Safety Verification of Hybrid Automata with Transient Safe Modes

Guobin Wang and Jing Liu

***Abstract*—This paper investigates safety verification for hybrid automata with transient safe modes. Safety properties might be violated by solely transient safe modes in some finite time, with respect to permanently safe modes. A new kind of barrier certificate with time constraints is proposed to derive a criterion for safety of such kind of hybrid automata. The improved barrier certificates are more suitable for hybrid automata whose safety heavily relies on well-defined real-time scheduling. With the help of numerical solvers such as SOSTOOLS or SOSOPT for MATLAB, the proposed barrier certificates could be computed by solving some sum-of-squares program as well as bilinear sum-of-squares program problems. The validity of the proposed verification method is supported by a numerical example.**

***Index Terms*—Formal methods, hybrid system, safety verification.**

I. Introduction

Hybrid automata could model dynamical systems exhibiting both continuous and discrete behavior, and arise naturally in a number of engineering applications, such as automotive control, process control, highway systems, manufacturing and so on [1], [2]. An important class of hybrid automata which consists of both permanently safe modes and transient safe modes is more ubiquitous in real application. For permanently safe modes, safety properties are followed over infinite time intervals, while for transient safe modes, safety properties would be violated in finite time eventually. Such temporal safety could be specified by the temporal modal operators such as *always* ($\Box$), *eventually* ($\Diamond$) and *until* (U) as indicated in [3]. Safety of such kind of hybrid automata relies heavily on real-time scheduling of permanently safe modes and transient safe modes, which makes safety verification more challenging than general hybrid automata.

There are two fundamental problems of safety verification of hybrid automata with transient safe modes. The first one is to estimate the maximum time interval over which safety properties could be guaranteed by transient safe modes. The other one is to examine whether the designed real-time scheduling orchestrating permanently safe modes and transient safe modes is capable of guaranteeing the safety of hybrid automata. Theoretical prerequisites for our work include [4]-[7]. The above-mentioned works inspire us to treat real-time scheduling as checking finite-time invariance or stability. In this paper, we follow the basic idea of safety verification based on sum-of-squares (SOS) programming [8], [9] and adopt an exponential-conditioned barrier certificate [10], [11] to estimate the maximum time guaranteeing safety of transient safe modes. Beside that, we can use such barrier certificates to verify the soundness of real-time scheduling enlightened from [12].

Intuitively, barrier certificate is a function of state which maps all the states in the reachable set to non-positive real numbers and all the states in the unsafe set to positive real numbers. Besides, the zero level set of the barrier certificate separates unsafe set from all possible reachable states starting from a given set of initial conditions. This separation of unsafe set from reachable states forms a certificate for safety, hence providing an exact proof of system safety [6], [13]. As indicated in [14], a stronger condition on barrier certificates usually means that less expressive barrier certificates can be synthesized. Unfortunately, synthesizing more expressive barrier certificates often means higher complexity. However, if the existence of a barrier certificate is guaranteed, then with the help of numerical solvers such as SOSTOOLS [15], [16] or SOSOPT [17] for MATLAB, the difficulty of constructing barrier certificates could be alleviated through solving certain SOS program or bilinear SOS program problems. Discussion on the converse barrier certificates theorem presented in [18] enables us to keep faith in the existence of a barrier certificate for safe dynamical systems. Besides theoretical research, safety verification using barrier certificates has acquired greatly development in real applications, especially in the research of safe control strategy of mobile robots [19], [20] and collision avoidance of multi-agents systems [21].

Our contribution of this paper is twofold. First, we propose a barrier certificate based method for estimating the maximum time interval for safety of transient safe modes. Second, we develop a criterion of real-time scheduling of permanently safe modes and transient safe modes sufficient for safety of such hybrid automata.

The paper is organized as follows. Notations and some preliminary definitions are presented in Section II. In Section III, the formal model of hybrid automata, computation as well as formal definition on safety are presented. Section IV formally distinguishes permanently safe modes from transient safe modes by introducing the notion of exponential index for each mode. Besides, lemmas on estimation of such exponential indexes are also given. Section V presents two theorems on the criterion of well designed real-time scheduling and Section VI shows the validity of the proposed

Manuscript received April 25, 2016; revised June 23, 2016.

Guobin Wang is with the School of Computer Science and Software Engineering, East China Normal University, Shanghai 200062, China (e-mail: guobin_hyb_ver@126.com).

Jing Liu is with Shanghai Key Laboratory of Trustworthy Computing, National Trustworthy Embedded Software Engineering Technology Research Center, East China Normal University, Shanghai 200062, China (e-mail: jliu@sei.ecnu.edu.cn).

DOI: 10.7763/IJMO.2016.V6.533

method by a numerical example. Section VI comprises conclusions.

II. Mathematical Preliminaries

A. Notations

Let R denote the field of real numbers. R^n stands for the n-dimensional real vector space. Let the lower case alphabets such as i, j be variables, while alphabet such as x is vectorial variable, $x, x^{'}$ are different variables. Function $p(x)$ is said to be positive definite iff $p(x) > 0$ for all $x \in R^n \setminus \{\vec{0}\}$ with $p(\vec{0}) = 0$. P_n indicates the set of all polynomials in n independent variables. Semi-algebraic sets are described by and only by polynomial equalities and inequalities.

