Exact safety verification of hybrid systems using sums-of-squares representation

LIN Wang¹², WU Min¹, YANG ZhengFeng¹∗ & ZENG ZhenBing¹

¹Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200062, China; ²College of Mathematics and Information Science, Wenzhou University, Zhejiang 325035, China

Received August 12, 2013; accepted September 24, 2013; published online October 31, 2013

Abstract In this paper we discuss how to generate inductive invariants for safety verification of hybrid systems. A hybrid symbolic-numeric method is presented to compute inequality inductive invariants of the given systems. A numerical invariant of the given system can be obtained by solving a parameterized polynomial optimization problem via sum-of-squares (SOS) relaxation. And a method based on Gauss-Newton refinement and rational vector recovery is used to obtain the invariants with rational coefficients, which exactly satisfy the conditions of invariants. Several examples are given to illustrate our algorithm.

Keywords Symbolic computation, semidefinite programming, safety verification, invariant generation

1 Introduction

Cyber-physical systems are systems in which the techniques of sensing, control, communication and coordination are involved and interacted with each other. Among complex physical systems, many are safety critical systems, such as airplanes, railway, and automotive applications. Due to the complexity, ensuring correct functioning of these systems, e.g., spatial separation, especially collision avoidance of aircrafts during the entire flights, is among the most challenging and most important problems in computer science, mathematics and engineering.

As a common mathematical model for complex physical systems, hybrid systems [1] are dynamical systems that are governed by interacting discrete and continuous dynamics [1,2]. Continuous dynamics is specified by differential equations and for discrete transitions, the hybrid system changes state instantaneously and possibly discontinuously. In addition to stability analysis [3,4], the verification of hybrid systems is also an important problem that has been studied extensively both by the control theory, and the formal verification community for over a decade. Among the most important verification questions for hybrid systems are those of safety, i.e., deciding whether a given property \( \psi \) holds in all the reachable states, and the dual of safety, i.e., reachability, deciding if there exists a trajectory starting from the initial set that reaches a state satisfying the given property \( \psi \). In principle, safety verification or reachability

∗Corresponding author (email: zfyang@sei.ecnu.edu.cn)
analysis aims to show that no trajectories of the hybrid systems starting from the initial set can enter some unsafe regions in the state space.

Due to the infinite number of possible states in continuous state spaces, safety verification or reachability analysis of hybrid systems presents a challenge. Some well-established techniques have been proposed. In [5], level set methods and flow-pipe approximations for computing approximate reachable sets of hybrid systems were presented. By contrast, quantifier elimination was used in [6] to compute exact reachable sets for linear systems with certain eigenstructures and semi-algebraic initial sets, and this method was generalized in [7] to handle linear systems with almost arbitrary eigenstructures. Recently, some methods [7–11] based on invariant generation have been proposed for safety verification of hybrid systems. An invariant [12] of a hybrid system is a property that holds in all the reachable states of the system, or in other words, it is an over-approximation of all the reachable states of the system. If the invariants lie inside the safe regions, or their intersection with the unsafe regions is empty, then safety of hybrid systems is verified.

The problem of generating invariants of an arbitrary form is known to be computationally hard, intractable even for the simplest classes. The usual technique for generating invariants is to produce an inductive invariant, i.e., an assertion that holds at the initial states of the system, and is preserved by all discrete and continuous state changes. There has been a considerable volume of work towards invariant generation for hybrid systems using techniques in convex optimization, semi-algebraic system solving [8–20]. However, some of these techniques are only applicable to linear systems, some are subject to numerical errors and some suffer from high complexity. In virtue of the efficiency of numerical computation and the error-free property of symbolic computation, we proposed in [21] a hybrid symbolic-numeric method via sum-of-squares (SOS) relaxation and exact certificate to construct inequality invariants for continuous dynamic systems given by nonlinear vector fields. Our work is partly motivated by the use of SOS relaxation for safety verification in [9,10]. The advantage of SOS relaxation lies in the fact that it eliminates the universal quantifiers in the conditions of the inductive invariants. However, Refs. [9,10] can only produce numerical invariants and therefore we apply further a symbolic certification, which comes from [22], to construct exact rational certificates from the numerical results of SDP solvers. In this paper, we generalize the above symbolic-numeric method to nonlinear hybrid systems. More precisely, we use SOS relaxation via semidefinite programming (SDP) and exact SOS representation recovery, to generate inequality invariants of hybrid systems, which guarantee that all the reachable states never enter the given unsafe regions. The idea is as follows: 1) Given a safe property, we predetermine the templates of the invariants, and construct a semidefinite programming (SDP) system to solve the corresponding parametric polynomial optimization problem. 2) An exact invariant is obtained by recovering the exact SOS representation from the approximate solution of the associated SDP system. In the recovery step, Gauss-Newton iteration is deployed to refine the approximate solution from SDP solver. Then safety property of the hybrid systems can be easily verified by the exact SOS representations of the conditions of the invariants. More details will be shown in Section 3.

Unlike the numerical approaches, our method can yield exact invariants, which can overcome the unsoundness in the verification of hybrid systems caused by numerical errors [23]. Our approach is more efficient and practical than some symbolic approaches of invariant generation based on quantifier elimination technique, because parametric polynomial optimization problem based on SOS relaxation with fixed size can be solved in polynomial time theoretically.

The rest of the paper is organized as follows. In Section 2, we introduce some notions about hybrid systems and invariants. Section 3 is devoted to illustrating a symbolic-numeric approach to generate invariants for safety verification of hybrid systems. In Section 4, we present some examples on invariant generation for safety verification of hybrid systems. Section 5 concludes the paper and discusses some future work.

2 Invariants

To model hybrid systems, we recall the definition of hybrid automata [1,11].
Definition 1 (Hybrid system). A hybrid system $H : (V, L, T, \Theta, D, \Psi, \ell_0)$ consists of the following components:

1) $V = \{x_1, \ldots, x_n\}$, a set of real-valued system variables. A state is an interpretation of $V$, assigning to each $x_i \in V$ a real value. An assertion is a first-order formula over $V$. A state $s$ satisfies an assertion $\varphi$, written as $s \models \varphi$, if $\varphi$ holds on the state $s$. We will also write $\varphi_1 \models \varphi_2$ for two assertions $\varphi_1, \varphi_2$ to denote that $\varphi_2$ is true at least in all the states in which $\varphi_1$ is true;

2) $L$, a finite set of locations;

3) $T$, a set of (discrete) transitions. Each transition $\tau : \langle \ell, \ell', g_\tau, \rho_\tau \rangle \in T$ consists of a prelocation $\ell \in L$, a postlocation $\ell' \in L$, the guard condition $g_\tau$ over $V$, and an assertion $\rho_\tau$ over $V \cup V'$ representing the next-state relation, where $V' = \{x'_1, \ldots, x'_n\}$ denotes the next-state variables. Note that the transition $\tau$ can take place only if $g_\tau$ holds;

4) $\Theta$, an assertion specifying the initial condition;

5) $D$, a map that maps each location $\ell \in L$ to a differential rule (also known as a vector field or a flow field) $D(\ell)$, of the form $\dot{x}_i = f_{\ell,i}(V)$ for each $x_i \in V$, written briefly as $\dot{x} = f_{\ell}(x)$. The differential rule at a location specifies how the system variables evolve in that location;

6) $\Psi$, a map that maps each location $\ell \in L$ to a location condition (location invariant) $\Psi(\ell)$, an assertion over $V$;

7) $\ell_0 \in L$, the initial condition. We assume that the initial condition satisfies the location invariant at the initial location, that is, $\Theta \models \Psi(\ell_0)$.

