J Syst Sci Complex (20XX) XX: 1–19

GENERATING SEMI-ALGEBRAIC INVARIANTS FOR NON-AUTONOMOUS POLYNOMIAL HYBRID SYSTEMS

WANG Qiuye · LI Yangjia · XIA Bican · ZHAN Naijun

DOI:
Received: x x 20xx / Revised: x x 20xx

DOI:
Received: x x 20xx / Revised: x x 20xx
©The Editorial Office of JSSC & Springer-Verlag Berlin Heidelberg 2013

Abstract Hybrid systems are dynamic systems with interacting discrete computation and continuous physical processes, which have become more common, more indispensable, and more complicated in our modern life. Particularly, many of them are safety-critical, and therefore are required to meet a critical safety standard. Invariant generation plays a central role in the verification and synthesis of hybrid systems. In our previous work, we gave a necessary and sufficient condition for a semi-algebraic set being an invariant of a polynomial autonomous dynamical system, which gave a confirmative answer to the open problem. In addition, based on which a complete algorithm for generating all semi-algebraic invariants of a given polynomial autonomous hybrid system with the given shape was proposed. In this paper, we consider how to extend our previous work to non-autonomous dynamical and hybrid systems. Non-autonomous dynamical and hybrid systems are with inputs, which are very common in practice; in contrast, autonomous ones are without inputs. Furthermore, we present a sound and complete algorithm to verify semi-algebraic invariants for non-autonomous polynomial hybrid systems. Based

WANG Qiuye, Yangjia Li and Naijun Zhan
State Key Lab. of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China
E-mail : {wangqye,yangjia,znj}@ios.ac.cn
XIA Bican
LMAM & School of Math. Sci., Peking University, Beijing, China
E-mail : xbc@math.pku.edu.cn
* The first, second and fourth authors are supported partly by "973 Program" under grant No. 2014CB340701, by NSFC under grants 91418204 and 61625206, by CDZ project CAP (GZ 1023), and by the CAS/SAFEA International Partnership Program for Creative Research Teams; the third author is supported partly by NSFC under grants 11290141, 11271034 and 61532019.
◇ *This paper was recommended for publication by Editor ....*

on which, we propose a sound and complete algorithm to generate all invariants with a pre-defined template.

Keywords Hybrid Systems, polynomial ideals, semi-algebraic sets, non-autonomous systems, invariants

1 Introduction

Hybrid systems (HSs), also known as cyber-physical systems nowadays, are dynamic systems with interacting discrete computation and continuous physical processes, which have become more common, more indispensable, and more complicated in our modern life. Many hybrid systems in real applications, such as automotive, aerospace, and medical systems, are safety-crucial, and therefore are required to meet a rigid safety standard. In HSs, the physical processes evolve continuously with respect to time, and the discrete computers monitor and control the physical processes, to meet the safety requirement. The correct functionality of the control from the controllers is essential to guarantee the safety of HSs. In the literature, this issue has been studied extensively through system verification or controller synthesis. For an HS with a given controller, the verification of the HS, i.e., whether under the given controller the HS can achieve the desired safety requirement, can be done either through model-checking mainly depending on reachability computation [1–6] or through deductive way mainly depending on invariant generation [7–12]. As an alternative, given an incomplete HS and a specification, one can synthesize a correct controller which ensures that the given specification is satisfied by the system by restricting its behaviour. There are many approaches proposed for controller synthesis for HSs, e.g., [13–18], most of which essentially depend on invariant generation also.

So, the concept of *invariant* plays a central role in the verification and controller synthesis of HSs. An *invariant* of an HS is a property ϕ that holds in all the reachable states of the system. An *inductive invariant* of an HS is an assertion ϕ that holds at the initial states of the system, and preserved by all discrete and continuous state changes. Any inductive invariant is also an invariant. The key issue in generating and verifying inductive invariants of an HS is to deal with continuous dynamics, i.e. to construct and verify so-called *continuous invariant* of the continuous dynamics at each mode of the system. A mode of an HS is usually represented by a *constrained continuous dynamical system* (CCDS for short) of the form $(D, \mathbf{f})$, where $\mathbf{f}$ is a vector field and D is a domain restriction of continuous evolution. A property φ is called a *continuous invariant* (CI for short) [10, 11] of $(D, \mathbf{f})$, if for any trajectory ω starting from D, φ is satisfied along ω as long as ω still remains in the restricted domain D. An inductive invariant of an HS consists of a set of CIs such that the initial condition of the system entails the CI of the initial mode, and if there is a discrete transition between two modes of the system then the CI at the pre-mode implies the CI at the post-mode with respect to the discrete transition. We will call an inductive invariant of an HS a *global inductive invariant*, or just *global invariant* (GI for short) to distinguish from continuous invariants.

The problem of (inductive) invariant generation and verification have received wide attention in the analysis of programs [19–22] and hybrid systems [7, 8, 10, 11, 15, 23–27]. The

basic idea behind most of these approaches is as follows: first, predefine a property template (linear or non-linear, depending on the property to be verified); then, encode the conditions of a property to be inductive (discretely and/or continuously) into some constraints on state variables and parameters; finally, find out solutions to the constraints by symbolic computation, or numeric computation, or their combination. So, how to define inductive conditions and the power of constraint solving are essential to these approaches. In [28, 29], the authors independently proposed different approaches for constructing inductive invariants for linear HSs. Sankaranarayanan et al. presented a computational method to automatically generate algebraic invariants for algebraic HSs in [30], based on the theory of pseudo-ideals over polynomial rings and quantifier elimination. Prajna *et al.* in [31, 32] provided a new notion of inductive invariants called *barrier certificates* (BC) for verifying the safety of semi-algebraic HSs using the technique of sum-of-squares (SOS). Their approach was further extended by Kong *et al.* in [33] and by Dai *et al.* in [12] by considering how to relax the condition of BCs, but remain the condition to be convex so that synthesizing BCs can still be done using SDP efficiently. In [10], Platzer and Clarke extended the idea of BCs by considering Boolean combinations of multiple polynomial inequalities. In [9, 15], Tiwari *et al.* investigated how to generate inductive invariants with more expressiveness for semi-algebraic HSs through relaxing the inductive conditions by considering inductiveness only on the boundaries of predefined invariant templates. While in [11], Liu *et al.* considered how to further relax the inductive condition given in [9, 15] and established a first finite complete inductive condition to determine if a polynomial formula is an invariant for a given semi-algebraic HS, which gave a confirmative answer to an open problem proposed in [15]. In [34], Sloth *et al.* proposed an approach to constructing global inductive invariants from local differential invariants using optimization techniques.

However, most of existing approaches are only applicable to autonomous dynamical and hybrid systems, i.e., without inputs, but in practice, dynamical and hybrid systems with inputs (i.e., non-autonomous) are quite common. A simple solution to non-autonomous case is to reduce to autonomous case by treating time as a normal variable. But the disadvantages of this approach are obvious, including:

Scalability: Some invariants of some dynamical systems can not be synthesized by homogenizing, which is similar to stability analysis of dynamical systems.

Sucinctness: Invariants synthesized using homogenizing normally contain time, which is not what we expected.

Efficiency: This approach is not so efficient in general.

In this paper, we give a different approach to this issue by extending Liu *et al.*'s approach in another way. The basic idea behind our approach is as follows: We treat inputs as parameters, and extend Liu *et al.*'s approach to parametric case. That is, for any instantiation of the inputs, we construct an ascending chain of polynomial ideals obtained from higher-order Lie-derivatives of the polynomials occurring in the invariant template w.r.t. the considered vector fields (the details will be discussed later); Then, exploiting the results given in [12, 35, 36], we

can compute a uniform bound on the length of the family of ascending chains of parametric polynomial ideals; Thus, based on which, we can obtain a necessary and sufficient condition to verify whether a given semi-algebraic set is an invariant of a considered non-autonomous dynamical and hybrid systems. Furthermore, we can easily design an algorithm to synthesize all semi-algebraic invariants with a given shape for a considered non-autonomous dynamical and hybrid systems similar to [11]. In addition, an explicit complexity analysis of our approach is given, while the complexity of Liu *et al.*'s approach remains unknown. The advantages of our approach include:

Scalability: it can be applied to any kind of non-autonomous dynamical and hybrid systems.

Efficiency: we do not need to consider time.