B. Definitions

Definition 2.1: A multivariate polynomial $p(x)$ is a sum-of-squares polynomial (SOS polynomial) if there exist polynomials $\mathrm{p}_1(x), \cdots, p_m(x)$ such that $p(x) = \sum_{i=1}^{m} p_i^2(x)$. We take Σ_n (Σ for short) to denote the set of all SOS polynomials in x.

Definition 2.2: [17] The bilinear SOS program is a subclass of nonlinear program which takes the following form:

$$\text{Min } t$$
$$s.t.$$
$$tb_k(x,d) - a_k(x,d) \in \Sigma, k = 1, \cdots, N_g$$
$$b_k(x,d) \in \Sigma, k = 1, \cdots, N_g$$
$$c_j(x,d) = 0, j = 1, \cdots, N_e$$

where $t \in R, x \in R^n, d \in R^r$ are decision variables. $\{a_k(x,d)\}, \{b_k(x,d)\}, \{c_j(x,d)\}$ are polynomials with given data and affine in d.

III. Hybrid Automata, Computation and Safety

Throughout this paper, we adopt hybrid automata [22] as the hybrid modeling framework. Here, we only consider the special class of hybrid system which exhibits switchings as well as jumps in the state trajectories subordinating to specific dwell time regulations.

Definition 3.1: A hybrid automata is an abstraction of the continuous and discrete behaviors as well as their interactions of a hybrid dynamical system. A hybrid automata is a tuple $H = \{X, M, X_0, I, G, F, T\}$ with the following components:

1) $X \subseteq R^n$ is the state space of a hybrid dynamical system H. A state variable is an n-dimensional vector. The interpretation of the state variables is an assignment of an n-dimensional real-valued vectors to the state variables.
2) M is a finite set of modes. The overall state space of H is denoted by a pair $(m, x) \in M \times X$. Particularly, $m_0 \in M$ is called the admissible initial mode.
3) $X_0 \subseteq X$ is the set of initial states x (x_0 for short) of H, correspondingly, (m_0, x_0) is an admissible initial state of H.
4) I is the invariant labeled with functions $I : M \to 2^X$, which assign to each mode $m \in M$ with some certain set of states $I(m) \subseteq X$. $I(m) \subseteq X$ contains all possible states at mode m, while for different modes $m_i, m_j \in M$, $I(m_i) \cap I(m_j) = \varnothing$ holds.
5) G is the guard labeled with functions $G : M \times M \times R_{\geq 0} \to \{0,1\}$, which assign to each pair of modes $m, m^{'} \in M$ with some dwell time constraints. Such kind of G tells that such switchings are time-dependent for H.
6) $F : X \to 2^{R^n}$ is a set of vectorial differential equation

$$\dot{x} = f_m(x) \tag{1}$$

which constrains the continuous evolution of $x \in I(m)$ at mode m.

7) $T \subseteq G \to M \times X$ is a relation capturing discrete transition with impulsive effects between two distinguished modes under the regulation of G. A transition $((m,x) \to (m^{'}, x^{'})) \in T$ indicates that $(m, m^{'}, t) \in G$ holds and H undergoes a discrete transition from mode m to mode $m^{'}$ with impulsive jumps on states from x to $x^{'}$. Each transition is constrained by following equations

$$m^{'} = g(m) \tag{2}$$
$$x^{'} = h(x) \tag{3}$$

where equations (2) and (3) describe mode switchings and state jumps, respectively.

Definition 3.2: A computation of a hybrid automata $\phi(m, x, t)$ is an infinite sequence of states $(m, x) \in M \times X$ of the following form:

$$(m_0, x_0), (m_1, x_1), \cdots$$

where x_i s are the values assigned to the variables in X such that

Initiation: the initial state of the computation satisfies the initial condition:

$$m_0 \in M \wedge x_0 \in X_0$$

For following consecutive states (m_i, x_i) and (m_{i+1}, x_{i+1}), at least one of the two consecution conditions is satisfied:

Continuous Consecution: for a state (m_i, x_i), there exists a time interval $t \in [0, \delta)(\delta \in R_{>0})$ without discrete transitions, $\phi(m,x,t)$ satisfies:

$$\phi(m,x,0) = (m_i, x_i)$$
$$\dot{\phi}(m,x,t) = f_m(x)$$

Besides, such a piece of continuous computation $\phi(m,x,t)$ is also called a flow.

Discrete Consecution: there exists a discrete transition with impulsive effect $(m_i, x_i) \to (m_{i+1}, x_{i+1}) \in T$ where t^- indicates the instant before discrete transition, while t^+ is the instant after discrete transition, $\phi(m,x,t)$ satisfies:

$$\phi(m_{i+1}, x_{i+1}, t^+) = \phi(g(m_i), h(x_i), t^-)$$

Besides, a state (m_i, x_i) is called a reachable state of H if it appears in some computation of $\phi(m,x,t)$.