By a state of a hybrid system $H : (V, L, T, \Theta, D, \Psi, \ell_0)$, we mean the tuple $(\ell, x) \in L \times \mathbb{R}^n$ where $n$ is the number of program variables in $H$.

Figure 1 is a graphical representation of a hybrid system with two locations $\ell_1, \ell_2$. A state of this hybrid system is denoted by $(\ell, x) \in \{\ell_1, \ell_2\} \times \mathbb{R}^n$, and the initial state set is $\ell_1 \times \Theta$. During a continuous flow, the discrete location $\ell_i$ is maintained and the continuous state variables $x$ evolve according to the differential equations $\dot{x}_i = f_{\ell,i}(x)$, with $x$ satisfying the location invariant $\Psi(\ell_i)$. At the state $(\ell_i, x)$, if the guard condition $g(\ell_i, \ell_j)$ is met, the system may undergo a transition to location $\ell_j$, and $x$ will take the new value $x'$, which is determined by the reset map $\rho(\ell_i, \ell_j)$.

Given a hybrid system with an initial set and a prespecified safe (or unsafe) region, the system is safe if starting from any state in the initial set, this system would never evolve to the given unsafe region or the system would always stay inside the safe region. More specifically, consider the hybrid system $H$ shown in Figure 1 and let $X_u \subset \mathbb{R}^n$ be an unsafe region. The system $H$ is said to be safe if none of the trajectories of the system starting from any state in $(\ell_1, x_0) \in \ell_1 \times \Theta$ can reach $X_u$, or any state in $X_u$ is not reachable.

In this work, we will apply the invariant generation method to verify safety of hybrid systems. The following definitions of invariants of hybrid systems come from [11].

Definition 2 (Invariant). An invariant of a hybrid system at location $\ell$ is an assertion $I$ such that for any reachable state $(\ell, x)$ of the hybrid system, $x \models I$.

An invariant of a hybrid system is an assertion that holds in all the reachable states of the system.
Clearly, invariants are over-approximation of the reachable sets of hybrid systems, since an invariant is true for all the reachable states of the system. The problem of how to generate invariants with arbitrary form is known to be computationally hard, intractable even for the simplest classes. The usual technique for generating invariants is to compute inductive invariants, defined as follows.

**Definition 3** (Inductive invariant). An inductive assertion map $I$ of a hybrid system $H : \langle V, L, T, \Theta, D, \Psi, \ell_0 \rangle$ is a map that associates with each location $\ell \in L$ an assertion $I(\ell)$ that holds initially and is preserved by all discrete transitions and continuous flows of $H$. More formally, an inductive assertion map satisfies the following requirements:

1. **[Initial]**. $\Theta \models I(\ell_0)$.

2. **[Discrete consecution]**. For each discrete transition $\tau : (\ell, \ell', g_r, \rho_r)$, starting from a state satisfying $I(\ell)$, and taking $\tau$ leads to a state satisfying $I(\ell')$. Formally, $I(\ell) \land g_r \land \rho_r \models I(\ell')$, where $I(\ell')$ represents the assertion $I(\ell)$ with the current state variables $x_1, \ldots, x_n$ replaced by the next state variables $x_1', \ldots, x_n'$, respectively.

3. **[Continuous consecution]**. For every location $\ell \in L$ and states $\langle \ell, x_1 \rangle, \langle \ell, x_2 \rangle$ such that $x_2$ evolves from $x_1$ according to the differential rule $D(\ell)$ at $\ell$, if $x_1 \models I(\ell)$ then $x_2 \models I(\ell)$.

Our definition of inductive invariants is slightly modified from that of Definition 4 in [10], and the only change made is taking the guard conditions into account.

### 3 Safety verification of hybrid systems

The aim of this section is to translate the problem of safety verification of hybrid systems into that of generating invariants, which can be transformed further into polynomial optimization problem with parameters. We will present a hybrid symbolic-numeric method, based on SOS relaxation, to solve this polynomial optimization problem, and obtain the invariants, which can guarantee the safety property of hybrid systems.

#### 3.1 Invariants and safety verification

In this paper, we are interested in hybrid systems in which the relations are given by (real) polynomials over the system variables. Then we define

**Definition 4** (Polynomial hybrid system). A polynomial hybrid system is a hybrid system: $H : \langle V, L, T, \Theta, D, \Psi, \ell_0 \rangle$, where

1. for each transition $\tau : (\ell, \ell', g_r, \rho_r) \in T$, the guard condition $g_r$ (resp. the reset relation $\rho_r$) is a conjunction of polynomial inequalities over $V$ (resp. $V \cup V'$); also, the initial condition $\Theta$ and the location invariant $\Psi(\ell)$, for each $\ell \in L$, are conjunctions of polynomial inequalities over $V$;

2. each rule $D(\ell)$ is of the form $\dot{x}_i = f_{\ell,i}(x)$ for each $x_i \in V$, where $f_{\ell,i}(x) \in \mathbb{R}[x]$.

We are interested in finding inductive invariants of the form $\varphi(\ell, x) \geq 0$ at location $\ell \in L$. Below is an alternative form of Definition 3.

**Theorem 1.** Let $H : \langle V, L, T, \Theta, D, \Psi, \ell_0 \rangle$ be a hybrid system. Suppose for each location $\ell \in L$, there exists a function $\varphi(\ell, x)$ satisfying the following conditions: (i) $\Theta \models \varphi_0(x) \geq 0$, (ii) $\varphi(\ell, x) \geq 0 \land g(\ell, \ell') \land \rho(\ell, \ell') \models \varphi(\ell', x') \geq 0$, for any transition $\tau : (\ell, \ell', g, \rho)$ going out from $\ell$, (iii) $\varphi(\ell, x) \geq 0 \land \Psi(\ell) \models \varphi(\ell) > 0$, where $\varphi(\ell, x)$ denotes the Lie-derivative of $\varphi$ along the vector field $D(\ell)$, i.e., $\varphi(\ell, x) = \sum_{i=1}^{n} \frac{\partial \varphi}{\partial x_i} \cdot f_{\ell,i}(x)$. Then $\varphi(\ell, x) \geq 0$ is an inductive invariant of the hybrid system $H$ at location $\ell$.

