10^4	9	25	10^{-3}	1	1	$\frac{1}{2}$	10	39.79
Ex.4	2	0	2	0.008	10^{-12}	10^4	0	25	10^{-3}	1	1	$\frac{1}{2}$	10	36.65
Ex.5	101	1	1	0.05	10^{-12}	10^4	0	203	10^{-3}	–	1	1	10	148.25

$\texttt{linear\_interval\_inequalities}(\cdot,\cdot)$ to encode the constraints in Theorem 3 is infeasible and consequently we have no knowledge of the safety of this system.

However, if we partition $\texttt{Inv}(1)$ and $\texttt{Uns}(1)$ into 400 and 1 interval boxes respectively, and then sample 10^4 states from $\texttt{Ini}(1)$, we obtain a PAC barrier certificate $B(x,y)$. $\{(x,y) \in \texttt{Ini}(1) \mid B(x,y) \leq 0\}$ is illustrated in Fig. 1. The number of removed samples is 180. Thus, the system is $\texttt{PAS}(0.021, 10^{-12})$. Note that the Chernoff-Hoeffding Bound indicates that the system is $\texttt{PAS}(0.052, 10^{-12})$.

This example also demonstrates that our approach can reduce the computational burden in safety verification of systems, albeit at the price of the computed barrier certificate being only probably approximately correct.

Example 2. We consider a hybrid model of a two-tank system, taken from [8]. The hybrid model has a continuous component of the state-space of dimension $n = 2$. It consists of 2 locations. The flow for each location is described by

$$\boldsymbol{f}_1\begin{pmatrix}x_1\\x_2\end{pmatrix} = \begin{pmatrix}1-\sqrt{x_1}\\ \sqrt{x_1}-\sqrt{x_2}\end{pmatrix}, \boldsymbol{f}_2\begin{pmatrix}x_1\\x_2\end{pmatrix} = \begin{pmatrix}1-\sqrt{x_1-x_2+1}\\ \sqrt{x_1-x_2+1}-\sqrt{x_2}\end{pmatrix}.$$

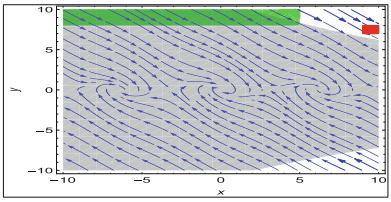

Fig. 1. An illustration of probably approximate safety verification for Example 1. Green, red and gray regions denote $\texttt{Ini}(1)$, $\texttt{Uns}(1)$ and $\{(x,y) \in \texttt{Inv}(1) \mid B(x,y) \leq 0\}$, respectively. Blue curves denote vector fields when $(d_0, d_1) = (1,1)$. (Color figure online)

The other parts of the hybrid automaton are:

1. Initial conditions: $\texttt{Ini}(1) = [5.25, 5.75] \times [0, 0.5]$ and $\texttt{Ini}(2) = [4,6] \times [1,1]$
2. Unsafe regions: $\texttt{Uns}(1) = [4, 4.5] \times [0, 0.5]$ and $\texttt{Uns}(2) = \emptyset$
3. Invariants: $\texttt{Inv}(1) = [4,6] \times [0,1]$ and $\texttt{Inv}(2) = [4,6] \times [1,2]$
4. Guards and resets: (a) $\texttt{G}_{1,2} = [4,6] \times [0.99, 1]$ and $\texttt{R}_{1,2}\begin{pmatrix}x_1\\x_2\end{pmatrix} = \begin{pmatrix}x_1\\1\end{pmatrix}$ (b) $\texttt{G}_{2,1} = \emptyset$ and $\texttt{R}_{2,1}\begin{pmatrix}x_1\\x_2\end{pmatrix} = \begin{pmatrix}x_1\\x_2\end{pmatrix}$.

The PAC barrier certificate templates are polynomials of this form $c_0 + c_1 x + c_2 y + c_3 x^2 + c_4 xy + c_5 y^2$. The sets $\texttt{Inv}(1)$, $\texttt{Inv}(2)$, $\texttt{G}_{1,2}$ are partitioned into 100, 1 and 1 interval boxes, respectively.

1. When no partition operator is implemented on $\texttt{Uns}(1)$, the number k of removed samples is 3559. According to (19), the system is $\texttt{PAS}(0.359, 10^{-12})$. Note that the Chernoff-Hoeffding Bound indicates that the system is $\texttt{PAS}(0.394, 10^{-12})$.