Succinctness: The generated invariants do not contain time, therefore more succinct.

The rest of this paper is organized as follows: Section 2 introduces basic definitions and theories on which our approach is developed; Section 3 discusses our method in a simple case when initial set, domain and invariant are all defined by a single polynomial; complexity analysis and some experimental results are reported in Section 4; we discuss how to generalize the simple case to the general case in Section 5, and conclude this paper in Section 6.

2 Preliminaries

In this section, we introduce some basic notions and theories, based on which our approach is developed.

In what follows, we will use $\mathbb{R}$ to stand for the set of real numbers, $\mathbf{x}$ for the variable vector $(x_1, \cdots, x_n)$, $\mathbb{K}[\mathbf{x}]$ for the polynomial ring in variables $\mathbf{x}$ with coefficients in $\mathbb{K}$.

2.1 Semi-algebraic Sets

A semi-algebraic set is a subset of $\mathbb{R}^n$ defined by a conjunction of finitely many polynomial equations and inequalities , or a union of finitely many these sets. Formally,

Definition 2.1 (Semi-algebraic Set) A set $S \subseteq \mathbb{R}^n$ is a semi-algebraic set if there exists formula ψ such that $S = S(\psi) = \{\mathbf{x} \in \mathbb{R}^n \mid \psi(\mathbf{x}) \text{ satisfies}\}$, where ψ is of the form

$$\psi \doteq \bigvee_{k=1}^{K} \bigwedge_{j=1}^{J_k} p_{kj} \rhd 0$$

where $p_{kj} \in \mathbb{R}[\mathbf{x}]$ and $\rhd \in \{\geq, >\}$.

It is easy to prove that there is a correspondence between the operations on semi-algebraic sets and their counterparts (i.e., the logical connectives) on formulas, i.e.,

- $S(\psi_1) \cap S(\psi_2) = S(\psi_1 \wedge \psi_2)$
- $S(\psi_1) \cup S(\psi_2) = S(\psi_1 \vee \psi_2)$
- $S(\psi)^c = S(\neg\psi)$

- $S(\psi_1)\backslash S(\psi_2) = S(\psi_1) \cap S(\psi_2)^c = S(\psi_1 \wedge \neg\psi_2)$

2.2 Hybrid Systems

As discussed in the introduction section, the hardest part in the invariant generation for HSs is how to deal with continuous dynamics, i.e., CI generation. If we know how to verify and/or generate invariant for HSs with one mode, i.e., constrained dynamical systems, it can be easily extended to HSs with multiple modes, e.g., see [37, 38]. So for simplicity, we only concentrate on HSs with one mode throughout this paper. Therefore, all HSs are referred to HS with one mode hereafter if not otherwise stated.

Definition 2.2 (Hybrid System (HS)) An HS is a tuple $(\boldsymbol{H}, \boldsymbol{I}, \mathbf{f})$, where:

(i) $\boldsymbol{H} \subseteq \mathbb{R}^n$ is the domain of system state;

(ii) $\boldsymbol{I} \subseteq \boldsymbol{H}$ is the set of initial states, which will be called the initial set; and

(iii) $\mathbf{f} \in \boldsymbol{H} \times [0, +\infty) \to \mathbb{R}^n$ is the evolution function of system. The evolution is subject to: $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t)$.

An HS is called *autonomous* if $\mathbf{f}$ is independent of time t, i.e., without inputs; otherwise, called *non-autonomous*.

Definition 2.3 (Trajectory) For an HS M, the trajectory originating from state $\mathbf{x}_0$ and time t_0 in time T is a differentiable function $\mathbf{x} \in [t_0, t_0 + T) \to \mathbb{R}^n$ satisfying

(i) for $t \in [t_0, t_0 + T)$, $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t)$; and

(ii) $\mathbf{x}(t_0) = \mathbf{x}_0$.

When it is clear from context, we will omit the phrase "from $\mathbf{x}_0$", "from time t_0".

Definition 2.4 (Safety Problem) Given an unsafe set $X_U \subseteq \mathbb{R}^n$, we say an HS is safe, if no trajectories satisfy:

(i) $\mathbf{x}(0) \in \boldsymbol{I}$; and

(ii) $\exists \tau \geq 0 : \mathbf{x}(\tau) \in X_U$.

Definition 2.5 (Continuous Invariant) The continuous invariant (CI) of an HS $(\boldsymbol{H}, \boldsymbol{I}, \mathbf{f})$ is a set $\boldsymbol{A} \subseteq \mathbb{R}^n$ satisfying

(i) **Initial Condition**: $\boldsymbol{I} \subseteq \boldsymbol{A}$, and

(ii) **Continuous Inductiveness**: for any $t_0 \geq 0$, $\mathbf{x}_0 \in \boldsymbol{A}$ and $T > 0$, any trajectory $\mathbf{x}$ with $\mathbf{x}(t_0) = \mathbf{x}_0$ guarantees that

$$(\forall t \in [t_0, t_0 + T], \mathbf{x}(t) \in \boldsymbol{H}) \Rightarrow (\forall t \in [t_0, t_0 + T], \mathbf{x}(t) \in \boldsymbol{A}).$$

Hereafter, the phase *invariant* always refers to CI if not otherwise stated.

Definition 2.6 (Polynomial Hybrid System (PHS)) An HS $(\boldsymbol{H}, \boldsymbol{I}, \mathbf{f})$ is polynomial, if

(i) domain $\boldsymbol{H}$ is a semi-algebraic set;

(ii) initial set $\boldsymbol{I}$ is a semi-algebraic set; and

(iii) vector field $\mathbf{f}$ is polynomial, i.e., $f_i(\mathbf{x}, t) \in \mathbb{R}[\mathbf{x}, t]$ for every $f_i(\mathbf{x}, t)$ in $\mathbf{f}(\mathbf{x}, t)$ where $\mathbb{R}[\mathbf{x}, t]$ is an abbreviation of $\mathbb{R}[x_1, x_2, \ldots, x_n, t]$ and $\mathbf{f}(\mathbf{x}, t) = (f_1(\mathbf{x}, t), f_2(\mathbf{x}, t), \ldots, f_n(\mathbf{x}, t))$.

Thus, in what follows, we always assume all HSs are polynomial, if not otherwise stated.

It is easy to verify that all polynomials satisfy *local Lipschitz condition*, therefore, by the famous Picard-Lindelöf theorem, there exists a unique solution to the initial value problem locally for any given polynomial HS.

2.3 Lie Derivatives

Lie derivatives describe the change of a tensor field along the flow of a vector field.

Definition 2.7 (Lie Derivatives) For a given function $\theta(\mathbf{x})$, a vector field $\mathbf{f}(t, \mathbf{x})$, we define Lie derivatives $\mathcal{L}^k_{t,\mathbf{f}}(\theta(\mathbf{x}))$ inductively as follows

$$\begin{aligned}
&\mathcal{L}^0_{t,\mathbf{f}}\theta(\mathbf{x}) = \theta(\mathbf{x}) \\
&\mathcal{L}^k_{t,\mathbf{f}}\theta(\mathbf{x}) = \langle \frac{\partial}{\partial \mathbf{x}} \mathcal{L}^{k-1}_{t,\mathbf{f}}\theta(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}, t) \rangle, \qquad k > 0
\end{aligned}$$

where $\langle * \cdot * \rangle$ is dot product.

Example 2.8 Assume $\mathbf{f}(x, y, t) = (-x + t, y - t)$ and $\theta(x, y) = x + y^2$, then:

$$\begin{aligned}
&\mathcal{L}^0_{t,\mathbf{f}}\theta(x, y) = x + y^2 \\
&\mathcal{L}^1_{t,\mathbf{f}}\theta(x, y) = 2y^2 - x - 2ty + t \\
&\mathcal{L}^2_{t,\mathbf{f}}\theta(x, y) = 4y^2 + x - 6ty + 2t^2 - t \\
&\qquad\qquad \cdots\cdots
\end{aligned}$$

Definition 2.9 (Pointwise Rank) For a given state $\mathbf{x}$ and time t, we define pointwise rank of Lie derivatives as:

$$\gamma_{\mathbf{f},\theta}(\mathbf{x}, t) = \min\{k \in \mathbb{N} \mid \mathcal{L}^k_{t,\mathbf{f}}\theta(\mathbf{x}) \neq 0\}$$

Note that Lie derivatives and their pointwise rank both are functions in variable t.