**Proof.** The proof follows directly from Definition 3.

Theorem 1 will yield an inductive invariant $\varphi(x)$ for each location $\ell$. By changing the functions $\varphi(\ell, x)$ in Theorem 1 to a common function $\varphi(x)$, we can consider a unified inductive invariant for all locations. In the sequel, for brevity, we shall use $\varphi(x)$ to denote both the invariant $\varphi(x) \geq 0$ and $\varphi(x)$.
Note that the conditions in Theorem 1 is not complete for constructing inductive invariants; however invariants are easier to compute under this formulation. Please refer to [15] and [24] for two complete criterions, where higher order Lie derivatives are taken into account.

The following theorem shows that inductive invariants $\varphi_\ell(x)$ shown in Theorem 1 can be applied to verify the safety property of hybrid systems.

**Theorem 2.** Let $H$ be a hybrid system, and $X_u(\ell)$ be the unsafe region at location $\ell$. Suppose there exists functions $\varphi_\ell(x)$, for $\ell \in L$, that satisfy the conditions (i)–(iii) in Theorem 1, and moreover, (iv) $X_u(\ell) \models \varphi_\ell(x) < 0$, then the safety of the system $H$ is guaranteed.

**Proof.** Clearly, $\varphi_\ell(x) \geq 0$ is an invariant of hybrid system $H$ at location $\ell$. Then the condition (iv) implies that all reachable sets at location $\ell$ lie outside the unsafe region $X_u(\ell)$, yielding the safety of the system.

**Remark 1.** Functions $\varphi_\ell(x)$, or $\varphi(x)$, in Theorem 2 are also known as barrier certificates in [9,10].

### 3.2 Sum of squares relaxation

According to Theorem 2, to verify the safety of hybrid system $H$, it suffices to compute real polynomials $\varphi_\ell(x)$ or $\varphi(x)$. In the following, we only discuss how to find the invariant $\varphi_\ell(x)$ at each location $\ell \in L$. The problem of computing the inductive invariant $\varphi(x)$ can be handled similarly.

Our idea of computing $\varphi_\ell(x)$ or $\varphi(x)$, based on Sum-of-Squares (SOS) relaxation and rational vector recovery, is as follows.

**Step 1:** Predetermine a template of polynomial invariants with the given degree and convert the problem of computing polynomial invariants to the associated parametric polynomial optimization problem. SOS relaxation method is then applied to obtain a polynomial invariant with floating point coefficients.

**Step 2:** Apply Gauss-Newton refinement and rational vector recovery on the approximate polynomial invariant.

The problem of computing the inductive invariant $\varphi_\ell(x)$ at each location $\ell \in L$, that satisfy the conditions in Theorem 2 can be transformed into the following problem

\[
\begin{aligned}
\text{find } \varphi_\ell(x) &\in \mathbb{R}[x], \forall \ell \in L \\
\text{s.t. } &\Theta \models \varphi_{\ell_0}(x) \geq 0, \\
&\varphi_\ell(x) \geq 0 \land g(\ell, \ell') \land \rho(\ell, \ell') \models \varphi_\ell'(x') \geq 0, \\
&\varphi_\ell(x) \geq 0 \land \Psi(\ell) \models \varphi_\ell(x) > 0, \\
&X_u(\ell) \models \varphi_\ell(x) < 0.
\end{aligned}
\]

Let us first predetermine a template of polynomial invariants with the given degree $d$; that is, we assume

\[
\varphi_\ell(x) = \sum_{\alpha} c_\alpha x^\alpha,
\]

where $x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ and $c_\alpha \in \mathbb{R}$ are parameters, with $\alpha \in \mathbb{Z}_{\geq 0}^n$ and $\sum_{i=1}^n \alpha_i \leq d$. One can apply quantifier elimination methods to solve the corresponding parametric semi-algebraic systems, and for the given template, quantifier elimination methods can yield sufficient and necessary conditions for the existence of invariants. Several computer algebra tools, such as QEPCAD [25], RAGLib (see http://www.calfor.lip6.fr/~safety/RAGLib) and DISCOVERER [26], are available to solve this problem. However, quantifier elimination methods based on the cylindrical algebraic decomposition (CAD) are of high complexity. Instead, we will explore the SOS relaxation techniques based on semidefinite programming (SDP) solving to obtain polynomial invariants.

In the sequel, we suppose that $\Theta = \{x \in \mathbb{R}^n : \bigwedge_{i=1}^q \theta_i(x) \geq 0\}$, $X_u(\ell) = \{x \in \mathbb{R}^n : \bigwedge_{i=1}^s \chi_\ell_i(x) \geq 0\}$, $\Psi(\ell) = \{x \in \mathbb{R}^n : \bigwedge_{i=1}^r \psi_{\ell,i}(x) \geq 0\}$, $g(\ell, \ell') = \{x \in \mathbb{R}^n : \bigwedge_{i=1}^r g_{\ell,i}(x) \geq 0\}$, $\rho(\ell, \ell')(x, x') = \ldots$
\[\{x' \in \mathbb{R}^n : \bigwedge_{u=1}^{\ell'} \rho_{\ell',u}(x, x') \geq 0\}, \text{ where } \ell, \ell' \in L, \text{ and } \theta_i(x), \zeta_{\ell,j}(x), \psi_{\ell,k}(x), g_{\ell',i}(x) \text{ and } \rho_{\ell',u}(x, x') \text{ are polynomials.}\]

Clearly, if there exist SOS polynomials \(\sigma_i \in \mathbb{R}[x]\) for \(i = 0, \ldots, m\), such that \(g(x) \in \mathbb{R}[x]\) can be written as \(g(x) = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x)f_i(x)\), then the assertion \(\bigwedge_{i=1}^{m} (f_i(x) \geq 0) \models g(x) > 0\) holds. For more details the reader can refer to [27]. Based on the above observation, problem (1) can be transformed into an equivalent SOS programming of the form:

\[
\begin{align*}
\text{find} & \quad \varphi_{\ell}(x) \in \mathbb{R}[x], \forall \ell \in L \\
s.t. & \quad \varphi_{\ell}(x) = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x)\theta_i(x), \\
& \quad \varphi_{\ell}(x') = \lambda_{\ell,0}(x) + \sum_{i=1}^{m} \lambda_{\ell,i}(x)g_{\ell',i}(x) + \sum_{u=1}^{\ell'} \gamma_{\ell',u}(x)\rho_{\ell',u}(x, x') + \eta_{\ell'}(x)\varphi_{\ell}(x), \\
& \quad -\varphi_{\ell}(x) = \mu_{\ell,0}(x) + \sum_{j=1}^{p} \mu_{\ell,j}(x)\zeta_{\ell,j}(x) + \epsilon_{\ell,2},
\end{align*}
\]

where \(\sigma_i(x), \lambda_{\ell,i}(x), \gamma_{\ell',u}(x), \eta_{\ell'}(x), \mu_{\ell,j}(x)\) are SOSs in \(\mathbb{R}[x]\), \(\nu_{\ell}(x) \in \mathbb{R}[x]\), and \(\epsilon_{\ell,1}, \epsilon_{\ell,2} \in \mathbb{R}_+\). The decision variables are the coefficients of all polynomials appearing in (3), such as \(\varphi_{\ell}(x), \sigma_i(x), \lambda_{\ell,i}(x), \gamma_{\ell',u}(x)\). In practice, to avoid high computational complexity, we simply set up a truncated SOS programming by fixing an a priori (much smaller) degree bound 2\(e\), with \(e \in \mathbb{Z}_+\), of all the unknown polynomials.