Based on hybrid automata and its computation, we would like to discuss formal definition of safety of hybrid automata, particularly distinguish two kinds of time-dependent safety, called transient safety and permanently safety, respectively. We argue that such characterization is necessary for discussions on safety verification of more ubiquitous hybrid dynamical systems.

To consider the issue of safety properties of hybrid automata H, the set of unsafe states should be explicitly defined. Throughout this paper, we take X_u as the set of unsafe states. Therefore, safety of H could be formally defined as:

Definition 3.3: Let a hybrid automata H along with its computation $\phi(m,x,t)$ be given. In general, H is said to be permanently safe if and only if $\phi(m,x,t)$ would not intersect X_u during the whole process, which is formally defined as:

$$\forall t \in [0, +\infty) : \phi(m,x,t) \cap X_u = \varnothing \qquad (4)$$

Notice that safety is guaranteed only when the logical assertion (4) holds over infinite time interval $[0, +\infty)$, we called this kind of safety property as permanently safety with respect to following transient safety.

Definition 3.4: Let a hybrid automata H along with its computation $\phi(m,x,t)$ be given. In general, H is said to be transient safe if and only if there exists a finite time interval $[0, T_H)(T_H \in R_{>0})$ over which $\phi(m,x,t)$ would not intersect X_u formally as:

$$\forall t \in [0, T_H] : \phi(m,x,t) \cap X_u = \varnothing \wedge \exists t \in (T_H, +\infty) :$$
$$\phi(m,x,t) \cap X_u \neq \varnothing \qquad (5)$$

Furthermore, we specify such a T_H satisfying assertion (5) as an admissible transient safe period.

Based on the characterization of permanently safety and transient safety, H acquires the capability to handle more challenging hybrid systems consisting of both permanently safe subsystems and transient safe subsystems, which we believe are more ubiquitous in real applications. Fig. (1) presents an illustrative example, the blue curve is a piece of permanently safe computation, while the green curve is a piece of transient safe computation. Intuitively, activation of permanently safe computation for arbitrary period is safe, while activation of transient safe computation would result into the violation of safety eventually. Naturally, we could still guarantee safety of hybrid automata with transient safe modes under appropriate time-dependent scheduling orchestrating the activation of transient safe modes. Take the case shown in Fig. 1 as an example, transient safe computation is converging to unsafe states region, while permanently safe computation is converging to equilibrium state away from unsafe states region. Under the regulation of appropriate switchings between such permanently safe and transient safe modes, safety still could be guaranteed for the whole system. To present a formal verification methodology for such hybrid automata, scheduling based on dwell time constraint as well as exponential index of each mode for estimating admissible safe period are discussed.

Definition 3.5: Given a hybrid automata H with M_H different modes, for the sequence of switching instants $t_0, t_1, \cdots, t_n, \cdots$, given M_H scalars $\{\tau_1, \cdots, \tau_{M_H}\}$ that for each activating transient safe mode m_i over $[t_k, t_{k+1}) : t_{k+1} - t_k \leq \tau_i$ holds, then τ_j is called the maximum dwell time of transient safe mode m_i of H. Contrarily, for each activating permanently safe mode m_j of H over $[t_k, t_{k+1}) : t_{k+1} - t_k \geq \tau_j$ holds, then τ_j is called the minimum dwell time of permanently safe mode m_j of H.

Intuitively, orchestration of consecutive activation between transient safe modes and permanently safe modes should be appropriately designed for the consideration of safety. As a prerequisite, we would like to propose two lemmas for estimating the exponential indexes for transient and permanently safe modes respectively.

IV. Sufficient Conditions for Permanently Safety and Transient Safety

In this section, we would like to present the sufficient conditions for permanently safety and transient safety, respectively. Based on the sufficient conditions, bilinear SOS program problem is formulated to estimate exponential indexes for permanently and transient safe modes. We focus on safety of modes of hybrid automata H in this section, and leave safety verification of the whole system to the next section.