2.4 Polynomial Ideal Theories

Polynomial ideal theories play an important role in our approach, so, let's recall some basic notions and results on polynomial ideal theories. Please refer to [39] for the detail.

Definition 2.10 (Polynomial Ideals) A subset $\boldsymbol{I} \subseteq \mathbb{R}[\mathbf{x}]$ is called *ideal* if

(i) $0 \in \boldsymbol{I}$;

(ii) for any $p(\mathbf{x}), q(\mathbf{x}) \in \boldsymbol{I}$, then $p(\mathbf{x}) + q(\mathbf{x}) \in \boldsymbol{I}$; and

(iii) for any $p(\mathbf{x}) \in \boldsymbol{I}$ and $h(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$, then $p(\mathbf{x})h(\mathbf{x}) = h(\mathbf{x})p(\mathbf{x}) \in \boldsymbol{I}$.

Definition 2.11 (Basis of an Ideal) we call

$$\boldsymbol{I} \doteq \bigcap_{p_1,p_2,\ldots,p_k \in \boldsymbol{I}', \boldsymbol{I}' \text{is an ideal}} \boldsymbol{I}'$$

an ideal generated by $p_1, p_2, \ldots, p_k$, denoted by $\langle p_1, p_2, \ldots, p_k \rangle$. $\{p_1, p_2, \ldots, p_k\}$ is called a *basis* (*generator*) of I.

It is easy to prove that

$$\langle p_1, p_2, \ldots, p_k \rangle = \{\sum_{i=1}^{k} p_i h_i \mid \forall i : h_i \in \mathbb{R}[\mathbf{x}]\}.$$

A well-known result on polynomial ideals is that every polynomial ideal of $\mathbb{R}[\mathbf{x}]$ can be generated by a finite basis, i.e.,

Theorem 2.12 (Hilbert's Basis Theorem) *Any polynomial ideal $\boldsymbol{I} \subseteq \mathbb{R}[\mathbf{x}]$ can be generated by a finite basis. That is, for any ideal $\boldsymbol{I} \subseteq \mathbb{R}[\mathbf{x}]$, there exist $p_1, p_2, \ldots, p_k \in \mathbb{R}[\mathbf{x}]$ such that $\boldsymbol{I} = \langle p_1, p_2, \ldots, p_k \rangle$.*

A corollary of the above theorem is:

Theorem 2.13 (Ascending Chain Theorem) *For any infinite ascending chain of polynomial ideals of $\mathbb{R}[\mathbf{x}]$ $\boldsymbol{I}_1 \subseteq \boldsymbol{I}_2 \subseteq \cdots \boldsymbol{I}_k \subseteq \cdots$, there exists N such that for any $k \geq N$, $\boldsymbol{I}_k = \boldsymbol{I}_N$.*

3 Invariant Generation in Simple Case

In this section, we focus on a simplified version of the original problem in order to explain the basic idea behind our approach. More specifically, we only consider invariants that is represented by a single polynomial inequality $\theta(\mathbf{x}) \geq 0$, denoted by Φ. Also, we assume that the domain $\boldsymbol{H}$ and initial set $\boldsymbol{I}$ can be defined by a single polynomial inequality. We will use $\mathbb{H}(\mathbf{x}) \geq 0$ and $\mathbb{I}(\mathbf{x}) \geq 0$ to denote them respectively.

3.1 Basic Idea

Intuitively, an invariant is such a set that at any time if the system state falls inside it, then the trajectory starting from the state will never go out of the set if the trajectory stays inside the domain. Obviously, when a system state is inside Φ, the trajectory starting from the point must reach the boundary of Φ before it goes out of Φ because of the continuity of the trajectory. So the invariant of Φ can only be violated at the boundary of Φ. That is to say, in this simple case, Φ is an invariant if and only if, for every state $\mathbf{x}_0$ and time t_0, if $\theta(\mathbf{x}_0) = 0$ with $\mathbf{x}_0 = \mathbf{x}(t_0)$, then there exists a constant $\varepsilon > 0$ such that the trajectory originating from state $\mathbf{x}_0$ at time t_0 keeps $\theta(\mathbf{x}(t))$ nonnegative for all $t \in [t_0, t_0 + \varepsilon)$.

Theorem 3.1 *Given a polynomial θ and an HS $(\boldsymbol{H}, \boldsymbol{I}, \mathbf{f})$, $\gamma_{\mathbf{f},\theta}(\mathbf{x}_0, t) \neq 0$ if and only if $\mathbf{x}_0 \in S(\theta(\mathbf{x}) = 0)$, where $S(\theta(\mathbf{x}) = 0)$ stands for the set of solutions of equation $\theta(\mathbf{x}) = 0$. Moreover, let $\mathbf{x}(t_0) = \mathbf{x}_0$, then*

(i) if $\gamma_{\mathbf{f},\theta}(\mathbf{x}_0,t_0)<\infty$ *and* $\mathcal{L}_{t_0,\mathbf{f}}^{\gamma_{\mathbf{f},\theta}(\mathbf{x}_0,t_0)}\theta(\mathbf{x}_0)>0$*, then* $\exists\varepsilon>0,\forall t\in(t_0,t_0+\varepsilon),\theta(\mathbf{x}(t))>0$*;*

(ii) if $\gamma_{\mathbf{f},\theta}(\mathbf{x}_0,t_0)<\infty$ *and* $\mathcal{L}_{t_0,\mathbf{f}}^{\gamma_{\mathbf{f},\theta}(\mathbf{x}_0,t_0)}\theta(\mathbf{x}_0)<0$*, then* $\exists\varepsilon>0,\forall t\in(t_0,t_0+\varepsilon),\theta(\mathbf{x}(t))<0$*;*

(iii) if $\gamma_{\mathbf{f},\theta}(\mathbf{x}_0,t_0)=\infty$*, then* $\exists\varepsilon>0,\forall t\in(t_0,t_0+\varepsilon),\theta(\mathbf{x}(t))=0$*.*

Proof First, recall Definition 2.9, $\gamma_{\mathbf{f},\theta}(\mathbf{x}_0,t)\neq 0$ if and only if $\mathcal{L}_{t,\mathbf{f}}^{0}\theta(\mathbf{x})=0$, which, by Definition 2.7, is equivalent to $\theta(\mathbf{x}_0)=0$.

Next, assume now $\gamma_{\mathbf{f},\theta}(\mathbf{x}_0,t)\neq 0$. Since $\mathbf{f}$ is a polynomial vector function, we know $\mathbf{f}$ is analytic. So the initial value problem for differential equation $\dot{\mathbf{x}}(t)=\mathbf{f}(\mathbf{x}(t),t)$ and $\mathbf{x}(t_0)=\mathbf{x}_0$ exists a unique solution near t_0 [40]. Since θ is also a polynomial function, we can conclude that $\theta(\mathbf{x}(t))$ is analytic near t_0. So we have:

$$\begin{aligned}\theta(\mathbf{x}(t))&=\theta(\mathbf{x}(t_0))+\frac{\mathrm{d}\theta}{\mathrm{d}t}(t_0)*(t-t_0)+\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2}*\frac{(t-t_0)^2}{2!}+\dots\\&=\mathcal{L}_{t_0,\mathbf{f}}^{0}\theta(\mathbf{x}_0)+\mathcal{L}_{t_0,\mathbf{f}}^{1}\theta(\mathbf{x}_0)*(t-t_0)+\mathcal{L}_{t_0,\mathbf{f}}^{2}\theta(\mathbf{x}_0)*\frac{(t-t_0)^2}{2!}+\dots\end{aligned}$$

at the neighbourhood of t_0.

For the first two cases, we see that the first $\gamma_{\mathbf{f},\theta}(\mathbf{x}_0,t_0)$ terms of the right side equation all are equal to 0, and the coefficient of the dominant term has the same sign as $\mathcal{L}_{t_0,\mathbf{f}}^{\gamma_{\mathbf{f},\theta}(\mathbf{x}_0,t_0)}\theta(\mathbf{x}_0)$, so we can immediately get the corresponding conclusions.