Since the coefficients of \(\varphi_{\ell}(x), \eta_{\ell}(x)\) and \(\nu_{\ell}(x)\) are unknown, some nonlinear terms that are products of these coefficients occur in the second and third constraints of (3). The SOS relaxation will then lead to a non-convex bilinear matrix inequalities (BMI) problem, as illustrated in [9,10]. To avoid BMI problem, we adopt stronger conditions to compute the invariants of hybrid systems.

**Theorem 3.** Under the assumptions in Theorem 1, suppose that for each \(\ell \in L\), \(\varphi_{\ell}(x)\) satisfies the following conditions: (i) \(\Theta \models \varphi_{\ell}(x) \geq 0\), (ii') \(g(\ell, \ell', q, \rho) \models \varphi_{\ell}(x') \geq 0\), for any transition \((\ell, \ell', q, \rho)\) going out from \(\ell\), (iii') \(\Psi(\ell) \models \varphi_{\ell}(x) > 0\). Then \(\varphi_{\ell}(x) \geq 0\) is an inductive invariant of the hybrid system \(H\) at location \(\ell\). In addition, if \(\varphi_{\ell}(x)\) satisfies (iv) \(X_{\ell}(\ell) \models \varphi_{\ell}(x) < 0\), \(\forall \ell \in L\), then the safety of the system is guaranteed.

**Proof.** Since conditions (ii') and (iii') are stronger than conditions (ii) and (iii) in Theorem 1 respectively, \(\varphi_{\ell}\) is an invariant at location \(\ell\). According to Theorem 2, the condition (iv) can guarantee the safety of this system.

Note that although the constraints (ii') and (iii') in Theorem 3 are more conservative than those in Theorem 1, their implementations are more tractable.

Having Theorem 3, program (3) can be modified into the following problem:

\[
\begin{align*}
\text{find} & \quad \varphi_{\ell}(x) \in \mathbb{R}[x], \forall \ell \in L \\
s.t. & \quad \varphi_{\ell}(x) = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x)\theta_i(x), \\
& \quad \varphi_{\ell}(x') = \lambda_{\ell,0}(x) + \sum_{i=1}^{m} \lambda_{\ell,i}(x)g_{\ell',i}(x) + \sum_{u=1}^{\ell'} \gamma_{\ell',u}(x)\rho_{\ell',u}(x, x') + \epsilon_{\ell,1}, \\
& \quad -\varphi_{\ell}(x) = \mu_{\ell,0}(x) + \sum_{j=1}^{p} \mu_{\ell,j}(x)\zeta_{\ell,j}(x) + \epsilon_{\ell,2},
\end{align*}
\]

where \(\sigma_i(x), \lambda_{\ell,i}(x), \gamma_{\ell',u}(x), \mu_{\ell,j}(x)\) are SOSs in \(\mathbb{R}[x]\), and \(\epsilon_{\ell,1}, \epsilon_{\ell,2} \in \mathbb{R}_+\). The program is equivalent to the following SDP problem:

\[
\begin{align*}
\text{inf} & \quad \text{Tr}(M, W, V, P, Q) \\
s.t. & \quad \varphi_{\ell}(x) = m_0(x)^T \cdot M[0] \cdot m_0(x) + \sum_{i=1}^{m} m_i(x)^T \cdot M[i] \cdot m_i(x)\theta_i(x), \\
& \quad \varphi_{\ell}(x') = w_{\ell,0}(x)^T \cdot W[0] \cdot w_{\ell,0}(x) + \sum_{i=1}^{m} w_{\ell,i}(x)^T \cdot W[i] \cdot w_{\ell,i}(x)g_{\ell',i}(x) + \sum_{u=1}^{\ell'} w_{\ell',u}(x)^T \cdot V[u] \cdot w_{\ell',u}(x)\rho_{\ell',u}(x, x'), \\
& \quad \varphi_{\ell}(x) = -q_{\ell,0}(x)^T \cdot Q[0] \cdot q_{\ell,0}(x) - \sum_{j=1}^{p} q_{\ell,j}(x)^T \cdot Q[j] \cdot q_{\ell,j}(x)\zeta_{\ell,j}(x) + \epsilon_{\ell,2},
\end{align*}
\]
where all the matrices $M^l$, $W^{l',l}$, $V^{l',u}$, $P^{l,k}$, $Q^{l,j}$ are symmetric and positive semidefinite, and the function $\text{Tr}(M,W,V,P,Q)$ denotes the sum of traces of all these matrices.

In practice, $\text{Tr}(M,W,V,P,Q)$ acts as a dummy objective function commonly used in SDP solver for optimization problem with no objective function. Since problem (4) is purely a feasibility problem, here scalar-valued objective functions can also be used in problem (5). Usually, different objective functions will give different numerical feasible solutions to problem (5), as illustrated in Example 2.

Many Matlab packages of SDP solvers, such as SOSTOOLS [28], YALMIP [29], and SeDuMi [30], are available to solve problem (5) efficiently.

**Remark 2.** It may happen that the SDP solver yields no numerical feasible solutions to problem (5). In this case, we cannot conclude that there exists no invariants for the given hybrid system, which satisfy the conditions in Theorem 3 exactly.

### 3.3 Exact certificate of sum of squares decomposition

Since the SDP solvers in Matlab is running in fixed precision, the techniques in Subsection 3.2 will yield numerical solutions to the associated SDP problem (5), where the numerical polynomial $\varphi_l(x)$ and numerical positive semidefinite matrices $M^0, \ldots, Q^{l,j}$ satisfy the constraints in (5) approximately. For instance,

$$\varphi_l(x) \approx m_0(x)^T \cdot M^0 \cdot m_0(x) + \sum_{i=1}^q m_i(x)^T \cdot M^l \cdot m_i(x) \theta_l(x), \quad M^l \succeq 0. \quad (6)$$

However, due to round-off errors, $\varphi_l(x) \geq 0$ may not necessarily be an invariant of the given hybrid system at location $\ell$, because the constraints in (5) may not hold exactly, as illustrated by the following example.

