

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

WANG Qiuye · XIA Bican · ZHAN Naijun

DOI:

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

©The Editorial Office of JSSC & Springer-Verlag Berlin Heidelberg 2013

DOI:

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

©The Editorial Office of JSSC & Springer-Verlag Berlin Heidelberg 2013

Abstract In our previous work, we gave a necessary and sufficient condition for an 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 relative 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.

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

WANG Qiuye

State Key Lab. of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China

E-mail: wangqye@ios.ac.cn

XIA Bican

LMAM & School of Math. Sci., Peking University

E-mail: xbc@math.pku.edu.cn

ZHAN Naijun

State Key Lab. of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China

E-mail: znj@ios.ac.cn

* The first and third authors are supported partly by “973 Program” under grant No. 2014CB340701, by NSFC under grant 91418204, by CDZ project CAP (GZ 1023), and by the CAS/SAFEA International Partnership Program for Creative Research Teams; the second author is supported partly by NSFC under grants 11290141, 11271034 and 61532019.

◇ This paper was recommended for publication by Editor

1 Introduction

Embedded systems make use of computer units to control physical devices so that the behavior of the controlled devices meets expected requirements. In modern life, embedded systems have become more common, more indispensable, and more complicated. How to design correct embedded systems is a grand challenge in computer science and control theory. Model-driven development (MDD) was considered as an effective way of developing correct complex embedded systems, and has been successfully applied in industry [9, 11]. In the framework of MDD, a formal model of the system to be developed is defined at the beginning, and then extensive analysis and verification are done based on the formal model so that errors can be detected and corrected at the very early stage of the design of the system. Afterwards, model transformation techniques are applied to transform the abstract formal model into lower level models, even into source code. Hybrid systems combine discrete mode changes with continuous evolutions specified in the form of differential equations. With mathematically precise semantics, hybrid systems can serve as an appropriate model of embedded systems [1, 2, 6, 8, 14, 16, 17] and is a fundamental structure of various logics and calculi such as differential dynamic logic (\mathbf{dL}) in [16, 18], differential-algebraic dynamic logic (DAL) by Platzer [15] and hybrid Hoare logic by Liu et al. [12] for modeling, specifying and verifying embedded systems or cyber-physical systems.

A core problem of verifying hybrid systems is so-called safety verification problem. It asks whether a hybrid system will reach some unsafe states from given initial states. Obviously, if we can compute the reachable set of given system, which includes all states reachable from initial states, then we can solve safety verification problem by simply examines whether any unsafe state is in reachable set or not [4].

But for common differential equations, the reachable set is usually intractable, especially when time is unbounded. So the intuitive methods of directly computing reachable set can only be used for very special systems. To avoid the computing of reachable sets, methods based on invariants are brought up [19, 20].

Invariants can be seen as a special kind of over-approximation of reachable set. Informally, invariants (or inductive invariants, if called precisely) are sets satisfying that current state included in invariants implies that states evolved from current state are all included in invariants. A virtue of invariants is that they can be computed without solving differential equations.

Non-autonomous hybrid systems, i.e. systems that evolution function \mathbf{f} changes with time, are an important class of hybrid systems. Most previous works in that field focus on autonomous case, i.e. systems that evolution function \mathbf{f} doesn't change with time [13]. In this article we will focus on non-autonomous case, We propose a sound and complete algorithm for verifying semi-algebraic invariants for non-autonomous polynomial hybrid system. From it, we obtain a sound and relatively complete algorithm for generating these invariants.

The basic idea is to transform the definition of continuous invariant of non-autonomous polynomial hybrid system into a first order formula of finite length, then use quantifier elimination algorithm to obtain a equivalent quantifier-free formula. When verifying, the original

formula contains only bounded variables, so the corresponding quantifier-free formula will be either *TRUE* or *FALSE*, which represents whether the verified semi-algebraic set is an invariant of our system or not. When generating, some free variables representing the coefficients of possible invariants are introduced. The resulting quantifier-free formula can be seen as a constraint on those coefficients. Any solution to the constraint yields a valid invariant, and a *no solution* result denotes no invariant satisfies given template.

The main contribution of this article include:

- (i) we present a whole set of definitions for continuous invariants in non-autonomous case;
- (ii) we give a sound and complete algorithm for verifying semi-algebraic invariants for polynomial hybrid systems.
- (iii) we give a sound and relatively complete algorithm for generating semi-algebraic invariants for polynomial hybrid systems.