As for the last case, $\gamma_{\mathbf{f},\theta}(\mathbf{x},t_0)=\infty$ implies that every term of the right side expansion is equal to 0, which means $\theta(\mathbf{x}(t))=0$ on the interval. □

Note that the inverse of Theorem 3.1 also holds, i.e.,

Corollary 3.2 *For a sequence* $\{t_i\}$ *with* $t_i>t_0$ *and* $\lim_{i\to\infty}t_i=t_0$*, let* $\mathbf{x}_0\doteq\mathbf{x}(t_0)$*, then*

(i) if for all i, $\theta(\mathbf{x}(t_i))>0$*, then* $\gamma_{\mathbf{f},\theta}(\mathbf{x}_0,t_0)<\infty$ *and* $\mathcal{L}_{t_0,\mathbf{f}}^{\gamma_{\mathbf{f},\theta}(\mathbf{x}_0,t_0)}\theta(\mathbf{x}_0)>0$*;*

(ii) if for all i, $\theta(\mathbf{x}(t_i))<0$*, then* $\gamma_{\mathbf{f},\theta}(\mathbf{x}_0,t_0)<\infty$ *and* $\mathcal{L}_{t_0,\mathbf{f}}^{\gamma_{\mathbf{f},\theta}(\mathbf{x}_0,t_0)}\theta(\mathbf{x}_0)<0$*;*

(iii) if for all i, $\theta(\mathbf{x}(t_i))=0$*, then* $\gamma_{\mathbf{f},\theta}(\mathbf{x}_0,t_0)=\infty$*.*

3.2 Transverse Set and Invariant

In the following, for our purpose, we will extend some notions and results for autonomous HSs given in [11] to nonautonomous HSs.

Definition 3.3 (Transverse Set [11]) Given an HS $(\boldsymbol{H},\boldsymbol{I},\mathbf{f})$, the transverse set of a region $S(p(\mathbf{x})\geq 0)$ is defined as

$$\mathrm{Trans}_{\mathbf{f}\uparrow p}^{(t)}\doteq\{\mathbf{x}\in\mathbb{R}^n|\gamma_{\mathbf{f},p}(\mathbf{x},t)<\infty\wedge\mathcal{L}_{t,\mathbf{f}}^{\gamma_{\mathbf{f},p}(\mathbf{x},t)}p(\mathbf{x})<0\}$$

Intuitively, the transverse set of region $S(p(\mathbf{x})\geq 0)$ contains all elements that are either not in the region or will leave the region immediately when the system continuously evolves.

Like [11], using transverse set, we can give a necessary and sufficient condition of continuous invariants as follows.

Theorem 3.4 *$S(\theta(\mathbf{x}) \geq 0)$ is an invariant of a given HS $(\boldsymbol{H}, \boldsymbol{I}, \mathbf{f})$ if and only if*

(i) it implies the initial set $\boldsymbol{I}$, i.e., $(\mathbb{I}(\mathbf{x}) \geq 0) \Rightarrow (\theta(x) \geq 0)$;

(ii) at any time, any trajectory starting from any state on the boundary $S(\theta(\mathbf{x}) = 0)$ will never leave the region $S(\theta(\mathbf{x}) \geq 0)$, as long as the system evolves within its domain, i.e.,

$$(\theta(\mathbf{x}) = 0) \Rightarrow (\forall t > 0 : \mathbf{x} \in (\mathrm{Trans}^{(t)}_{\mathbf{f}\uparrow\theta})^c \cup \mathrm{Trans}^{(t)}_{\mathbf{f}\uparrow H})$$

Proof First we consider the necessary part. If the first condition is not satisfied, then we have $(\mathbb{I}(\mathbf{x}) \geq 0) \not\Rightarrow (\theta(x) \geq 0)$. That is to say $\boldsymbol{I} \not\subseteq S(\theta(\mathbf{x}) \geq 0)$, which contradicts the first condition of Definition 2.5.

Now if the second condition is not satisfied, then there exists $\mathbf{x}_0$ and $t_0 > 0$ such that $\theta(\boldsymbol{x_0}) = 0$ and $\mathbf{x}_0 \in \mathrm{Trans}^{(t_0)}_{\mathbf{f}\uparrow\theta} \cap (\mathrm{Trans}^{(t_0)}_{\mathbf{f}\uparrow H})^c$. By Definition 3.3 and Theorem 3.1, it follows

$$\exists \varepsilon > 0, \forall t \in (t_0, t_0 + \varepsilon), (\mathbb{H}(\mathbf{x}(t)) \geq 0) \cap (\theta(\mathbf{x}(t)) < 0)$$

which contradicts the second condition of Definition 2.5. This concludes the necessary part.

Now, let's consider the sufficient part. It's easy to see that the first condition of this theorem is essentially the same as the first condition of Definition 2.5. So we only need to prove $S(\theta(\mathbf{x}) \geq 0)$ also satisfies the second condition of Definition 2.5. Suppose the second condition does not satisfy, which means there exists $t_0 \geq 0$, $\mathbf{x}_0 \in S(\theta(\mathbf{x}) \geq 0)$ and $T_0 > 0$ such that for a certain trajectory $\mathbf{x}$ with $\mathbf{x}(t_0) = \mathbf{x}_0$,

$$(\forall t \in [t_0, t_0 + T_0], \mathbf{x}(t) \in \boldsymbol{H}) \not\Rightarrow (\forall t \in [t_0, t_0 + T_0], \mathbf{x}(t) \in S(\theta(\mathbf{x}) \geq 0)),$$

which is equivalent to

$$(\forall t \in [t_0, t_0 + T_0], \mathbb{H}(\mathbf{x}(t)) \geq 0) \wedge (\exists t \in [t_0, t_0 + T_0], \theta(\mathbf{x}(t)) < 0).$$

Now, consider $\theta(\mathbf{x}(t))$ as a function. We know that $\theta(\mathbf{x}(t_0)) = \theta(\mathbf{x}_0) \geq 0$ and for some $t_c \in [t_0, t_0 + T_0]$, $\theta(\mathbf{x}(t_c)) < 0$. By the continuity of $\mathbf{x}$, there exists $t_z \in [t_0, t_c)$ such that $\theta(\mathbf{x}(t_z)) = 0$, which means the set $\{t \in [t_0, t_c) \mid \theta(\mathbf{x}(t)) = 0\}$ is not empty. Set $t_m \doteq \sup\{t \in [t_0, t_c] \mid \theta(\mathbf{x}(t)) = 0\}$. By the definition of supremum and continuity of $\theta(\mathbf{x}(t))$, $\theta(\mathbf{x}(t_m)) = 0$. Denote $\mathbf{x}(t_m)$ by $\mathbf{x}_m$. As t_m is the rightmost zero point of $\theta(\mathbf{x}(t))$ and $\theta(\mathbf{x}(t_c)) < 0$, we know: $\forall t \in (t_m, t_c), \theta(\mathbf{x}(t)) < 0$. Use Corollary 3.2, we get: $\gamma_{\mathbf{f},\theta}(\mathbf{x}_m, t_m) < \infty$ and $\mathcal{L}^{\gamma_{\mathbf{f},\theta}(\mathbf{x}_m,t_m)}_{t_m,\mathbf{f}}\theta(\mathbf{x}(t_m)) < 0$. Recall Definition 3.3, that is exactly $\mathbf{x}_m \in \mathrm{Trans}^{(t_m)}_{\mathbf{f}\uparrow\theta}$.

Since $\forall t \in [t_0, t_0 + T_0], \mathbb{H}(\mathbf{x}(t)) \geq 0$, it follows $\forall t \in (t_m, t_c), \mathbb{H}(\mathbf{x}(t)) \geq 0$. Combine the first and third cases of Corollary 3.2, we get

(i) $\gamma_{\mathbf{f},\mathbb{H}}(\mathbf{x}_m, t_m) < \infty$ and $\mathcal{L}^{\gamma_{\mathbf{f},\mathbb{H}}(\mathbf{x}_m,t_m)}_{t_m,\mathbf{f}}\mathbb{H}(\mathbf{x}(t_m)) > 0$, or:

(ii) $\gamma_{\mathbf{f},\mathbb{H}}(\mathbf{x}_m, t_m) = \infty$.