**Example 1 ([9], page 31).** Consider the following nonlinear system:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -x_1 + \frac{1}{3}x_1^3 - x_2 \end{bmatrix},$$

we want to verify that all trajectories of the system starting from the initial set $\Theta$ will never enter the unsafe region $X_u$, where $\Theta = \{(x_1,x_2) \in \mathbb{R}^2 : (x_1 - 1.5)^2 + x_2^2 \leq 0.25\}$ and $X_u = \{(x_1,x_2) \in \mathbb{R}^2 : (x_1 + 1)^2 + (x_2 + 1)^2 \leq 0.16\}$. It suffices to find an invariant $\varphi(x_1,x_2)$ with rational coefficients, which satisfies all the conditions in (4). We can set up an SDP system using LMI constraints, and apply the SDP solver to find a numerical polynomial invariant $\varphi(x_1,x_2) = -1.3686 + 0.62499x_1^2 + 1.0669x_1x_2 + 1.5086x_2^2 - 0.56749x_1x_2^2 - 0.15231x_1^3 - 0.10417x_1^2 + 0.35564x_2^3 - 0.23739x_1^2x_2^2 - 0.234152x_1x_2^3$, and some associated numerical positive semidefinite matrices. However, $\varphi(x_1,x_2)$ cannot satisfy the conditions in Theorem 3 exactly, because there exists a sample point $p = (-127/64, -7/8)$ such that the condition (iii”) in Theorem 3 cannot be satisfied. Therefore, $\varphi(x_1,x_2)$ is not an exact invariant of this system.

Therefore in the next step, from the numerical polynomials $\varphi_l(x)$ and the numerical positive semidefinite matrices $M^0, \ldots, Q^{l,j}$, we will recover polynomials $\tilde{\varphi}_l(x)$ with rational coefficients, which satisfy (5) exactly.

As described in [22], finding a polynomial with rational coefficients can be translated into the problem of rational vector recovery. In Subsection 3.2, a numerical vector $v^*_j$ denoting the coefficient (column) vector of $\varphi_l(x)$ is obtained by solving the SDP system, i.e., $\varphi_l(x) = v^{*T}_j \cdot T(x)$, where $T(x)$ is the column vector of all terms in $\varphi_l(x)$. To obtain a rational vector $\tilde{v}_j$ near to $v^*_j$, we can employ the simultaneous Diophantine approximation algorithm [31], once the bound of the common denominator of $\tilde{v}_j$ is given.

The recovery of the matrices $M^0, \ldots, Q^{l,j}$ into rational positive semidefinite matrices is split into two steps. We first recover the matrices $\widetilde{M}^{l}, \ldots, \widetilde{Q}^{l,j}$ for $1 \leq l \leq q, \ldots, 1 \leq j \leq p$ and then recover $\widetilde{M}^{0}, \ldots, \widetilde{Q}^{0,j}$. To illustrate the idea, we only discuss how to recover $M^l$ for $1 \leq l \leq q$ and the matrices $W^{l',l}$, $V^{l',u}$, $P^{l,k}$, $Q^{l,j}$ can be recovered similarly.
Given the numerical positive semidefinite matrices $M[l]$, $1 \leq l \leq q$ in (5), we can find the nearby rational positive semidefinite matrices $\tilde{M}[l]$ by use of the rational vector recovery technique. In practice, all the $M[l]$ are numerical diagonal matrices; in other words, the off-diagonal entries are very tiny and the diagonal entries are nonnegative. Therefore, by setting the small entries of $M[l]$ to zeros we easily get the nearby rational positive semidefinite matrices $\tilde{M}[l]$ for $l = 1, \ldots, q$. The nearby rational positive semidefinite matrices $W[l,0]$, $V[l,0]$, $\tilde{P}[l,k]$, $Q[l,j]$ can be recovered similarly.

Having $\tilde{\varphi}_\ell(x) = \tilde{v}_\ell^T \cdot T_\ell(x)$ and $\tilde{M}[l], \ldots, \tilde{Q}[l,j]$ for $1 \leq l \leq q, 1 \leq j \leq p$, the program (5) is converted to

$$\begin{align*}
\text{inf} \quad & \text{Tr}(M[l]_0, W[l,0], P[l,0], Q[l,0]) \\
\text{s.t.} \quad & \tilde{\varphi}_\ell(x) - \sum_{i=1}^w m_1(x)^T \cdot \tilde{M}[l] \cdot m_1(x) \approx \tilde{m}_0(x)^T \cdot M[l]_0 \cdot \tilde{m}_0(x), \\
& \tilde{\varphi}_\ell(x) - \sum_{i=1}^w w[l,0](x)^T \cdot \tilde{W}[l,0] \cdot w[l,0](x) \\
& \quad \quad - \sum_{t=1}^s v[l,0](x)^T \cdot \tilde{V}[l,0] \cdot v[l,0](x) = \tilde{w}[l,0](x)^T \cdot W[l,0] \cdot \tilde{w}[l,0](x), \\
& \tilde{\varphi}_\ell(x) + \sum_{j=1}^s q[l,j](x)^T \cdot \tilde{Q}[l,j] \cdot q[l,j](x) \approx -q[l,0](x)^T \cdot Q[l,0] \cdot q[l,0](x). \\
\end{align*}$$

Observing (7), the matrices $M[l,0], \ldots, Q[l,0]$ have floating point entries, while the matrices $\tilde{M}[l,0], \ldots, \tilde{Q}[l,j]$ are rational positive semidefinite matrices. Therefore, the remaining task is to find nearby rational positive semidefinite matrices $\tilde{M}[l,0], \ldots, \tilde{Q}[l,j]$ such that the constraints in (7) hold exactly. To fulfill this task, we can first apply Gauss-Newton iteration to refine $M[l,0], \ldots, Q[l,0]$, and then recover the rational positive definite matrices $\tilde{M}[l,0], \ldots, \tilde{Q}[l,j]$ respectively from the refined $M[l,0], \ldots, Q[l,0]$, by orthogonal projection if the involved matrix is of full rank, or by rational vector recovery method otherwise.

Finally, we check if all the matrices $\tilde{M}[l,0], \ldots, \tilde{Q}[l,j]$ are positive semidefinite. If so, then return $\tilde{\varphi}_\ell(x) \geq 0$ as an invariant of the given hybrid system at location $\ell \in L$; otherwise, return “we cannot find invariants of the given degree bound”.

**Remark 3.** The above technique based on SOS relaxation and exact polynomial recovery can be applied to computing the inductive invariants of hybrid systems, which guarantee the safety of the given hybrid system.

### 3.4 Algorithm

The discussion in Subsection 3.3 leads to an algorithm of computing the (inductive) invariants of polynomial hybrid systems. As stated above, we only present how to compute the invariants $\varphi_\ell(x)$, for $\ell \in L$, that satisfy (5), and the case of computing the inductive invariants is similar.

**Algorithm** Polynomial inequality invariant generation

**Input:**
- $H : (V, L, T, \Theta, \Psi, \eta)$ a polynomial hybrid system.
- $d \in \mathbb{Z}_{>0}$: the degree bound of the candidate polynomial invariants.
- $D \in \mathbb{Z}_{>0}$: the bound of the common denominator of the coefficient vector of the polynomial invariants.
- $e \in \mathbb{Z}_{>0}$: the degree bound $2\epsilon$ of the SOSes used to construct the SDP system.
- $\tau \in \mathbb{R}_{>0}$: the given tolerance.