2. When the unsafe set $\texttt{Uns}(1)$ is partitioned into 25 interval boxes, the number k of removed samples is 9. According to (19), the system is $\texttt{PAS}(0.004, 10^{-12})$. The Chernoff-Hoeffding Bound indicates that the system is $\texttt{PAS}(0.039, 10^{-12})$.
3. When the unsafe set $\texttt{Uns}(1)$ is partitioned into 100 interval boxes, the number k of removed samples is 0. According to (20), the system is $\texttt{PAS}(0.003, 10^{-12})$. The Chernoff-Hoeffding Bound indicates that the system is $\texttt{PAS}(0.038, 10^{-12})$. For this case we use the satisfiability checker iSAT3 [12] to obtain that the computed PAC barrier certificate actually is a true barrier certificate satisfying (4), indicating that this system is safe.
The zero sublevel sets of the computed $\texttt{PACBC}(\epsilon, \beta)$ for these three cases are illustrated in Fig. 2. From this example we observe that the size of linear program (15) depends on these two probability measures ϵ and β.

Example 3. To demonstrate applicability of our approach to high-dimensional systems, we consider a scalable non-polynomial example adapted from [25], which we instantiate with a rather high continuous dimension of 101.

$$\dot{x}_1 = d_0 + \frac{1}{100}\Big(\sum_{i\in\{1,\ldots,l\}} x_{i+1} + x_{i+2}\Big),$$

$$\dot{x}_2 = x_3, \dot{x}_3 = -10\sin x_2 - x_2,$$

$$\ldots$$

$$\dot{x}_{2l} = x_{2l+1}, \dot{x}_{2l+1} = -10\sin x_{2l} - x_2,$$

where $l = 50$, $D(1) = \{d_0 \mid d_0 \in [0.9, 1.1]\}$, $\texttt{Inv}(1) = [-0.3, 0.3]^{2l+1}$, $\texttt{Ini}(1) = [-0.30, 0.00] \times [-0.2, 0.30]^{2l}$ and $\texttt{Uns}(1) = [-0.20, -0.15] \times [-0.30, -0.25]^{2l}$.

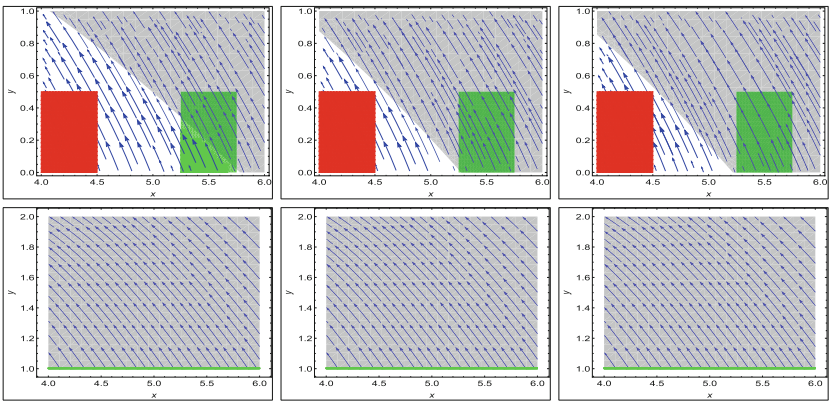

Fig. 2. An illustration of probably approximate safety verification for Example 2 with Case 1–3 (from left to right). Above: Gray region, Green region an Red region denote $\{(x, y) \in \texttt{Inv}(1) \mid B_1(x, y) \leq 0\}$, $\texttt{Ini}(1)$ and $\texttt{Uns}(1)$, respectively. Below: Gray region and Green region denote $\{(x, y) \in \texttt{Inv}(2) \mid B_2(x, y) \leq 0\}$ and $\texttt{Ini}(2)$, respectively. (Color figure online)

The PAC barrier certificate template is chosen as $c_0 + \sum_{i=1}^{101} c_i x_i$. When no partition operator is implemented on the invariant set $\texttt{Inv}(1)$, unsafe set $\texttt{Uns}(1)$ and initial set $\texttt{Ini}(1)$, the system of linear constraints constructed by using $\texttt{linear\_interval\_inequalities}(\cdot,\cdot)$ to encode the constraints in Theorem 3 is infeasible. However, our method verifies that the system is $\texttt{PAS}(0.003, 10^{-12})$ when no partition operator is implemented on $\texttt{Inv}(1)$ and $\texttt{Uns}(1)$. Note that the Chernoff-Hoeffding Bound indicates that the system is $\texttt{PAS}(0.038, 10^{-12})$.