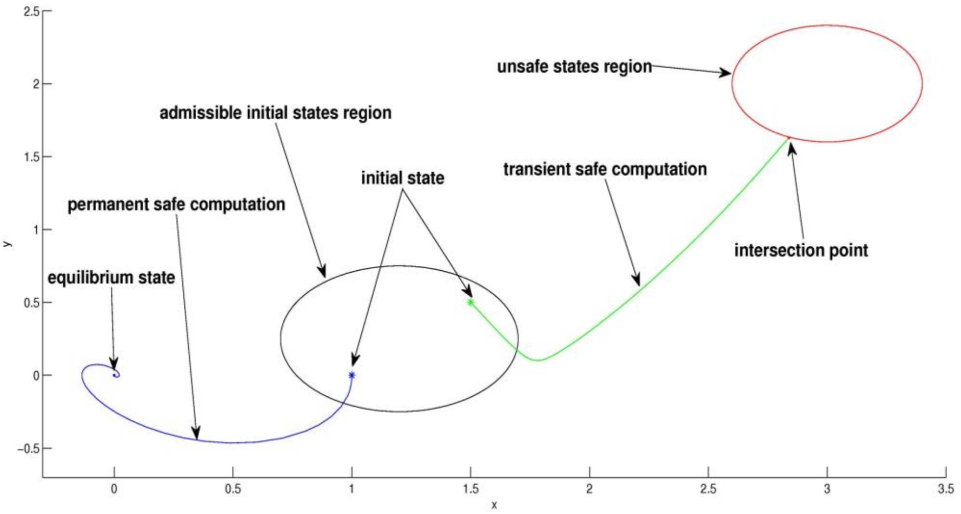

Fig. 1. The above figure gives an intuitive illustration on permanently safe mode and transient safe mode. The blue curve describes the flow of a permanently safe mode which would not intersect the unsafe states region (the red circle) forever. The green curve describes the flow of a transient safe mode which intersects the unsafe states region after a period of activation.

Lemma 4.1: Let a hybrid automata H be given with $m_i, m_j \in \{m_0, \cdots, m_{M_H}\}$ (m_0 is trivially defined) as its modes. Without loss of generality, we assume that H consists of K modes $\{m_1, \cdots, m_K\}$ and $M_H - K$ modes $\{m_{K+1}, \cdots, m_{M_H}\}$. Define two positive scalars c_1, c_2 with $c_1 < c_2$ and positive definite differential functions $B_i(x), B_j(x): M \times X \to R_{>0}$ satisfying

$$\forall x \in X_0 : B_i(x) \le c_1 \qquad (6)$$

$$\forall x \in I(m_i) : B_i(x) \le c_2 \qquad (7)$$

$$\forall x \in X_u : B_i(x) > c_2 \qquad (8)$$

If for each mode $m_i \in \{m_1, \cdots, m_K\}$, there exists a corresponding positive exponential index λ_i such that

$$\forall m_i \in \{m_1, \cdots, m_K\} : \dot{B}_i(x) \le \lambda_i B_i(x) \qquad (9)$$

holds, then m_i is a transient safe mode. Similarly, if for each modes $m_j \in \{m_{K+1}, \cdots, m_{M_H}\}$, there exists a corresponding non-positive exponential index $-\lambda_j$ such that

$$\forall m_j \in \{m_{K+1}, \cdots, m_{M_H}\} : \dot{B}_j(x) \le -\lambda_j B_j(x) \qquad (10)$$

holds, then m_j is a permanently safe mode.

Proof: For each transient safe mode m_j satisfying equation (1), suppose there exists $B_i(x)$ satisfying constraints (6), (7) and (8). According to **Definition 3.4**, there exists a piece of computation $\phi(m, x, t)$ over finite time interval $[0, T_H]$ such that

$$0 < B_i(\phi(m_i, x, 0)) = c_0 \le c_1 \wedge B_i(\phi(m_i, x, T_H)) = c_2$$

Besides, $B_i(x(t))$ is differentiable according to constraint (9), we have

$$B_i(x) \le c_1 e^{\lambda_i t}$$

and

$$T_H = \frac{1}{\lambda_i} \ln \frac{c_2}{c_1}$$

Such that

$$\forall t \in [0, T_H] : B_i(x) \le c_2 \wedge \forall t \in [T_H, +\infty] : B_i(x) > c_2$$

Equivalently, when $0 \le t \le T_H$, mode m_i is safe, while after the instant $t = T_H$, mode m_i is unsafe. Therefore, the existence of a positive scalar λ_i satisfying constraint (9) indicates that m_i is a transient safe mode. Similarly, if there exists a non-positive exponential index $-\lambda_j$ satisfying constraint (10), we have

$$B_i(x) \le c_1 e^{-\lambda_j t} < c_1 < c_2$$

It concludes that mode m_j is safe over infinite time interval $[0, +\infty)$, thus m_j is permanently safe mode.

Naturally, modes of hybrid automata H are categorized into permanently safe modes and transient safe modes with respect to their exponential indexes. It should be pointed out in **Lemma 4.1**, positive definite differential functions $B_i(x)$ and $B_j(x)$ are implicit, we would like to present one of the possible forms of such functions based on Lyapunov functions. Here, we take $B(x) = e^{\|x\|}$ for all modes $m_i, m_j \in M$ such that

- For modes $\{m_1, \cdots, m_K\}$, let $\phi_i(m_i, x, t)$ s stand for flows of subsystem m_i where $i \in \{1,\cdots,K\}$. Then, there exist nonnegative exponential indexes for their flows, and $e^{\phi_i(m_i,x,t)} \le B_i(x) \le \alpha_i e^{\lambda_i t}, i \in \{1,\cdots,K\}$.

- For subsystems $\{m_{K+1},\cdots,m_{M_H}\}$, let $\phi_j(m_j,x,t)$ s stand for flows of subsystem m_j where $j \in \{K+1,\cdots,M_H\}$. Then, there exist negative exponential indexes for their flows, and $B_j(x) \le e^{\phi_j(m_j,x,t)} \le \alpha_j e^{\lambda_j t}, j \in \{K+1,\cdots,M_H\}$.