**Output:**
- $\tilde{\varphi}_\ell(x) \geq 0$: the verified polynomial invariant at each location $\ell \in L$.

1. Compute the candidates of polynomial invariants
   (i) For each location $\ell \in L$, predetermine the templates of $\varphi_\ell(x)$, with degree $d$, and construct an SDP system of form (5), where the degree bounds of all the involved SOSes are $2\epsilon$. If the SDP system (5) has no feasible solutions, return “we can’t find polynomial invariants with degree $\leq d$ at each location”; Otherwise, obtain a numerical vector $\eta^\tau$, numerical constants $\epsilon_{\ell,1}$, $\epsilon_{\ell,2}$ and numerical positive semidefinite matrices $M[l]$, $W[l,0]$, $V[l,0]$, $\tilde{P}[l,k]$, $Q[l,j]$ for $0 \leq l \leq q$, $0 \leq i < s$, $1 \leq u < t$, $0 \leq k \leq r$, $0 \leq j \leq p$.
   (ii) For the common denominator bound $D$, compute from $\eta^\tau$ a rational vector $\tilde{v}_\ell$ by Diophantine approximation algorithm, and get the associated rational polynomial $\tilde{\varphi}_\ell(x)$. Similarly, the nearby positive contacts $\epsilon_{\ell,1}$ and $\epsilon_{\ell,2}$ are obtained.
(iii) Convert all the $M^{[0]}, \ldots, Q^{[\ell,j]}$ into rational and positive semidefinite matrices $\tilde{M}^{[0]}, \ldots, \tilde{Q}^{[\ell,j]}$, for $1 \leq \ell \leq q, \ldots, 1 \leq j \leq p$.

2. Compute the exact SOS decomposition

   (i) Reconstruct an SDP system of form (7) to get approximate positive semidefinite matrices $M^{[0]}, \ldots, Q^{[\ell,0]}$ satisfying (7).

   (ii) Apply Gauss-Newton iteration to refine the matrices $M^{[0]}, \ldots, Q^{[\ell,0]}$ obtained in Step 2 (i).

   (iii) From the refined $M^{[0]}, \ldots, Q^{[\ell,0]}$, compute the rational matrices $\tilde{M}^{[0]}, \ldots, \tilde{Q}^{[\ell,0]}$ respectively by orthogonal projection method if the involved matrix is of full rank, or by rational vector recovery if the matrix is singular.

   (iv) Check whether all the matrices $\tilde{M}^{[0]}, \ldots, \tilde{Q}^{[\ell,0]}$ are positive semidefinite. If so, return $\tilde{\varphi}(x)$ as an invariant at location $\ell \in L$; otherwise, return “we cannot find polynomial invariants with degree $\leq d$.”

**Remark 4.** Our algorithm cannot guarantee that rational solutions will always be found since there exist limitations in the above algorithm on choosing the degree bound $e$ and the common denominator bound $D$. Furthermore, it is difficult to determine in advance whether there exist invariants with rational coefficients or not. Therefore, even if our algorithm cannot find the invariants, it does not mean that the given hybrid system has no invariants with the given degree bound $d$.

## 4 Experiments

In this section, some examples consider illustrate our method for safety verification of hybrid systems.

**Example 2** ([32], Example CLOCK). Consider a nonlinear continuous system

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -\frac{11}{2} y + y^2 \\ 6x - x^2 \end{bmatrix},$$

with location invariant $\Psi = \{(x, y) \in \mathbb{R}^2 : 1 \leq x \leq 5 \land 1 \leq y \leq 5\}$. The problem is how to verify that all trajectories of the system starting from the initial set $\Theta = \{(x, y) \in \mathbb{R}^2 : 4 \leq x \leq 4.5 \land y = 1\}$ will never reach the unsafe set $X_u = \{(x, y) \in \mathbb{R}^2 : 1 \leq x \leq 2 \land 2 \leq y \leq 3\}$. The safety of the continuous system can be verified if we can find a polynomial $\varphi(x, y)$ which satisfies conditions in Theorem 3. We rewrite $\Theta$, $X_u$, $\Psi$ as $\Theta = \{(x, y) \in \mathbb{R}^2 : \theta_1(x, y) \geq 0 \land \theta_2(x, y) \geq 0 \land \theta_3(x, y) \geq 0\}$, $\Psi = \{(x, y) \in \mathbb{R}^2 : \psi_1(x, y) \geq 0 \land \psi_2(x, y) \geq 0\}$, $X_u = \{(x, y) \in \mathbb{R}^2 : \zeta_1(x, y) \geq 0 \land \zeta_2(x, y) \geq 0\}$, where $\theta_1(x, y) = (4 - x)(x - 4.5)$, $\theta_2(x, y) = y - 1$, $\theta_3(x, y) = 1 - y$, $\psi_1(x, y) = (1 - x)(x - 5)$, $\psi_2(x, y) = (1 - y)(y - 5)$, $\zeta_1(x, y) = (1 - x)(x - 2)$, $\zeta_2(x, y) = (2 - y)(y - 3)$. Assuming $\deg(\varphi(x, y)) = d$, for $d = 1, 2, \ldots$ and the degree bound of all the involved SOSes in the program (4) is $2e = 10$, the SOS program (4) becomes $\varphi(x, y) = \sigma_0(x, y) + \sigma_1(x, y)\theta_1(x, y) + \sigma_2(x, y)\theta_2(x, y) + \sigma_3(x, y)\theta_3(x, y)$, $\varphi(x, y) = \phi_0(x, y) + \phi_1(x, y)\psi_1(x, y) + \phi_2(x, y)\psi_2(x, y) + \epsilon_1$, $- \varphi(x, y) = \mu_0(x, y) + \mu_1(x, y)\zeta_1(x, y) + \mu_2(x, y)\zeta_2(x, y) + \epsilon_2$, where $\sigma_0(x, y), \phi_1(x, y), \mu_0(x, y) \in \Sigma_{2,2e}, \epsilon_1, \epsilon_2 \in \mathbb{R}^+$.

We apply the algorithm in Subsection 3.4, and increase $d$ by 1 from 1 to 10 until a feasible solution of the SDP system is obtained. When $d = 4$, we obtain a feasible solution of the associated SDP system. Here we just list one approximate polynomial $\varphi(x, y) = -4.3296 - 1.2975x - 0.10418y + 0.92562x^2 + 0.18428xy + 0.35738y^2 + \cdots + 0.94032 \cdot 10^{-6}x^4 - 0.17047 \cdot 10^{-5}y^4$. Let the tolerance $\tau = 10^{-2}$, and the bound of the common denominator of the polynomial coefficients be 1000. By use of the rational SOS recovery technique described in Subsection 3.3, we obtain all the corresponding polynomials with rational coefficients, for instance, $\tilde{\varphi}(x, y) = -\frac{4113}{950} - \frac{4233}{950}x - \frac{99}{950}y + \frac{879}{950}x^2 + \frac{64}{950}y^2 + \frac{7}{78}xy - \frac{6}{785}xy^2 - \frac{46}{475}x^3$, which also guarantees the safety of the given system.
Example 3 ([32], Example ECO). Consider a predator-prey hybrid system depicted in Figure 2, where

\[ f_1(x) = f_2(x) = \begin{bmatrix} -x_1 + x_1 x_2 \\ x_2 - x_1 x_2 \end{bmatrix}. \]

The system starts in location \( \ell_1 \), with an initial state in \( \Theta = \{ (x_1, x_2) \in \mathbb{R}^2 : (x_1 - 0.8)^2 + (x_2 - 0.2)^2 \leq 0.01 \} \). Our task is to verify that the system never reaches the states in \( X_u(\ell_1) = \{ (x_1, x_2) \in \mathbb{R}^2 : 0.8 \leq x_1 \leq 0.9 \wedge 0.8 \leq x_2 \leq 0.9 \} \).