The article is organized as follows: Section 2 includes definitions and some background knowledge; Section 3 illuminates our method using a simplified problem when initial sets, domain and invariants are all defined by a single polynomial; Section 4 gives detailed version of our algorithms which can be used in general semi-algebraic sets; we conclude the article with Section 5.

2 Preliminaries

2.1 Definitions

The main difficulty in handling hybrid system is to deal with continuous dynamics. If we know how to verify or generate invariant for a certain mode of hybrid system, similar methods can be easily extended to the whole system, see [10, 19]. So for simplicity, we only consider single-mode hybrid systems:

Definition 2.1 (Hybrid System) A hybrid system is a tuple $(\mathbf{H}, \mathbf{I}, \mathbf{f})$, where:

- (i) $\mathbf{H} \subseteq \mathbb{R}^n$ is the domain of system state.
- (ii) $\mathbf{I} \subseteq \mathbf{H}$ is the set of initial states, which will be called the initial set.
- (iii) $\mathbf{f} \in \mathbf{H} \times [0, +\infty) \rightarrow \mathbb{R}^n$ is the evolution function of system. The evolution of system subjects to: $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t)$.

Definition 2.2 (Trajectory) For a hybrid system M , the trajectory originating from state \mathbf{x}_0 and time t_0 in time T is a differentiable function $\mathbf{x} \in [t_0, t_0 + T) \rightarrow \mathbb{R}^n$ satisfies:

- (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's obvious from context, we will omit the phrase “from \mathbf{x}_0 ”, “from time t_0 ”.

Definition 2.3 (Safety) Given an unsafe set $X_U \subseteq \mathbb{R}^n$, we say a hybrid system is safe, if no trajectories satisfy:

- (i) $\mathbf{x}(0) \in \mathbf{I}$; and
- (ii) $\exists \tau \geq 0 : \mathbf{x}(\tau) \in X_U$

Definition 2.4 (Continuous Invariant) The continuous invariant of a hybrid system $(\mathbf{H}, \mathbf{I}, \mathbf{f})$ is a set $\mathbf{A} \subseteq \mathbb{R}^n$ satisfies:

- (i) $\mathbf{I} \subseteq \mathbf{A}$, and
- (ii) For any $t_0 \geq 0$, $\mathbf{x}_0 \in \mathbf{A}$ and $T > 0$, any trajectories \mathbf{x} satisfying $\mathbf{x}(t_0) = \mathbf{x}_0$ also satisfy:
 $(\forall t \in [t_0, t_0 + T], \mathbf{x}(t) \in \mathbf{H}) \Rightarrow (\forall t \in [t_0, t_0 + T], \mathbf{x}(t) \in \mathbf{A})$

In this article, the word *invariant* always means continuous invariant defined here, if not state explicitly otherwise.

The main subject of our study is polynomial hybrid system:

Definition 2.5 (Polynomial Hybrid System) We say a hybrid system $(\mathbf{H}, \mathbf{I}, \mathbf{f})$ is polynomial, if:

- (i) Domain \mathbf{H} is a semi-algebraic set;
- (ii) Initial set \mathbf{I} is a semi-algebraic set;
- (iii) Evolution function \mathbf{f} is a polynomial vector function for t and \mathbf{x} . That is to say: every $f_i(\mathbf{x}, t)$ in $\mathbf{f}(\mathbf{x}, t) = (f_1(\mathbf{x}, t), f_2(\mathbf{x}, t), \dots, f_n(\mathbf{x}, t))$ satisfies: $f_i(\mathbf{x}, t) \in \mathbb{R}[\mathbf{x}, t]$, where $\mathbb{R}[\mathbf{x}, t]$ is a short version of $\mathbb{R}[x_1, x_2, \dots, x_n, t]$.

In this article, we always assume a hybrid system is polynomial, if not state explicitly otherwise.

It's easy to verify that polynomial functions satisfy Lipschitz condition. By the famous Picard-Lindelöf theorem, there exists a unique solution to the initial value problem locally.

2.2 Background: Semi-algebraic Set

Definition 2.6 (Semi-algebraic Set) A semi-algebraic set is a subset of \mathbb{R}^n defined by a finite sequence of polynomial equations and inequalities, or any finite union of such sets.

Formally, a set $S \subseteq \mathbb{R}^n$ is a semi-algebraic set if and only if there exists formula ψ such that $S = \{\mathbf{x} \in \mathbb{R}^n \mid \psi(\mathbf{x}) \text{ satisfies}\}$, where ψ defined by:

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

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

It's easy to prove that operations on semi-algebraic set is equivalent to corresponding operations on the formula defining it. More specifically, if we use $S(\psi)$ to represent semi-algebraic set defined by ψ , we have the following properties:

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

2.3 Background: Lie Derivatives

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

Definition 2.7 (Lie Derivatives) For a given function ϕ , a given vector function $\mathbf{f}(t, \mathbf{x})$, we define Lie derivatives $\mathcal{L}_{t,f}$ inductively as:

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

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

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

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

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

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

In this article we will just call *pointwise rank* as *rank*.