The dimensionality of this example demonstrates that our approach has great potential to open up a promising prospect for formal verification of industrial-scale (hybrid) systems by selecting appropriate ϵ, β and barrier certificate templates. In order to further enhance the scalability of our approach, we will encode constraint (5) using the scenario approach in our future work.

5 Conclusion

We have successfully leveraged the idea of scenario optimization to conduct safety verification of hybrid systems over the infinite time horizon in the framework of PAC learning theory. Based on scenario approaches and linear interval inequalities, a linear programming based method was proposed to compute PAC barrier functions and thus conduct probably approximate safety verification of hybrid systems in the sense that with at least $1 - \beta$ confidence, the probability that the system is safe is larger than $1 - \epsilon$. We have demonstrated the performance and merits of our approach on some benchmark examples.

Acknowledgements. Bai Xue was funded by CAS Pioneer Hundred Talents Program under grant No. Y8YC235015, NSFC under grant No. 61872341 and 61836005. Martin Fränzle was funded by Deutsche Forschungsgemeinschaft through grant FR 2715/4. Hengjun Zhao was funded by NSFC under grant No. 61702425. Naijun Zhan was funded by NSFC under grant No. 61625206 and 61732001. Arvind Easwaran was supported by the Energy Research Institute (ERI@N), NTU, Singapore.

References

1. Alur, R.: Formal verification of hybrid systems. In: EMSOFT 2011, pp. 273–278. IEEE (2011)
2. Alur, R., Courcoubetis, C., Henzinger, T.A., Ho, P.-H.: Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems. In: Grossman, R.L., Nerode, A., Ravn, A.P., Rischel, H. (eds.) HS 1991-1992. LNCS, vol. 736, pp. 209–229. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-57318-6_30
3. Asarin, E., Maler, O.: Achilles and the tortoise climbing up the arithmetical hierarchy. J. Comput. Syst. Sci. **57**(3), 389–398 (1998)
4. Calafiore, G.C.: Random convex programs. SIAM J. Optim. **20**(6), 3427–3464 (2010)
5. Campi, M.C., Garatti, S.: A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality. J. Optim. Theory Appl. **148**(2), 257–280 (2011)

6. Campi, M.C., Garatti, S., Prandini, M.: The scenario approach for systems and control design. Annu. Rev. Control **33**(2), 149–157 (2009)
7. Dai, L., Gan, T., Xia, B., Zhan, N.: Barrier certificates revisited. J. Symb. Comput. **80**, 62–86 (2017)
8. Djaballah, A.: Computation of barrier certificates for dynamical hybrids systems using interval analysis. Université Paris-Saclay (2017)
9. Egyed, A.: Invited talk: a roadmap for engineering safe and secure cyber-physical systems. In: MEDI 2018, pp. 113–114 (2018)
10. Fränzle, M.: Analysis of hybrid systems: an ounce of realism can save an infinity of states. In: Flum, J., Rodriguez-Artalejo, M. (eds.) CSL 1999. LNCS, vol. 1683, pp. 126–139. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48168-0_10
11. Fränzle, M., Gerwinn, S., Kröger, P., Abate, A., Katoen, J.-P.: Multi-objective parameter synthesis in probabilistic hybrid systems. In: Sankaranarayanan, S., Vicario, E. (eds.) FORMATS 2015. LNCS, vol. 9268, pp. 93–107. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22975-1_7
12. Fränzle, M., Herde, C., Teige, T., Ratschan, S., Schubert, T.: Efficient solving of large non-linear arithmetic constraint systems with complex boolean structure. J. Satisf. Boolean Model. Comput. **1**, 209–236 (2007)
13. Fränzle, M., Shirmohammadi, M., Swaminathan, M., Worrell, J.: Costs and rewards in priced timed automata. In: ICALP 2018, pp. 125:1–125:14 (2018)
14. Gao, S., Avigad, J., Clarke, E.M.: Delta-decidability over the reals. In: LICS 2012, pp. 305–314 (2012)
15. Haussler, D.: Probably approximately correct learning. Computer Research Laboratory, University of California, Santa Cruz (1990)
16. Henrion, D., Lasserre, J.B., Savorgnan, C.: Approximate volume and integration for basic semialgebraic sets. SIAM Rev. **51**(4), 722–743 (2009)
17. Henzinger, T.A., Kopke, P.W., Puri, A., Varaiya, P.: What's decidable about hybrid automata? J. Comput. Syst. Sci. **57**(1), 94–124 (1998)
18. Hoeffding, W.: Probability inequalities for sums of bounded random variables. J. Am. Stat. Assoc. **58**(301), 13–30 (1963)
19. Huang, C., Chen, X., Lin, W., Yang, Z., Li, X.: Probabilistic safety verification of stochastic hybrid systems using barrier certificates. ACM Trans. Embed. Comput. Syst. **16**(5), 186:1–186:19 (2017)
20. Kong, H., Bogomolov, S., Schilling, C., Jiang, Y., Henzinger, T.A.: Safety verification of nonlinear hybrid systems based on invariant clusters. In: HSCC 2017, pp. 163–172. ACM (2017)
21. Kong, H., He, F., Song, X., Hung, W.N.N., Gu, M.: Exponential-condition-based barrier certificate generation for safety verification of hybrid systems. In: Sharygina, N., Veith, H. (eds.) CAV 2013. LNCS, vol. 8044, pp. 242–257. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39799-8_17
22. Lin, W., Wu, M., Yang, Z., Zeng, Z.: Exact safety verification of hybrid systems using sums-of-squares representation. Sci. China Inf. Sci. **57**(5), 1–13 (2014)
23. Nahhal, T., Dang, T.: Test coverage for continuous and hybrid systems. In: Damm, W., Hermanns, H. (eds.) CAV 2007. LNCS, vol. 4590, pp. 449–462. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73368-3_47
24. Prajna, S., Jadbabaie, A.: Safety verification of hybrid systems using barrier certificates. In: Alur, R., Pappas, G.J. (eds.) HSCC 2004. LNCS, vol. 2993, pp. 477–492. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24743-2_32
25. Ratschan, S.: Simulation based computation of certificates for safety of dynamical systems. In: Abate, A., Geeraerts, G. (eds.) FORMATS 2017. LNCS, vol. 10419, pp. 303–317. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65765-3_17