To verify the safety of this system, we need to find the corresponding invariant polynomials \( \varphi_1(x_1, x_2) \) and \( \varphi_2(x_1, x_2) \) at locations \( \ell_1 \) and \( \ell_2 \), respectively.

Similar to Example 2, we construct the associated SOS system, and find the feasible numerical solutions from SDP solver: \( \varphi_1 = 0.34871 - 0.45903 x_1 + 0.018001 x_2 + 0.2212 x_1^2 - 0.45764 x_1 x_2 + 0.17991 x_2^2 \), \( \varphi_2 = 0.011167 + 1.2891 x_1 + 0.56568 x_2 + 0.88855 x_1^2 - 0.56553 x_1 x_2 - 0.18386 x_2^2 \).

Let the tolerance \( \tau = 10^{-2} \), and the bound of the common denominator of the polynomial coefficients vector be 1000. By use of the rational SOS recovery technique , we obtain all the corresponding polynomials with rational coefficients. The invariant polynomials with rational coefficients are \( \tilde{\varphi}_1(x_1, x_2) = \frac{300}{127} x_1 + \frac{127}{300} x_2 - \frac{17}{300} x_1 x_2 + \frac{221}{127} x_1^2 + \frac{127}{221} x_2^2 \), \( \tilde{\varphi}_2(x_1, x_2) = \frac{481}{300} x_1 + \frac{221}{481} x_2 - \frac{481}{727} x_1 x_2 + \frac{300}{481} x_1^2 + \frac{481}{300} x_2^2 \). Furthermore, all the remaining related polynomials in (4) can be written as SOSes of the polynomials, which means \( \tilde{\varphi}_1 \) and \( \tilde{\varphi}_2 \) satisfy all the conditions in Theorem 3. So the safety of hybrid system is proven.

Example 4. Consider a hybrid system depicted in Figure 3, where

\[ f_1(x) = \begin{bmatrix} 2x_1 + x_2 \\ x_1 x_2 - x_2^2 - 1 \end{bmatrix}, \quad f_2(x) = \begin{bmatrix} x_2 \\ -x_1 + x_2 \end{bmatrix}. \]
The system starts in location $\ell_1$, with an initial state in $\Theta = \{(x_1, x_2) \in \mathbb{R}^2 : (x_1 - 1.5)^2 + x_2^2 \leq 0.25\}$. Our task is to verify that the system never reaches the states in $X_u(\ell_1) = \{(x_1, x_2) \in \mathbb{R}^2 : (x_1 + 1)^2 + (x_2 + 1)^2 \leq 0.16\}$.

To prove the safety of the hybrid system, it suffices to find an inductive invariant polynomial $\varphi(x_1, x_2)$ which satisfies all the conditions in Theorem 3.

Using the same techniques as illustrated in Examples 1 and 2, we obtain the inductive invariant polynomial with rational coefficients $\tilde{\varphi}(x_1, x_2) = -\frac{22}{49} + \frac{119}{931} x_1 - \frac{451}{931} x_2 + \frac{239}{931} x_1^2$. Moreover, $\tilde{\varphi}$ satisfies the conditions in Theorem 3 exactly. Therefore, the inductive invariant can guarantee the safety of the hybrid system. More details about the verification of conditions in Theorem 3, based on SOS representations of polynomials with rational coefficients, can be found in Appendix.

Note that all the computations have been performed on an Intel Pentium D 3.40 GHz processor with 1 GB of memory. For each example in this section, Algorithm 3.4 can yield the corresponding invariant within 15 s. We also apply the quantifier elimination method to the prescribed invariant template to verify the safety of these systems. When calling the computer algebra tool QEPCAD, we cannot obtain the result within 24 h.

5 Conclusion

In this paper, we present a symbolic-numeric approach to computing inequality invariants for safety verification of hybrid systems. Employing SOS relaxation and rational vector recovery techniques, it can be guaranteed that an exact invariant, rather than a numerical one, can be obtained efficiently and practically. This approach avoids both the weakness of numerical approaches to verifying safety of hybrid systems and the high complexity of symbolic invariant generation methods based on quantifier elimination.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (Grant Nos. 61021004, 91118007, 10801052, 10901055), Fundamental Research Funds for the Central Universities (Grant No. 78210043), NKBRPC (Grant No. 2011CB302802), and Scientific Research Project of The Graduate School of East China Normal University (Grant No. CX2011009). We thank the reviewers for their careful reading and useful comments that help us to improve the paper.

References

1. Henzinger T. The theory of hybrid automata. In: Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, 1996. 278–292
2. Alur R, Courcoubetis C, Halbwachs N, et al. The algorithmic analysis of hybrid systems. Theor Comput Sci, 1995, 1: 3–34
3. Wang Y, Wang W, Liu G. Stability of linear discrete switched systems with delays based on average dwell time method. Sci China Inf Sci, 2010, 53: 1216–1223
4. Feng W, Zhang J. Input-to-state stability of switched nonlinear systems. Sci China Inf Sci, 2008, 51: 1992–2004
5. Chutinan A, Krogh B H. Computational techniques for hybrid system verification. IEEE Trans Automat Contr, 2003, 48: 64–75
6. Lafferriere G, Pappas G J, Yovine S. Symbolic reachability computations for families of linear vector fields. J Symb Comput, 2001, 32: 231–253
7. Tiwari A. Approximate reachability for linear systems. In: Proceedings of the International Workshop on Hybrid Systems: Computation and Control, Prague, 2003. 514–525
8. Platzer A, Clarke E M. Computing differential invariants of hybrid systems as fixedpoints. Form Method Syst Des, 2009, 35: 98–120
9. Prajna S. Optimization-based methods for nonlinear and hybrid systems verification. Dissertation of Doctoral Degree. California Institute of Technology, 2005
Solution to Example 3:

Let the tolerance $\tau = 10^{-2}$, and the bound of the common denominator of the polynomial coefficients vector $\rho$ be $\rho$. The initial state $\Theta$, the unsafe region $X_0$, the state invariant $\Psi$, the guard condition $g$, and the reset map $r$ can be expressed as $\Theta = \{(x_1, x_2) \in \mathbb{R}^2 : \theta_1(x_1, x_2) \geq 0, \theta_2(x_1, x_2) \geq 0, \theta_3(x_1, x_2) \geq 0\}$, $\Psi(1) = \{(x_1, x_2) \in \mathbb{R}^2 : \psi_1(x_1, x_2) \geq 0, \psi_2(x_1, x_2) \geq 0, \psi_3(x_1, x_2) \geq 0\}$, $g(1) = \{(x_1, x_2) \in \mathbb{R}^2 : g_1(x_1, x_2) \geq 0, g_2(x_1, x_2) \geq 0\}$, and $r(1) = \{(x_1, x_2) \in \mathbb{R}^2 : r_1(x_1, x_2) \geq 0, r_2(x_1, x_2) \geq 0\}$.