Note that Lie derivatives and their rank both contains variable t .

2.4 Background: Polynomial Ideals Theory

Polynomial ideal is a powerful tool when we deal with polynomial related problems. Here we recall some basic contents about it, from [5].

For a given field \mathbb{K} and variables x_1, x_2, \dots, x_n , we can define polynomial ring over \mathbb{K} as $\mathbb{K}[x_1, x_2, \dots, x_n]$. We write $\mathbb{K}[\mathbf{x}]$ to represent $\mathbb{K}[x_1, x_2, \dots, x_n]$ and always assume $\mathbb{K} = \mathbb{R}$, i.e. coefficients are real numbers.

Definition 2.10 (Polynomial Ideals) Given a polynomial ring $\mathbb{K}[\mathbf{x}]$, a subset $\mathbf{I} \subseteq \mathbb{K}[\mathbf{x}]$ is called ideal if:

- (i) $0 \in \mathbf{I}$;
- (ii) For any $p(\mathbf{x}), q(\mathbf{x}) \in \mathbf{I}$, we have: $p(\mathbf{x}) + q(\mathbf{x}) \in \mathbf{I}$;

(iii) For any $p(x) \in \mathbf{I}$ and $h(x) \in \mathbb{K}[\mathbf{x}]$, we have: $p(x)h(x) = h(x)p(x) \in \mathbf{I}$.

Definition 2.11 (Generated Ideals) we call

$$\mathbf{I} \doteq \bigcap_{p_1, p_2, \dots, p_k \in \mathbf{I}', \mathbf{I}' \text{ is a ideal}} \mathbf{I}'$$

ideal generated by basis p_1, p_2, \dots, p_k , denoted by $\langle p_1, p_2, \dots, p_k \rangle$.

It's easy to prove that:

$$\langle p_1, p_2, \dots, p_k \rangle = \left\{ \sum_{i=1}^k p_i h_i \mid \forall i : h_i \in \mathbb{K}[\mathbf{x}] \right\}$$

A well-known result is every polynomial ideal in real field can be generated by finite basis:

Theorem 2.12 (Hibert's Basis Theorem) *Any polynomial ideal $\mathbf{I} \in \mathbb{R}[\mathbf{x}]$ can be generated by a finite class of basis. i.e. for any ideal $\mathbf{I} \in \mathbb{R}[\mathbf{x}]$, there exists $p_1, p_2, \dots, p_k \in \mathbb{R}[\mathbf{x}]$ such that $\mathbf{I} = \langle p_1, p_2, \dots, p_k \rangle$.*

The above theorem can take a different form:

Theorem 2.13 (Ascending Chain Theorem) *For any infinite ascending chain of ideals in $\mathbb{K}[\mathbf{x}]$:*

$$\mathbf{I}_1 \subseteq \mathbf{I}_2 \subseteq \dots \mathbf{I}_k \subseteq \dots$$

There exists N such that for any $l \geq N$, $\mathbf{I}_l = \mathbf{I}_N$.

3 Simple Case

In this section, we focus on a simplified version of our original problem. More specifically, we only consider invariants that can be defined by a single polynomial inequality $\phi(\mathbf{x}) \geq 0$, we use Φ to denote the invariant. Also we assume that the domain \mathbf{H} and initial set \mathbf{I} can be defined by a single polynomial inequality. We will use $\mathbf{H}(\mathbf{x}) \geq 0$ and $\mathbf{I}(\mathbf{x}) \geq 0$ to denote them separately. In this section, if not explicitly state otherwise, we always suppose our hybrid system takes that form.

3.1 Intuitive Idea and Lie Derivatives

Intuitively, an invariant is a set that if any time system state falls in it, it will never come out of it. Suppose we follow the trajectory of our system: when system state is inside Φ , by the continuity of system trajectories, obviously it must reach the boundary of Φ before it comes out of Φ . So the real danger of coming out of Φ occurs when system state is in the boundary of Φ . In this simple case, that is to say: for every states \mathbf{x}_0 satisfying $\phi(\mathbf{x}_0) = 0$ and any time $t_0 \geq 0$, there exists some positive time ε such that the trajectory originating from state \mathbf{x}_0 and time t_0 in time ε doesn't decrease the value ϕ .