In either way, we have $\mathbf{x}_m \notin \mathrm{Trans}^{(t_m)}_{\mathbf{f}\uparrow H}$. But $\theta(\mathbf{x}_m) = 0$ contradicts the second condition of this theorem. The sufficient part is proved. □

3.3 Ideals Generated by Lie Derivatives

Definition 3.5 (Ideal Generated by Lie Derivatives) Given a PHS $(\boldsymbol{H}, \boldsymbol{I}, \mathbf{f})$ and a polynomial $\theta(\mathbf{x})$, in $\mathbb{R}[\mathbf{x}]$, we call $\boldsymbol{I}_k^{(t)} \doteq \langle \mathcal{L}_{t,\mathbf{f}}^0\theta(\mathbf{x}), \mathcal{L}_{t,\mathbf{f}}^1\theta(\mathbf{x}), \ldots, \mathcal{L}_{t,\mathbf{f}}^k\theta(\mathbf{x})\rangle$ as *k-th ideal generated by Lie derivatives*, and $\boldsymbol{I}^{(t)} \doteq \bigcup_k \boldsymbol{I}_k^{(t)}$ as *ideal generated by Lie derivatives*, where $t \in \mathbb{R}$ is a parameter.

In order to make the above parameterized ideals generated by Lie derivatives computable, we can see the parameter itself as a normal variable, and obtain so-called *total ideals generated by Lie derivatives*. Thus, all members of *k-th total ideal generated by Lie derivatives* and *total ideal generated by Lie derivatives* are from $\mathbb{R}[\mathbf{x}, t]$.

Regarding the parameterized ideals generated by Lie derivative, we can prove the following theorem, which is similar to the one in autonomous HSs in [11].

Theorem 3.6 *There exists $N \in \mathbb{N}$ such that for every $t \in \mathbb{R}$, $\boldsymbol{I}^{(t)} = \boldsymbol{I}_N^{(t)}$.*

Proof Total ideals generated by Lie derivatives form an ascending ideal chain in $\mathbb{R}[\mathbf{x}, t]$, i.e.,

$$\boldsymbol{I}_0 \subseteq \boldsymbol{I}_1 \subseteq \cdots \subseteq \boldsymbol{I}_k \subseteq \ldots$$

By Theorem 2.13, we know that there exists N such that for any $l \geq N$, $\boldsymbol{I}_l = \boldsymbol{I}_N$, which means $\boldsymbol{I}_N = \boldsymbol{I}$.

Now for arbitrary $t_0 \in \mathbb{R}$, we claim that $\boldsymbol{I}_N^{(t_0)} = \boldsymbol{I}^{(t_0)}$. Clearly, $\boldsymbol{I}_N = \boldsymbol{I}$ derives that for any $l \geq N$, $\mathcal{L}_{t,\mathbf{f}}^l\theta \in \langle \mathcal{L}_{t,\mathbf{f}}^0\theta, \mathcal{L}_{t,\mathbf{f}}^1\theta, \ldots, \mathcal{L}_{t,\mathbf{f}}^N\theta\rangle$, which implies that there exists $\{g_j\} \in \mathbb{R}[\mathbf{x}, t]$ such that

$$\mathcal{L}_{t,\mathbf{f}}^l\theta = \sum_{0\leq j\leq N} g_j \mathcal{L}_{t,\mathbf{f}}^j\theta.$$

Replace t with t_0 in the both side of the above equation, we get

$$\mathcal{L}_{t,\mathbf{f}}^l\theta = \sum_{0\leq j\leq N} g_j[t_0/t] \mathcal{L}_{t,\mathbf{f}}^j\theta.$$

Therefore, for any $t \in \mathbb{R}$, any $l \geq N$, $\mathcal{L}_{t,\mathbf{f}}^l\theta \in \boldsymbol{I}_N^{(t)}$ holds, which immediately leads us to the final conclusion. □

Corollary 3.7 *There exists $N \in \mathbb{N}$, which is independent of $\mathbf{x}$ and t, such that $\gamma_{\mathbf{f},\theta}(\mathbf{x}, t) < \infty$ if and only if $\gamma_{\mathbf{f},\theta}(\mathbf{x}, t) \leq N$.*

Proof The sufficient part is obvious. We only need to consider the necessary part.

Take N as defined in Theorem 3.6, suppose $\gamma_{\mathbf{f},\theta}(\mathbf{x}, t) > N$, since $\boldsymbol{I}_N = \boldsymbol{I}$, it follows $\mathcal{L}_{t,\mathbf{f}}^{\gamma_{\mathbf{f},p}(\mathbf{x},t)}p(\mathbf{x}) \in \boldsymbol{I}_N$. So,

$$\mathcal{L}_{t,\mathbf{f}}^{\gamma_{\mathbf{f},p}(\mathbf{x},t)}p(\mathbf{x}) = \sum_{0\leq j\leq N} g_j \mathcal{L}_{t,\mathbf{f}}^j p(\mathbf{x})$$

for some g_js in $\mathbb{R}[\mathbf{x}]$.

On the other hand, as $\gamma_{\mathbf{f},\theta}(\mathbf{x}, t) > N$, for $0 \leq j \leq N$, we have $\mathcal{L}_{t,\mathbf{f}}^j p(\mathbf{x}) = 0$. Thus, it follows $\mathcal{L}_{t,\mathbf{f}}^{\gamma_{\mathbf{f},p}(\mathbf{x},t)} = 0$, which contradicts to Definition 2.9. □

Lemma 3.8 (Fixed Point Theorem [11]) *If $\mathcal{L}_{t,\mathbf{f}}^{i+1}\theta \in \boldsymbol{I}_i^{(t)}$ for some i, then $\mathcal{L}_{t,\mathbf{f}}^{m}\theta \in \boldsymbol{I}_i^{(t)}$ for any $m > i$.*

Proof The proof can be given similarly to [11]. □

Algorithm 1: Computing Upper Bound N: Naive

```
i := 0;
B := {L^0_{t,f}};
while true do
    Compute L^{i+1}_{t,f} from L^i_{t,f} ;
    Generate ideal I from basis B ;
    Check whether L^{i+1}_{t,f} ∈ I using Gröbner basis ;
    if L^{i+1}_{t,f} ∉ I then
        B := B ∪ {L^{i+1}_{t,f}};
        i := i + 1;
    else
        break;
N := i
```

Algorithm 1 is a naive algorithm to implement Lemma 3.8, a more efficient algorithm will be presented in Algorithm 2. To this end, we need some basic notions and results.

Definition 3.9 A monomial ideal of $\mathbb{R}[\mathbf{x}]$ is an ideal that can be generated by a set of monomials (maybe infinite). More specifically, $\boldsymbol{I}$ is a monomial ideal of $\mathbb{R}[\mathbf{x}]$ if and only if

$$\boldsymbol{I} = \{\sum_i g_i m_i \mid g_i \in \mathbb{R}[\mathbf{x}], m_i \in M\},$$

where M is a set of monomials in $\mathbb{R}[\mathbf{x}]$.

In what follows, we adopt the *degree-lexicographic order* over monomials.

For simplicity, we use $\langle M\rangle$ to denote the monomial ideal generated by M, and $m(p)$ to denote the leading term of polynomial p, i.e., the largest term w.r.t. the given order. For a polynomial ideal $\boldsymbol{I}$, we define $m(\boldsymbol{I}) \doteq \langle\{m(p) \mid p \in \boldsymbol{I}\}\rangle$.

Lemma 3.10 *Given polynomials $p_0, p_1, \ldots, p_n$, let $\boldsymbol{I}_i = \langle p_0, \ldots, p_i\rangle$. If $\boldsymbol{I}_0 \subsetneq \boldsymbol{I}_1 \subsetneq \cdots \subsetneq \boldsymbol{I}_n$ then $m(\boldsymbol{I}_0) \subsetneq m(\boldsymbol{I}_1) \subsetneq \cdots \subsetneq m(\boldsymbol{I}_n)$.*

Proof Since $\boldsymbol{I}_i \subsetneq \boldsymbol{I}_{i+1}$, it derives $p_{i+1} \notin \boldsymbol{I}_i$. If $m(\boldsymbol{p}_{i+1}) \notin m(\boldsymbol{I}_i)$, then $m(\boldsymbol{I}_i) \subsetneq m(\boldsymbol{I}_{i+1})$. Otherwise, there is some $j \leq i$ such that $m(p_j)$ divides $m(p_{i+1})$. In that case, replace p_{i+1} with the non-zero remainder p'_{i+1} of the ordered division of p_{i+1} by p_j. Clearly, $\boldsymbol{I}_k$ and $m(\boldsymbol{I}_k)$ keep unchanged. As the degree of p_{i+1} decreases, the process will terminate in finite steps, so we have $m(\boldsymbol{I}_i) \subsetneq m(\boldsymbol{I}_{i+1})$.