The initial state $x_{1}(0)$, the state invariant $\Psi(\ell)$, the guard condition $g(\ell, \ell')$ and the reset map $r(\ell, \ell')$ can be expressed as $\Theta = \{(x_1, x_2) \in \mathbb{R}^2 \cup \{(x_1, x_2) \in \mathbb{R}^2 : \psi_1(x_1, x_2) \geq 0, \psi_2(x_1, x_2) \geq 0, \psi_3(x_1, x_2) \geq 0\}$, $\Psi(1) = \{(x_1, x_2) \in \mathbb{R}^2 : \psi_1(x_1, x_2) \geq 0, \psi_2(x_1, x_2) \geq 0, \psi_3(x_1, x_2) \geq 0\}$, $g(1) = \{(x_1, x_2) \in \mathbb{R}^2 : g_1(x_1, x_2) \geq 0, g_2(x_1, x_2) \geq 0\}$, and $r(1) = \{(x_1, x_2) \in \mathbb{R}^2 : r_1(x_1, x_2) \geq 0, r_2(x_1, x_2) \geq 0\}$.

The initial state $x_{1}(0)$, the state invariant $\Psi(\ell)$, the guard condition $g(\ell, \ell')$ and the reset map $r(\ell, \ell')$ can be expressed as $\Theta = \{(x_1, x_2) \in \mathbb{R}^2 : \theta_1(x_1, x_2) \geq 0, \theta_2(x_1, x_2) \geq 0, \theta_3(x_1, x_2) \geq 0\}$, $\Psi(1) = \{(x_1, x_2) \in \mathbb{R}^2 : \psi_1(x_1, x_2) \geq 0, \psi_2(x_1, x_2) \geq 0, \psi_3(x_1, x_2) \geq 0\}$, $g(1) = \{(x_1, x_2) \in \mathbb{R}^2 : g_1(x_1, x_2) \geq 0, g_2(x_1, x_2) \geq 0\}$, and $r(1) = \{(x_1, x_2) \in \mathbb{R}^2 : r_1(x_1, x_2) \geq 0, r_2(x_1, x_2) \geq 0\}$.

The initial state $x_{1}(0)$, the state invariant $\Psi(\ell)$, the guard condition $g(\ell, \ell')$ and the reset map $r(\ell, \ell')$ can be expressed as $\Theta = \{(x_1, x_2) \in \mathbb{R}^2 : \theta_1(x_1, x_2) \geq 0, \theta_2(x_1, x_2) \geq 0, \theta_3(x_1, x_2) \geq 0\}$, $\Psi(1) = \{(x_1, x_2) \in \mathbb{R}^2 : \psi_1(x_1, x_2) \geq 0, \psi_2(x_1, x_2) \geq 0, \psi_3(x_1, x_2) \geq 0\}$, $g(1) = \{(x_1, x_2) \in \mathbb{R}^2 : g_1(x_1, x_2) \geq 0, g_2(x_1, x_2) \geq 0\}$, and $r(1) = \{(x_1, x_2) \in \mathbb{R}^2 : r_1(x_1, x_2) \geq 0, r_2(x_1, x_2) \geq 0\}$.

The initial state $x_{1}(0)$, the state invariant $\Psi(\ell)$, the guard condition $g(\ell, \ell')$ and the reset map $r(\ell, \ell')$ can be expressed as $\Theta = \{(x_1, x_2) \in \mathbb{R}^2 : \theta_1(x_1, x_2) \geq 0, \theta_2(x_1, x_2) \geq 0, \theta_3(x_1, x_2) \geq 0\}$, $\Psi(1) = \{(x_1, x_2) \in \mathbb{R}^2 : \psi_1(x_1, x_2) \geq 0, \psi_2(x_1, x_2) \geq 0, \psi_3(x_1, x_2) \geq 0\}$, $g(1) = \{(x_1, x_2) \in \mathbb{R}^2 : g_1(x_1, x_2) \geq 0, g_2(x_1, x_2) \geq 0\}$, and $r(1) = \{(x_1, x_2) \in \mathbb{R}^2 : r_1(x_1, x_2) \geq 0, r_2(x_1, x_2) \geq 0\}$.

The initial state $x_{1}(0)$, the state invariant $\Psi(\ell)$, the guard condition $g(\ell, \ell')$ and the reset map $r(\ell, \ell')$ can be expressed as $\Theta = \{(x_1, x_2) \in \mathbb{R}^2 : \theta_1(x_1, x_2) \geq 0, \theta_2(x_1, x_2) \geq 0, \theta_3(x_1, x_2) \geq 0\}$, $\Psi(1) = \{(x_1, x_2) \in \mathbb{R}^2 : \psi_1(x_1, x_2) \geq 0, \psi_2(x_1, x_2) \geq 0, \psi_3(x_1, x_2) \geq 0\}$, $g(1) = \{(x_1, x_2) \in \mathbb{R}^2 : g_1(x_1, x_2) \geq 0, g_2(x_1, x_2) \geq 0\}$, and $r(1) = \{(x_1, x_2) \in \mathbb{R}^2 : r_1(x_1, x_2) \geq 0, r_2(x_1, x_2) \geq 0\}$.
be 1000. Then we can find that the inductive invariant polynomial \( \overline{\psi}(x_1, x_2) \) satisfies \( \overline{\psi}(x_1, x_2) = \overline{\sigma}(x_1, x_2) + \overline{\sigma}_1(x_1, x_2) \theta_1(x_1, x_2) \), \( \overline{\psi}(x_1, x_2) = \lambda_{120}(x_1, x_2) + \lambda_{121}(x_1, x_2) \phi_{10}(x_1, x_2) + \gamma_{12}(x_1, x_2) \phi_1(x_1, x_2) \), \( \overline{\psi}(x_1, x_2) = \lambda_{120}(x_1, x_2) + \lambda_{121}(x_1, x_2) p_2(x_1, x_2) + \gamma_{12}(x_1, x_2) p_1(x_1, x_2) \), \( \overline{\psi}(x_1, x_2) = \phi_{10}(x_1, x_2) + \phi_{11}(x_1, x_2) \phi_{11}(x_1, x_2) + \phi_{12}(x_1, x_2) + \phi_2(x_1, x_2) + \phi_2(x_1, x_2) + \epsilon_1 \), where \( \overline{\psi}(x_1, x_2) = \overline{\sigma}(x_1, x_2) + \overline{\sigma}_1(x_1, x_2) \theta_1(x_1, x_2) + \overline{\sigma}_2(x_1, x_2) + \epsilon_2 \).