Naturally, both for those two cases, λ_i s as well as λ_j s are nonnegative. Throughout this paper, we take $\lambda^+ = \max_{1\le i\le k}\{\lambda_i\}$ and $\lambda^- = \min_{K+1\le j\le M_H}\{\lambda_j\}$.

Following bilinear SOS programming problem could be formulated to estimate λ_i and λ_j, meanwhile, corresponding $B_m(x)$ s could also be derived at the same time.

Lemma 4.2: [23], [24] Let a hybrid automata H be given, for each transient safe mode $m_i \in M$, an underestimation of λ_i could be derived by solving a bilinear SOS program

$$\text{Min } \lambda_i$$

$s.t.$

$$\forall x \in X_0 : B_i(x) \le c_1 \tag{11}$$

$$\forall x \in X_u : B_i(x) > c_2 \tag{12}$$

$$\forall x \in I(m) : \lambda_i B_i(x) - \dot{B}_i(x) f_i(x) \in \Sigma \tag{13}$$

$$B_i(x) - x^2 \in \Sigma \tag{14}$$

where λ_i is a scalar decision variable, while $B_i(x)$ is an SOS polynomial decision variable. Similarly, for each permanently safe mode $m_j \in M$, an overestimation of λ_j could be derived by solving a bilinear SOS program

$$\text{Max } \lambda_j$$

$s.t.$

$$\forall x \in X_0 : B_j(x) \le c_1 \tag{15}$$

$$\forall x \in X_u : B_j(x) > c_2 \tag{16}$$

$$\forall x \in I(m) : \lambda_j B_j(x) - \dot{B}_j(x) f_j(x) \in \Sigma \tag{17}$$

$$B_j(x) - x^2 \in \Sigma \tag{18}$$

Based on **Lemma 4.2**, $B_m(x)$ with respect to λ_i could be derived, where λ_i is the index indicating an underestimation of the maximum rate of computation $\phi_i(m_i,x,t)$ converging to unsafe states region. In contrast, λ_j is the index indicating an overestimation of the minimum rate of computation $\phi_j(m_j,x,t)$ diverging away from unsafe states region. Particularly, for transient safe mode m_i, conservative estimation of maximum dwell time τ_i could be derived by

$$\tau_i = \frac{1}{\lambda_i}\ln\frac{c_2}{c_1}.$$

If the activating period of a transient safe mode m_i is restricted to be less than its corresponding maximum dwell time τ_i, the piece of computation does not intersect unsafe states. Additionally, activation of permanently safe mode could leverage states away from unsafe states region. In order to verify safety of H, we have to make a comprehensive comparison between permanently safe and transient safe modes.

V. Formal Safety Verification of Hybrid Automata with Transient Safe Modes

In this section, we would like to discuss sufficient conditions for safety of H over arbitrary finite time interval $[0,T_H]$, considering safety of H could be viewed as the special case by taking $T_H = +\infty$. For hybrid automata with transient safe modes, permanently safe modes should interleave along transient safe modes, which means each mode of such H should be activated periodically. Our assumption of finite time of activation imposes itself upon this periodicity.

Lemma 5.1: For hybrid automata H, specify a constant real number T such that $[0,T_H]$ is the interested fixed finite-time interval. Given (c_1,c_2,T_H), let $T^+[0,t]$ stand for the total activation of the transient safe modes over $[0,t]$ and $T^-[0,t]$ stand for the total activation of permanently safe modes over $[0,t]$. For a given positive real number λ, choose a scalar $\lambda^* \in [0,\lambda]$, if the activating periods of $T^+[0,t]$ and $T^-[0,t]$ satisfy the following inequality:

$$\forall t \in [0,T_H] : \frac{T^-[0,t]}{T^+[0,t]} \ge \frac{\lambda^+ - \lambda^*}{\lambda^- + \lambda^*} \tag{19}$$

Then if

$$c_2 \ge e^{\lambda^* T_H} c_1 \tag{20}$$

Holds, we derive that

$$\forall x_0 \in X_0 : \|x_0\| \le c_1 \wedge \forall x_u \in X_u : \|x\| \ge c_2 \Rightarrow$$
$$e^{T^+[0,T_H]\lambda^+ - T^-[0,T_H]\lambda^-} < c_2 \tag{21}$$

Proof: From inequality (19), following inequality is derived

$$\forall t \in [0,T_H] : T^+[0,t]\lambda^+ - T^-[0,t]\lambda^- \le t\lambda^*$$

And take $t = T_H$, then

$$\|x_0\| e^{T^+[0,T_H]\lambda^+ - T^-[0,T_H]\lambda^-} \le c_1 e^{\lambda^* T_H}$$

together with inequality (20) guarantee the sufficiency of inequality (21).