To turn this intuitive idea into practical algorithm, the first step is to find a tool to describe the change of ϕ as system evolves. Lie derivatives are suitable for this.

Theorem 3.1 *Given polynomial ϕ and hybrid system $(\mathbf{H}, \mathbf{I}, \mathbf{f})$, $\gamma_{f, \phi}(\mathbf{x}_0, t) \neq 0$ if and only if $\mathbf{x}_0 \in S(\phi(\mathbf{x}) = 0)$, and if we take $\mathbf{x}(t_0) = \mathbf{x}_0$, then it follows that:*

(i) if $\gamma_{f,\phi}(\mathbf{x}_0, t_0) < \infty$ and $\mathcal{L}_{t_0,f}^{\gamma_{f,\phi}(\mathbf{x}_0,t_0)}\phi(\mathbf{x}_0) > 0$, then:

$$\exists \varepsilon > 0, \forall t \in (t_0, t_0 + \varepsilon), \phi(\mathbf{x}(t)) > 0$$

(ii) if $\gamma_{f,\phi}(\mathbf{x}_0, t_0) < \infty$ and $\mathcal{L}_{t_0,f}^{\gamma_{f,\phi}(\mathbf{x}_0,t_0)}\phi(\mathbf{x}_0) < 0$, then:

$$\exists \varepsilon > 0, \forall t \in (t_0, t_0 + \varepsilon), \phi(\mathbf{x}(t)) < 0$$

(iii) if $\gamma_{f,\phi}(\mathbf{x}_0, t_0) = \infty$, then:

$$\exists \varepsilon > 0, \forall t \in (t_0, t_0 + \varepsilon), \phi(\mathbf{x}(t)) = 0$$

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

Next, assume now $\gamma_{f,\phi}(\mathbf{x}_0, t) \neq 0$. 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 [22]. Since ϕ is also a polynomial function, we can conclude that $\phi(\mathbf{x}(t))$ is analytic near t_0 . So we have:

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

satisfies near t_0 .

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

As to the last situation, $\gamma_{f,\phi}(\mathbf{x}, t_0) = \infty$ implies that every term of the right side expansion is equal to 0, which means $\phi(\mathbf{x}(t)) = 0$ satisfies in the interval where unique solution exists and ϕ is analytic. ■

Note the three situations in Theorem 3.1 include all possible value of $\mathcal{L}_{t_0,f}^k$, so an inverse version of this theorem is possible, we state it as a corollary:

Corollary 3.2 For a sequence $\{t_i\}$ satisfying $t_i > t_0$ and $\lim_{i \rightarrow \infty} t_i = t_0$, we have:

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

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

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

3.2 Transverse Set and Invariant

We use Lie derivatives to define *transverse set* :

Definition 3.3 (Transverse Set) Given a hybrid system $(\mathbf{H}, \mathbf{I}, \mathbf{f})$, we define transverse set of region $S(p(\mathbf{x}) \geq 0)$ as:

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

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

Using transverse set, we can give the necessary and sufficient condition of continuous invariants:

Theorem 3.4 Given a hybrid system $(\mathbf{H}, \mathbf{I}, \mathbf{f})$, $S(\phi(\mathbf{x}) \geq 0)$ is an invariant of the system if and only if:

(i) It contains the initial set \mathbf{I} . Which is:

$$(\mathbf{I}(\mathbf{x}) \geq 0) \Rightarrow (\phi(\mathbf{x}) \geq 0)$$

(ii) Any time, as long as system evolves within its domain, any state in the boundary $S(\phi(\mathbf{x}) = 0)$ will not come out of the region $S(\phi(\mathbf{x}) \geq 0)$. Which is:

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

Proof First we consider the necessary part. If the first condition is not satisfied, which means $(\mathbf{I}(\mathbf{x}) \geq 0) \not\Rightarrow (\phi(\mathbf{x}) \geq 0)$, that is to say $\mathbf{I} \not\subseteq S(\phi(\mathbf{x}) \geq 0)$, which contradicts the first condition of Definition 2.4.

Now if the second condition is not satisfied, which means: there exists \mathbf{x}_0 and $t_0 > 0$ such that $\phi(\mathbf{x}_0) = 0$ and $\mathbf{x}_0 \in \text{Trans}_{f \uparrow \phi}^{(t_0)} \cap (\text{Trans}_{f \uparrow H}^{(t_0)})^c$. Then by Definition 3.3 and Theorem 3.1 we know:

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

which contradicts the second condition of Definition 2.4. That concludes the proof for the necessary part.