Take i from 0 to $n-1$, we get the conclusion of this lemma. □

Now an improved algorithm to compute N can be given in Algorithm 2.

Algorithm 2: Computing Upper Bound N: Improved

$i := 0$;
$p_0 := \mathcal{L}^0_{t,\mathbf{f}}$;
$\boldsymbol{B} := \{m(p_0)\}$;
while *true* **do**
  Compute $\mathcal{L}^{i+1}_{t,\mathbf{f}}$ from $\mathcal{L}^i_{t,\mathbf{f}}$;
  $p_{i+1} := \mathcal{L}^{i+1}_{t,\mathbf{f}}$;
  while $p_{i+1} \neq 0$ *and there exists* $m(p_j) \in \boldsymbol{B}$ *divides* $m(p_{i+1})$ **do**
    $p_{i+1} :=$ the remainder of the division of p_{i+1} by p_j ;
  if p_{i+1} *equals to* 0 **then**
    break;
  $\boldsymbol{B} := \boldsymbol{B} \cup \{m(p_{i+1})\}$;
  $i := i + 1$;
$N := i$;

Every execution of the inner while loop decreases the degree of p_{i+1} strictly at least 1, so the inner while loop can only be executed finitely many times. On the other hand, each execution of the outer loop generates a strictly ascending monomial ideal chain, so by Theorem 2.13 it also terminates after finitely many iterations.

3.4 A Necessary and Sufficient Condition

Based on the previous discussion, we can present the following necessary and sufficient condition on a polynomial formula being an invariant in the simple case.

Theorem 3.11 *Given an HS* $(\boldsymbol{H}, \boldsymbol{I}, \mathbf{f})$ *and a polynomial* $\theta(\mathbf{x})$*. We can construct a finite length formula* $\Pi(\theta, \mathbb{H}, \mathbf{f}, \mathbf{x}, t)$ *such that* $\theta(\mathbf{x}) \geq 0$ *is an invariant if and only if formula* $(\boldsymbol{I}(\mathbf{x}) \geq 0) \to (\theta(\mathbf{x}) \geq 0)$ *and* $\Pi(\theta, \mathbb{H}, \mathbf{f}, \mathbf{x}, t)$ *hold.*

Proof By Theorem 3.4, obviously, $\boldsymbol{I} \subseteq S(\theta(\mathbf{x}) \geq 0)$ if and only if

$$(\boldsymbol{I}(\mathbf{x}) \geq 0) \to (\theta(\mathbf{x}) \geq 0).$$

By Definition 3.3, $\forall t \geq 0, \mathbf{x} \in (\mathrm{Trans}^{(t)}_{\mathbf{f}\uparrow\theta})^c \cup \mathrm{Trans}^{(t)}_{\mathbf{f}\uparrow H}$ if and only if

$$\forall t \geq 0, \neg(\gamma_{\mathbf{f},\theta}(\mathbf{x}, t) < \infty \wedge \mathcal{L}^{\gamma_{\mathbf{f},\theta}(\mathbf{x},t)}_{t,\mathbf{f}} < 0) \vee (\gamma_{\mathbf{f},\mathbb{H}}(\mathbf{x}, t) < \infty \wedge \mathcal{L}^{\gamma_{\mathbf{f},H}(\mathbf{x},t)}_{t,\mathbf{f}} < 0).$$

According to Corollary 3.7, it is equivalent to

$$\forall t \geq 0, \neg(\gamma_{\mathbf{f},\theta}(\mathbf{x}, t) \leq N \wedge \mathcal{L}^{\gamma_{\mathbf{f},\theta}(\mathbf{x},t)}_{t,\mathbf{f}} < 0) \vee (\gamma_{\mathbf{f},\mathbb{H}}(\mathbf{x}, t) \leq N \wedge \mathcal{L}^{\gamma_{\mathbf{f},H}(\mathbf{x},t)}_{t,\mathbf{f}} < 0),$$

for some natural number N.

Thus, let

$$\begin{aligned}\pi^{(0)}(\theta,\mathbf{f},\mathbf{x},t) &\doteq \theta(\mathbf{x})<0\\ \pi^{(i)}(\theta,\mathbf{f},\mathbf{x},t) &\doteq (\bigwedge_{0\le j<i}\mathcal{L}^j_{t,\mathbf{f}}\theta(\mathbf{x})=0)\wedge\mathcal{L}^i_{t,\mathbf{f}}\theta(\mathbf{x})<0 \qquad \text{for } i>0\\ \pi(\theta,\mathbf{f},\mathbf{x},t) &\doteq \bigvee_{0\le i\le N}\pi^{(i)}(\theta,\mathbf{f},\mathbf{x},t),\end{aligned}$$

then $\forall t\ge 0, \mathbf{x}\in(\mathrm{Trans}^{(t)}_{\mathbf{f}\uparrow\theta})^c\cup\mathrm{Trans}^{(t)}_{\mathbf{f}\uparrow H}$ if and only if formula

$$\forall t((t\ge 0)\to(\neg\pi(\theta,\mathbf{f},\mathbf{x},t)\vee\pi(\mathbb{H},\mathbf{f},\mathbf{x},t)))$$

which is further equivalent to

$$\forall t((t<0)\vee(\neg\pi(\theta,\mathbf{f},\mathbf{x},t)\vee\pi(\mathbb{H},\mathbf{f},\mathbf{x},t))).$$

Now, let

$$\Pi(\theta,\mathbb{H},\mathbf{f},\mathbf{x},t)\doteq(\theta(\mathbf{x})=0)\to(\forall t((t<0)\vee(\neg\pi(\theta,\mathbf{f},\mathbf{x},t)\vee\pi(\mathbb{H},\mathbf{f},\mathbf{x},t))))$$

Thus, this completes the proof. □

3.5 Invariant Synthesis

A template polynomial is a parameterized polynomial $p(\boldsymbol{u},\mathbf{x})\in\mathbb{R}[u_1,\ldots,u_m,x_1,\ldots,x_n]$, where $u_1,\ldots,u_m$ are parameters. Our purpose is to find an invariant $\theta(\mathbf{x})$ such that for some $\boldsymbol{u}_0\in\mathbb{R}^m$, $p(\boldsymbol{u}_0,\mathbf{x})=\theta(\mathbf{x})$, or report "no such invariant exists".

Notice that Theorem 3.6 and related corollaries can be extended to parametric case without any substantial change.

Lemma 3.12 *There exists $N\in\mathbb{N}$ which is independent of $\mathbf{x}$, t and $\boldsymbol{u}$ such that $\gamma_{\mathbf{f},\theta}(\mathbf{x},t)<\infty$ if and only if $\gamma_{\mathbf{f},\theta}(\mathbf{x},t)\le N$.*

Now basic steps of invariant generation can be sketched as follows:

(i) Predefine a template $p(\boldsymbol{u},\mathbf{x})\in\mathbb{R}[\boldsymbol{u},\mathbf{x}]$.

(ii) Compute the upper bound N w.r.t p and $\mathbf{f}$.

(iii) Construct $\Pi(p,\mathbb{H},\mathbf{f},\boldsymbol{u},\mathbf{x},t)$ as in the proof of Theorem 3.11.

(iv) Apply quantifier elimination algorithms to

$$\forall\mathbf{x}.(\Pi(p,\mathbb{H},\mathbf{f},\boldsymbol{u},\mathbf{x},t)\wedge((\boldsymbol{I}(\mathbf{x})\ge 0)\to(p(\boldsymbol{u},\mathbf{x})\ge 0))).$$

Clearly, the resulted quantifier-free formula $\Xi(\boldsymbol{u})$ is a constraint on $\boldsymbol{u}$.

(v) Apply some constraint solver, such as DISCOVERER [41], to find a solution $\boldsymbol{u}_0$ to $\Xi(\boldsymbol{u})$. If it returns *false* then report "There does not exist an invariant with the predefined template"; otherwise, let $\theta(\mathbf{x})=p(\boldsymbol{u}_0,\mathbf{x})$. Obviously, $\theta(\mathbf{x})\ge 0$ is an invariant.