Theorem 5.1: Given a hybrid automata H with (c_1,c_2,T_H) same to **Lemma 5.1**. Assume that H consists of K transient safe mode $\{m_1,\cdots,m_K\}$ and M_H-K permanently safe mode $\{m_{K+1},\cdots,m_{M_H}\}$. Pick up λ^+, λ^- as $\lambda^+=\max_{1\le i\le k}\{\lambda_i\}$ and $\lambda^-=\min_{K+1\le j\le M_H}\{\lambda_j\}$, respectively. Then specify a positive real number λ, if there exist a piecewise defined function $B(t,x)$ in (t,x) with respect to $B_m(x)$ corresponding to the sequence of activating modes and a positive real number λ^* satisfying following constraints:

$$\forall x\in X_0: B(t,x)\le c_1 \tag{22}$$

$$\forall x\in X_u: B(t,x)> c_2 \tag{23}$$

$$\forall x\in I: \dot{B}(t,x)-\lambda^* B(t,x)\le 0 \tag{24}$$

$$0<\lambda^*\le\lambda \tag{25}$$

$$\lambda<-\frac{1}{T_H}\ln\frac{c_1}{c_2} \tag{26}$$

$$\lambda^* T_H\le \lambda T_H \tag{27}$$

$$\forall t\in(0,T_H]: \frac{T^-(0,t)}{T^+(0,t)}\ge\frac{\lambda^+-\lambda^*}{\lambda^-+\lambda^*} \tag{28}$$

Then, any computation starting from X_0 would not intersect the unsafe states X_u over $[0,T_H]$, thus H is safe over $[0,T_H]$.

Proof: From inequality (27), derive $\frac{1}{T_H}\le\lambda-\lambda^*$, together with the constraint (26), we get that $\frac{1}{T_H}\le\frac{1}{T_H}\ln\frac{c_2}{c_1}-\lambda^*$, multiply T_H and then exponentiate on both sides, $e\le\frac{\frac{c_2}{c_1}}{e^{\lambda^* T_H}}$. Then we have $c_1 e^{\lambda^* T_H}\le c_2$. Take inequalities (22) and (24) together, and integrate $B(t,x)$ piecewise over $[0,T_H]$, we have $\int_0^{T_H} B(t,x)dt\le c_1 e^{\lambda^* T_H}$, consider the inequality (28) and **Lemma 5.1**, there exists a real number c such that $c_1 e^{\lambda^* T_H}\le c_2$ holds. Therefore, we can conclude that the value of $B(t,x)$ of all states starting from initial states along the trajectories is less than or equal to c_2, while $B(t,x)$ for all unsafe states is greater than c_2, therefore, H is safe over finite-time interval $[0,T_H]$.

λ denotes the rate of states x converging to the unsafe states set. During the activation of transient safe modes, x is converging to unsafe states, while during the activation of permanently safe modes, x is diverging away from unsafe states, and λ^* indicates the convergence rate of x towards unsafe states. Transition is also assumed to be harmful, for $\lambda-\lambda^*$ indicates the index of transition approaching to unsafe states set.

Lemma 5.2: [9] Given functions $g_0(x),g_1(x),\cdots,g_m(x)\in P_n$, if there exist $s_1,s_2,\cdots,s_m\in\Sigma_n$ such that $g_0(x)-\sum_{i=1}^{m}s_i g_i(x)\in\Sigma_n$, then it holds that $\{x\in R_n: g_1(x),\cdots,g_m(x)\ge 0\}\subseteq\{x\in R_n: g_0(x)\ge 0\}$ A pplying **Lemma 5.2** to **Theorem 5.1**, searching for piecewise differential positive definite function $B(t,x)$ could be formulated as a bilinear SOS programming problem which could be solved numerically.

Theorem 5.2: Given a hybrid automaton H with (c_1,c_2,T_H) under the same assumption of **Theorem 5.1**. Restrict initial states set, unsafe states set and invariant states set as

$$X_0=\{x\in X_0: p_0\ge 0\}, X_u=\{x\in X_u: p_u(x)\ge 0\}$$

and $X_{inv}=\{x\in I: p_{inv}(x)\ge 0\}$. Suppose there exist a piecewise positive definite polynomial $B(t,x)$ and a real number λ, a positive number ε, and vectors of SOS polynomials $s_0,s_u,s_{inv}\in\Sigma$. Then, barrier certificate $B(t,x)$ and the positive real number λ^* could be derived by solving the following bilinear SOS program:

$$c_1-B(x)-s_0 p_0(x)\in\Sigma \tag{29}$$

$$B(x)-c_2-\varepsilon-s_u p_u(x)\in\Sigma \tag{30}$$

$$-\dot{B}(t,x)+\lambda^* B(t,x)-s_{inv}p_{inv}(x)\in\Sigma \tag{31}$$

$$0<\lambda^*\le\lambda \tag{32}$$

$$\lambda<-\frac{1}{T_H}\ln\frac{c_1}{c_2} \tag{33}$$

$$\lambda^* t\le\lambda t \tag{34}$$

$$\forall t\in(0,T_H]: \frac{T^-(0,t)}{T^+(0,t)}\ge\frac{\lambda^+-\lambda^*}{\lambda^-+\lambda^*} \tag{35}$$

Proof: Here is a sketch of the proof. Reformulate each inequalities (22), (23) and (24) as the forms of (29), (30) and (30) through applying **Lemma 5.2** and relaxing positivity to an equivalent SOS polynomial, **Theorem 5.2** could be derived directly from **Theorem 5.1**.