26. Ratschan, S., She, Z.: Safety verification of hybrid systems by constraint propagation-based abstraction refinement. ACM Trans. Embed. Comput. S. **6**(1), 8 (2007)
27. Ratschan, S., She, Z.: Providing a basin of attraction to a target region of polynomial systems by computation of lyapunov-like functions. SIAM J. Control Optim. **48**(7), 4377–4394 (2010)
28. Rohn, J.I., Kreslova, J.: Linear interval inequalities. Linear Multilinear Algebra **38**, 79–82 (1994)
29. Sankaranarayanan, S., Chen, X., Ábrahám, E.: Lyapunov function synthesis using Handelman representations. In: NOLCOS 2013, pp. 576–581 (2013)
30. Schupp, S., et al.: Current challenges in the verification of hybrid systems. In: Berger, C., Mousavi, M.R. (eds.) CyPhy 2015. LNCS, vol. 9361, pp. 8–24. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25141-7_2
31. Shmarov, F., Zuliani, P.: ProbReach: verified probabilistic delta-reachability for stochastic hybrid systems. In: HSCC 2015, pp. 134–139 (2015)
32. Sogokon, A., Ghorbal, K., Tan, Y.K., Platzer, A.: Vector barrier certificates and comparison systems. In: Havelund, K., Peleska, J., Roscoe, B., de Vink, E. (eds.) FM 2018. LNCS, vol. 10951, pp. 418–437. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95582-7_25
33. Xue, B. Fränzle, M., Zhan, N.: Inner-approximating reachable sets for polynomial systems with time-varying uncertainties. IEEE Trans. Autom. Control (2019)
34. Xue, B., She, Z., Easwaran, A.: Under-approximating backward reachable sets by polytopes. In: Chaudhuri, S., Farzan, A. (eds.) CAV 2016. LNCS, vol. 9779, pp. 457–476. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41528-4_25
35. Xue, B., She, Z., Easwaran, A.: Underapproximating backward reachable sets by semialgebraic sets. IEEE Trans. Autom. Control **62**(10), 5185–5197 (2017)
36. Xue, B., Wang, Q., Zhan, N., Fränzle, M.: Robust invariant sets generation for state-constrained perturbed polynomial systems. In: HSCC 2019, pp. 128–137 (2019)
37. Zhang, Y., Yang, Z., Lin, W., Zhu, H., Chen, X., Li, X.: Safety verification of nonlinear hybrid systems based on bilinear programming. IEEE Trans. CAD Integr. Circuits Syst. **37**(11), 2768–2778 (2018)
38. Zuliani, P., Platzer, A., Clarke, E.M.: Bayesian statistical model checking with application to simulink/stateflow verification. In: HSCC 2010, pp. 243–252 (2010)