4 Complexity Analysis and Experimental Results

4.1 Complexity Analysis

In order to conduct complexity analysis of our approach, an explicit upper bound on the N is necessary. But how to compute an upper bound on N is mathematically hard, which makes the complexity analysis of Liu *et al.*'s approach [11] unclear so far. In this section, by exploiting the results reported in [42], we give an explicit upper bound on N, and based on which the complexity analysis of our approach is achieved. Similarly, the complexity of Liu *et al.*'s approach [11] can be obtained similarly.

In the following, we sketch the main results reported in [42], please refer to [42] for the details.

Definition 4.1 For any increasing function $f : \mathbb{N} \to \mathbb{N}$ (that is, $f(x) \leq f(y)$ for all $x \leq y$), an ascending chain $I_1 \subseteq I_2 \subseteq \cdots \subseteq I_n$ of polynomial ideals of $\mathbb{K}[\boldsymbol{x}]$ is called f-bounded, if $\mathrm{gdeg}(I_i) \leq f(i)$ for all $i \geq 1$. Denote by $L(d, f)$ the greatest length of all strictly ascending chains of $\mathbb{K}[\boldsymbol{x}]$ which are f-bounded.

Remark 4.2 (i) The condition of f-boundedness is necessary to define the greatest length, as the length of chains with unbounded degrees could be arbitrarily large (for instance, the length of $\langle x^n \rangle \subset \langle x^{n-1} \rangle \subset \cdots \subset \langle 1 \rangle$ could be arbitrarily large if n is unbounded).

(ii) For ease of discussion, we assume f is *increasing* without loss of generality. In fact, for a general f, consider the increasing function $F : \mathbb{N} \to \mathbb{N}$, $F(n) \triangleq \max\{f(1), f(2), \cdots, f(n)\}$. Then a f-bounded chain is always a F-bounded chain since $f(n) \leq F(n)$ for all n. So $L(d, f) \leq L(d, F)$, and we can use $L(d, F)$ as the upper bound of the chains.

Given a number $d \in \mathbb{N}$, an increasing function $f : \mathbb{N} \to \mathbb{N}$, and a number $t \in \{0, 1, \cdots, f(1)\}$, $\Omega(d, f, t)$ is recursively calculated as follows:

(i) $\Omega(0, f, t) = 1$ and $\Omega(d, f, 0) = 1$, for any $d \geq 1$, f and t;

(ii) Write $n_t \triangleq \Omega(d, f, t)$ for $t = 0, 1, \cdots, f(1)$, then they can be successively calculated by $n_0 = 0$ and for $t \geq 1$,

$$n_t = n_{t-1} + \Omega(d - 1, f_{+n_{t-1}}, f(n_{t-1} + 1) - f(1) + 1).$$

Here f_{+m} is a function defined as $f_{+m}(n) = f(m + n)$.

Theorem 4.3 $L(d, f) = \Omega(d, f, f(1))$.

Proof Please refer to [42]. □

Exploiting the results reported in [36], we can prove that $\Omega(d, f, t)$ is primitive recursive and lies in the level 2 of the Fast Growing Hierarchy, which includes all the elementary functions if d is fixed, otherwise, it will have Ackermann's function as its lower bound, i.e., it becomes a general recursive function.

Theorem 4.4 (Complexity) *For a given hybrid system with d variables, suppose the degree of vector field is n, and the degree of parametric polynomials in a predefined invariant is m, then the complexity of our algorithm is* $O(2^{2^d} * \Omega(d, n-1, n-1))$.

Proof It can be easily obtained by the complexity of Gröbner basis [43], Algorithm 1, Algorithm 2, and Theorem 4.3. □

4.2 Experimental Results

In this part, we report some experimental results in Table 4.2. In Table 4.2, the first column is the considered vector field, the second column is the polynomial in the invariant to be verified (here, we only consider the simple case), the third column is the set of variables occurring in the test polynomial, the fourth column is the corresponding time using Algorithm 1, while the fifth column is the corresponding time using Algorithm 2, the sixth column is the N computed by Algorithm 1, the seventh column is the N computed by Algorithm 2, and the eighth column is the upper bound on N.

vector fields	test polynomial	variables	time consumed(naive)	time consumed(improved)	result(naive)	result(improved)	upper bound
$\{y-t, x+t, x+y\}$	$x+2y+z$	x,y,z	0.00391	0.003256	2	2	4
$\{y-t, x+t, x+y\}$	x^2+y+z	x,y,z	0.043852	0.039875	3	3	9
$\{y^2-t, x^2+t, x+y\}$	x^2+y+z	x,y,z	2.69769	2.56613	3	3	262155
$\{y^2-t, x^2+t, x+y\}$	$(x+y+z)^2+y$	x,y,z	38.1724	36.194	3	3	262155
$\{y, x^2-t, x+y\}$	$(x+y+z)^2+y$	x,y,z	1.8461	1.94017	3	3	262155
$\{y, x^2-t, x+y\}$	$(x+y+z)^3+y$	x,y,z	over 5 minutes	over 5 minutes	none	none	too large
$\{y+t, x-t, x+y\}$	$(x+y+z)^3+y$	x,y,z	0.124435	0.11002	3	3	14
$\{y+t, x-t, x+y\}$	$(x+y+z)^4+y$	x,y,z	0.745951	0.750883	3	3	19
$\{y+t, x-t, x+y\}$	$(x+y+z)^5+y$	x,y,z	6.33599	5.7566	3	3	24
$\{y+t, x-t, x+y\}$	$(x+y+z)^6+y$	x,y,z	37.2067	36.4658	3	3	29
$\{y+t, x-t, x+y\}$	$(x+y+z)^7+y$	x,y,z	over 5 minutes	over 5 minutes	none	none	34
$\{y+t, x-t, x+y, z+t\}$	$(x+y+z+u)^2+y$	x,y,z,u	0.30055	0.261325	4	4	13
$\{y+t, x-t, x+y, z+t, u+x\}$	$(x+y+z+u+v)^2+y$	x,y,z,u,v	over 5 minutes	over 5 minutes	none	none	17

Table 1: Experimental results

Note that the complexity of our approach is quite high, so it is hard to solve a problem in which the vector field and the test polynomial both are non-linear. In particular, the degree of the vector field influence the efficiency so much, as indicated in Table 4.2. That is one of our future work to invent more efficient approach based on numeric computation or the combination of numeric computation and symbolic computation.

5 General Case

In this part, we consider invariant verification and generation for general polynomial hybrid systems. The domain and initial set will become to

$$\boldsymbol{H} = S(\bigvee_{i=1}^{I} \bigwedge_{j=1}^{J_i} p_{ij}(\mathbf{x}) \rhd 0)$$

$$\boldsymbol{I} = S(\bigvee_{i=1}^{N} \bigwedge_{j=1}^{M_i} q_{ij}(\mathbf{x}) \rhd 0)$$

and the possible invariant takes the form

$$\theta = \bigvee_{k=1}^{K} \bigwedge_{j=1}^{L_k} (p_{kl}(\mathbf{x}) \rhd 0)$$

where $\rhd \in \{>, \geq\}$.

The basic idea is almost the same as that of the simple case given above. However, the complex boundaries of invariant, domain and initial set require some extra discussion, but can be coped with as in the autonomous case, please refer to [44] (the full version of [11]) for the details.

6 Conclusion

In this paper we propose a sound and complete algorithm to verify invariants of polynomial non-autonomous hybrid systems, based on which, we derive a sound and relatively complete algorithm to automatically generate invariants for such systems."Relatively" means our approach can generate all invariants with the pre-defined template.

The ascending chain of ideals generated by Lie derivatives has finite length. A general recursive upper bound on the length can be obtained by the results reported in [35, 36, 42]. This upper-bound is useful in constructive proof and time complexity analysis, but it is often too large to be practical. So, we proposed two practical algorithms to compute how many steps the ascending chain of ideals can reach a fixed point.

Comparing with homogenizing non-autonomous hybrid systems to autonomous ones, and then applying the techniques for invariant generation of autonomous hybrid systems, the advantage of our approach include the following points:

- it is more scalable, which means that invariants for some non-autonomous hybrid system can be synthesized with our approach, but cannot be obtained by homogenizing it first and then applying invariant generation techniques for autonomous hybrid systems. This is quite similar to the well-known result that the stability of a non-autonomous hybrid system cannot be achieved by homogenizing it.
- It is more efficient, as we do not need to consider time.
- The generated invariants do not contain time, and therefore are more succinct.