Bilinear SOS program is hard to solve in general, however, with the help of numerical solvers such as SOSTOOLS or SOSOPT for MATLAB, $B(t,x)$ consisting of a sequence of $\{B_m(x)\}$ could be computed automatically. For more information on computational details, the readers are strongly encouraged to refer to [15], [16] or SOSOPT [17].

VI. Example

Consider the switched system described by

$$\begin{cases} \mathrm{mod}\,e1: \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = (2-0.5x_1^2)x_2 - x_1 \end{cases} \\ \mathrm{mod}\,e2: \begin{cases} \dot{x}_1 = -0.1x_1 + 10x_2 \\ \dot{x}_2 = -100x_1 - 5x_2 \end{cases} \end{cases}$$

With initial states $X_0 = \{1 \le x_1 \le 1.5, 0.5 \le x_2 \le 2\}$, unsafe states $X_u = \{(x_1+4)^2 + (x_2+2)^2 \le 4\}$. The software environment to test our method consists of SOSOPT and SeDuMi on MATLAB (R2012b) and monomials whose coefficient are less than 0.01 are omitted. The derived barrier certificate for transient safe mode 1 is as follows:

$$0.9x_1^4 + 0.09x_1^3x_2 + 0.18x_1^2x_2^2 + 0.01x_1x_2^3 + 0.01x_2^4 \quad (36)$$

With $\lambda_1 = 10.19$, and barrier certificate for permanently safe mode 2 is as follows:

$$0.01x_1^6 - 0.13x_1^4 - 0.07x_1^3x_2 + 0.84x_1^2 + 0.81x_1x_2 + \\ 0.2x_2^2 + 0.2x_1 + 0.01x_2 + 2.98 \quad (37)$$

With $\lambda_2 = 35.12$. Therefore, when $\frac{T^-(0,t)}{T^+(0,t)} \ge \frac{\lambda_2}{\lambda_1} = 3.5$, safety of H is guaranteed with respect to the barrier certificate consisting of polynomial (36) and (37). Fig. 2 presents the simulation, which indicates mode 1 is transient safe while mode 2 is permanently safe. Besides, $\frac{T^-(0,t)}{T^+(0,t)} \ge 3.5$ derived is conservative for guaranteeing safety of hybrid automata H.

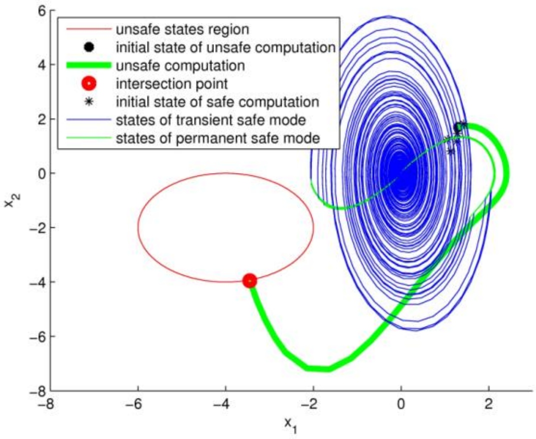

Fig. 2. 5 pieces of computation of the example, initial randomly in X_0 simulated over period [0,35].

Activating period of transient safe mode 1 is 2 seconds, while activating period of permanent safe mode 2 is 5 seconds. The bold green line shows the flow of mode 1 intersects unsafe states region at the 2.95 second indicating mode 1 is only transient safe. Therefore, the maximum activating period of mode 1 should be less than 2.95 seconds that permanently safe mode 2 should be activated. As indicated by $\frac{T^-(0,t)}{T^+(0,t)} \ge \frac{\lambda_2}{\lambda_1} = 3.5$, it's sufficient for permanently safe mode to be active lasting more than 10.33 seconds to guarantee the safety of the hybrid automata, however, as shown in the simulation, 5 seconds are already enough for guaranteeing safety of H. As a result, the activating ratio of permanently safe and transient safe mode derived is very sufficient for the safety of H.

VII. Conclusion

We have investigated the safety verification problem for a class of hybrid automata with both permanently safe modes and transient safe modes. These results are suitable to verify safety properties of time-critical hybrid automata whose mode switchings are under real-time scheduling. The most interesting improvement is the discussions of barrier certificates of hybrid automata with both permanently safe modes and transient safe modes. Through estimating exponential indexes of both permanently safe modes and transient safe modes, we are able to conduct quantitative analysis on safety of the computation of hybrid automata. And **Theorem 5.2** provides us computationally tractable means to determine the safety through solving a bilinear SOS programming problem. Considering the ubiquity and necessity of transient safe modes in real applications, we believe it is both meaningful and urgent to investigate formal safety verification method for this special kind of hybrid automata.

Acknowledgment

This paper is partially supported by the projects funded by the NSFC Trustworthy Software Track 91318301, NSFC Creative Team 61321064, NSFC Key Project 61332008, NSFC 61572195, and the National Natural Science Foundation of China (Grant No. 61021004). The Shanghai Trustworthy Computing Key Lab is supported by Shanghai Knowledge Service Platform ZF1213, NSFC Creative Team 61321064 and Shanghai Project (No.13511503100, No.012FU125X15, SHEITC160306).

References

[1] M. S. Branicky, V. S. Borkar, S. Mitter *et al.*, “A unified framework for hybrid control: B background, model, and theory,” 1994.

[2] R. Goebel, R. G. Sanfelice, and A. Teel, “Hybrid dynamical systems,” *Control Systems*, vol. 29, no. 2, pp. 28–93, 2009.

[3] T. Wongpiromsarn, U. Topcu, and A. G. Lamperski, “Automata theory meets barrier certificates: Temporal logic verification of nonlinear systems,” *CoRR*, vol. abs/1403.3524, 2014.

[4] R. G. Sanfelice, R. Goebel, and A. R. Teel, “Generalized solutions to hybrid dynamical systems,” *ESAIM: Control, Optimization and Calculus of Variations*, vol. 14, pp. 699–724, 2008.

[5] A. A. Julius and G. J. Pappas, “Trajectory based verification using local finite-time invariance,” *Hybrid Systems: Computation and Control*, pp. 223–236, Springer, 2009.

[6] P. A. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization," PhD thesis, California Institute of Technology, 2000.

[7] Y. Wang, X. Shi, Z. Zuo, M. Z. Chen, and Y. Shao, “On finite-time stability for nonlinear impulsive switched systems,” *Nonlinear Analysis: Real World Applications*, vol. 14, no. 1, pp. 807–814, 2013.

[8] H. Yazarel, S. Prajna, and G. J. Pappas, “Sos for safety,” in *Proc. 43rd IEEE Conference on Decision and Control Decision and Control*, vol. 1, pp. 461–466, IEEE, 2004.

[9] Z. Jarvis-Wloszek, R. Feeley, W. Tan, K. Sun, and A. Packard, "Control applications of sum of squares programming,” *Positive Polynomials in Control*, pp. 3–22, Springer, 2005.

[10] H. Kong, F. He, X. Song, W. N. Hung, and M. Gu, “Exponential-condition-based barrier certificate generation for safety verification of hybrid systems,” *Computer Aided Verification*, pp. 242–257, Springer, 2013.

[11] H. Kong, X. Song, D. Han, M. Gu, and J. Sun, “A new barrier certificate for safety verification of hybrid systems,” *The Computer Journal*, vol. 57, no. 7, pp. 1033–1045, 2014.

[12] G. Zhai, B. Hu, K. Yasuda, and A. N. Michel, “Stability analysis of switched systems with stable and unstable subsystems: An average dwell time approach,” *International Journal of Systems Science*, vol. 32, no. 8, pp. 1055–1061, 2001.

[13] S. Prajna and A. Jadbabaie, “Safety verification of hybrid systems using barrier certificates,” *Hybrid Systems: Computation and Control*, pp. 477–492, Springer, 2004.

[14] L. Dai, T. Gan, B. Xia, and N. Zhan, “Barrier certificates revisited,” 2013.

[15] S. Prajna, A. Papachristodoulou, P. Seiler, and P. A. Parrilo, “Sostools and its control applications,” *Positive Polynomials in Control*, pp. 273–292, Springer, 2005.

[16] A. Papachristodoulou, J. Anderson, G. Valmorbida, S. Prajna, P. Seiler, and P. Parrilo, “Sostools version 3.00 sum of squares optimization toolbox for matlab,” 2013.

[17] P. Seiler, “Sosopt: A toolbox for polynomial optimization,” 2013.

[18] R. Wisniewski and C. Sloth, “Converse barrier certificate theorems,” *IEEE Transactions on Automatic Control*, vol. 61, no. 5, pp. 1356–1361, 2016.

[19] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, “Robustness of control barrier functions for safety critical control,” *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.

[20] M. Z. Romdlony and B. Jayawardhana, “Stabilization with guaranteed safety using control lyapunov-barrier function,” *Automatica*, vol. 66, pp. 39–47, 2016.

[21] U. Borrmann, L. Wang, A. D. Ames, and M. Egerstedt, “Control barrier certificates for safe swarm behavior,” *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 68–73, 2015.

[22] T. A. Henzinger, "Verification of digital and hybrid systems," *The Theory of Hybrid Automata*, pp. 265–292. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000.

[23] P. Seiler and G. J. Balas, “Quasiconvex sum-of-squares programming,” in *Proc. the 9th IEEE Conference on Decision and Control Decision and Control*, pp. 3337–3342, 2010.

[24] S. M. Tabatabaeipour and M. Blanke, “Compositional finite-time stability analysis of nonlinear systems,” in *Proc. American Control Conference (ACC)*, 2014, pp. 1851–1858.

Guobin Wang was born in China, in 1985. Currently, he is a PhD candidate in the School of Computer Science and Software Engineering, East China Normal University. His current research interests include formal methods and safety verification of hybrid systems.

Jing Liu was born in China, in 1965. She is currently a professor of computer science and software engineering in the School of Computer Science and Software Engineering at East China Normal University, Shanghai, China. Her current research interests include formal methods, service-oriented architecture and model-driven architecture.