In the future, we will focus on investigating some heuristic strategies on how to determine an invariant template for a given property to be verified. Another interesting work is to improve the efficiency of our approach by considering numeric computation or the combination of symbolic computation with numeric computation.

References

[1] R. Alur, C. Courcoubetis, T. A. Henzinger, and P. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid Systems, LNCS 736*, pages 209–229, 1992.

[2] E. Asarin, O. Bournez, T. Dang, and O. Maler. Approximate reachability analysis of piecewise-linear dynamical systems. In *HSCC 2000*, volume 1790 of *LNCS*, pages 20–31. Springer, 2000.

[3] G. Lafferriere, G. J. Pappas, and S. Yovine. Symbolic reachability computation for families of linear vector fields. *J. Symb. Comput.*, 32:231–253, Sept. 2001.

[4] R. Alur, T. Dang, and F. Ivancic. Predicate abstraction for reachability analysis of hybrid systems. *ACM Trasactions on Embedded Computing Systems*, 5(1):152–199, 2006.

[5] T. Gan, M. Chen, L. Dai, B. Xia, and N. Zhan. Decidability of the reachability for a family of linear vector fields. In *ATVA 2015, LNCS 9364*, pages 482–499, Shanghai, China, 12-15 Oct., 2015. Springer International Publishing, Switzerland.

[6] T. Gan, M. Chen, Y. Li, B. Xia, and N. Zhan. Computing reachable sets of linear vector fields revisited. To appear in ECC 2016.

[7] S. Prajna and A. Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *HSCC 2004*, volume 2993 of *LNCS*, pages 477–492. Springer, 2004.

[8] S. Sankaranarayanan, H. B. Sipma, and Z. Manna. Constructing invariants for hybrid systems. In *HSCC 2004*, volume 2993 of *LNCS*, pages 539–554. Springer, 2004.

[9] S. Gulwani and A. Tiwari. Constraint-based approach for analysis of hybrid systems. In *CAV 2008*, volume 5123 of *LNCS*, pages 190–203. Springer, 2008.

[10] A. Platzer and E. M. Clarke. Computing differential invariants of hybrid systems as fixedpoints. In *CAV 2008*, volume 5123 of *LNCS*, pages 176–189. Springer, 2008.

[11] J. Liu, N. Zhan, and H. Zhao. Computing semi-algebraic invariants for polynomial dynamical systems. In *EMSOFT 2011*, pages 97–106. ACM, 2011.

[12] L. Dai, T. Gan, B. Xia, and N. Zhan. Barrier certificate revisited. 2016. Journal of Symbolic Computation.

[13] E. Asarin, O. Bournez, T. Dang, O. Maler, and A. Pnueli. Effective synthesis of switching controllers for linear systems. In *Proceedings of the IEEE*, pages 1011–1025, 2000.

[14] C. Tomlin, J. Lygeros, and S. Sastry. A game theoretic approach to controller design for hybrid systems. *Proceedings of the IEEE*, 88(7):949–970, 2000.

[15] A. Taly and A. Tiwari. Deductive verification of continuous dynamical systems. In *FSTTCS 2009*, volume 4 of *LIPIcs*, pages 383–394, 2009.

[16] T. Sturm and A. Tiwari. Verification and synthesis using real quantifier elimination. In *ISSAC 2011*, pages 329–336. ACM, 2011.

[17] H. Zhao, N. Zhan, D. Kapur, and K. Larsen. A "hybrid" approach for synthesizing optimal controllers of hybrid systems: A case study of the oil pump industrial example. In *FM 2012, LNCS 7436*, pages 471–485, Paris, France, 27-31 Aug., 2012. Springer-Verlag, Berlin.

[18] H. Zhao, N. Zhan, and D. Kapur. Synthesizing switching controllers for hybrid systems by generating invariants. In *Theories of Programming and Formal Methods*, volume 8051 of *LNCS*, pages 354–373. 2013.

[19] S. Bensalem, M. Bozga, J.-C. Fernandez, L. Ghirvu, and Y. Lakhnech. A transformational approach for generating non-linear invariants. In *SAS 2000*, volume 1824 of *LNCS*, pages 58–74, 2000.

[20] M. A. Colòn, S. Sankaranarayanan, and H. B. Sipma. Linear invariant generation using non-linear constraint solving. In *CAV 2003*, pages 420–432. Springer, 2003.

[21] S. Sankaranarayanan, H. Sipma, and Z. Manna. Non-linear loop invariant generation using gröbner bases. In *POPL 2004*, pages 318–329, 2004.

[22] D. Kapur. Automatically generating loop invariants using quantifier elimination. In *IMACS Intl. Conf. on Applications of Computer Algebra*, 2004.

[23] J. Liu, J. Lv, Z. Quan, N. Zhan, H. Zhao, C. Zhou, and L. Zou. A calculus for hybrid CSP. In *APLAS 2010*, volume 6461 of *LNCS*, pages 1–15. 2010.

[24] A. Platzer. *Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics*. Springer, Heidelberg, 2010.

[25] S. Sankaranarayanan. Automatic abstraction of non-linear systems using change of bases transformations. In *HSCC 2011*, pages 143–152, New York, NY, USA, 2011. ACM.

[26] A. Platzer. A differential operator approach to equational differential invariants. In L. Beringer and A. Felty, editors, *Interactive Theorem Proving*, volume 7406 of *LNCS*, pages 28–48. Springer Berlin Heidelberg, 2012.

[27] A. Platzer. The complete proof theory of hybrid systems. In *LICS 2012*, pages 541–550, Washington, DC, USA, 2012. IEEE Computer Society.

[28] M. Jirstrand. Invariant sets for a class of hybrid systems. In *CDC'98*, volume 4, pages 3699–3704. IEEE, 1998.

[29] E. Rodríguez-Carbonell and A. Tiwari. Generating polynomial invariants for hybrid systems. In *HSCC 2005*, volume 3414 of *LNCS*, pages 590–605. Springer, 2005.

[30] S. Sankaranarayanan. Automatic invariant generation for hybrid systems using ideal fixed points. In *HSCC'10*, pages 221–230. ACM, 2010.

[31] S. Prajna and A. Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *HSCC 2004*, volume 2993 of *LNCS*, pages 477–492. Springer, 2004.

[32] S. Prajna, A. Jadbabaie, and G. Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52(8):1415–1428, 2007.

[33] H. Kong, F. He, X. Song, W. Hung, and M. Gu. Exponential-condition-based barrier certificate generation for safety verification of hybrid systems. In *CAV 2013*, volume 8044 of *LNCS*.

[34] C. Sloth, G. Pappas, and R. Wisniewski. Compositional safety analysis using barrier certificates. In *HSCC 2012*, pages 15–24, 2012.

[35] G. M. Socias. Length of polynomial ascending chains and primitive recursiveness. *Mathematica Scandinavica*, 71:181–205, 1992.

[36] D. Figueira, S. Figueira, S. Schmitz, and P. Schnoebelen. Ackermannian and primitive-recursive bounds with dickson's lemma. In *LICS 2011*, pages 269–278. IEEE, 2011.

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

[38] H. Kong, S. Bogomolov, C. Schilling, Y. Jiang, and T. A. Henzinger. Invariant clusters for hybrid systems. *arXiv preprint arXiv:1605.01450*, 2016.

[39] D. Cox, J. Little, and D. O'shea. *Ideals, varieties, and algorithms*, volume 3. Springer, 1992.

[40] M. Tenenbaum and H. Pollard. *Ordinary differential equations: An elementary textbook for students of mathematics, engineering, and the sciences*, chapter 9, pages 562–563. Courier Corporation, 1963.

[41] B. Xia. DISCOVERER: a tool for solving semi-algebraic systems. *ACM Communications in Computer Algebra*, 41(3):102–103, 2007.

[42] Y. Li, H. Lu, N. Zhan, M. Chen, and G. Wu. Termination analysis of polynomial programs with equality conditions. *CoRR*, abs/1510.05201, 2015.

[43] B. Buchberger. An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal. *Journal of Symbolic Computation*, 41(3-4):475–511, 2006.

[44] J. Liu, N. Zhan, and H. Zhao. Computing semi-algebraic invariants for polynomial dynamical systems. *ArXiv e-prints*, Feb. 2011. http://arxiv.org/abs/1102.0705, the full version of [11].