Next, 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 2.4. So we only need to prove $S(\phi(\mathbf{x}) \geq 0)$ also satisfies the second condition of Definition 2.4. Suppose the second condition doesn't satisfy, which means there exists $t_0 \geq 0$, $\mathbf{x}_0 \in S(\phi(\mathbf{x}) \geq 0)$ and $T_0 > 0$ such that for a certain trajectory $\mathbf{x}(t_0) = \mathbf{x}_0$, the following satisfies:

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

which is equivalent to:

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

Now consider $\phi(\mathbf{x}(t))$ as a function. We know that $\phi(\mathbf{x}(t_0)) = \phi(\mathbf{x}_0) \geq 0$ and for some $t_c \in [t_0, t_0 + T_0]$, $\phi(\mathbf{x}(t_c)) < 0$. By the continuity of \mathbf{x} , there exists $t_z \in [t_0, t_c]$ such that $\phi(\mathbf{x}(t_z)) = 0$, which means the set $\{t \in [t_0, t_c] \mid \phi(\mathbf{x}(t)) = 0\}$ is not empty.

Take $t_m \doteq \sup\{t \in [t_0, t_c] \mid \phi(\mathbf{x}(t)) = 0\}$. By the definition of supremum and continuity of $\phi(\mathbf{x}(t))$, $\phi(\mathbf{x}(t_m)) = 0$. Denote $\mathbf{x}(t_m)$ by \mathbf{x}_m . As t_m is the rightmost zero point of $\phi(\mathbf{x}(t))$, $\phi(\mathbf{x}(t_c)) < 0$, we know: $\forall t \in (t_m, t_c)$, $\phi(\mathbf{x}(t)) < 0$. Use Corollary 3.2, we get: $\gamma_{f,\phi}(\mathbf{x}_m, t_m) < \infty$ and $\mathcal{L}_{t_m,f}^{\gamma_{f,\phi}(\mathbf{x}_m, t_m)} \phi(\mathbf{x}(t_m)) < 0$. Recall Definition 3.3, that is exactly $\mathbf{x}_m \in \text{Trans}_{f \uparrow \phi}^{(t_m)}$.

Since $\forall t \in [t_0, t_0 + T_0]$, $\mathbf{H}(\mathbf{x}(t)) \geq 0$, certainly there is: $\forall t \in (t_m, t_c)$, $\mathbf{H}(\mathbf{x}(t)) \geq 0$. Combine the first and third situation of Corollary 3.2, we get:

- (i) $\gamma_{f,\phi}(\mathbf{x}_m, t_m) < \infty$ and $\mathcal{L}_{t_m,f}^{\gamma_{f,\phi}(\mathbf{x}_m, t_m)} \phi(\mathbf{x}(t_m)) > 0$, or:
- (ii) $\gamma_{f,\phi}(\mathbf{x}_m, t_m) = \infty$;

Either way, we have: $\mathbf{x}_m \notin \text{Trans}_{f \uparrow H}^{(t_m)}$.

But $\phi(\mathbf{x}_m) = 0$, that contradicts the second condition of this theorem. That concludes the proof for the sufficient part. ■

3.3 Ideals Generated by Lie Derivatives

The problem of Definition 3.3 is that it involves infinity. This subsection aims to prove that either the rank of Lie derivative is infinity, or it's less than a computable upper-bound N . The main tool used will be ideals generated by Lie derivatives:

Definition 3.5 (Ideal Generated by Lie Derivatives) Given polynomial hybrid system $(\mathbf{H}, \mathbf{I}, \mathbf{f})$ and polynomial $\phi(\mathbf{x})$, in $\mathbb{R}[\mathbf{x}]$ we call:

$$\mathbf{I}_k^{(t)} \doteq \langle \mathcal{L}_{t,f}^0 \phi(\mathbf{x}), \mathcal{L}_{t,f}^1 \phi(\mathbf{x}), \dots, \mathcal{L}_{t,f}^k \phi(\mathbf{x}) \rangle$$

as k -th ideal generated by Lie derivatives. Also call:

$$\mathbf{I}^{(t)} \doteq \bigcup_k \mathbf{I}_k^{(t)}$$

as ideal generated by Lie derivatives. where $t \in \mathbb{R}$ is a parameter.

We should note here t is considered to be a parameter in \mathbb{R} .

We want to prove that the ideal generated by Lie derivatives can be computed by finite steps of operations. For that, an extra definition is needed.

Definition 3.6 (Total Ideal Generated by Lie Derivatives) Consider $\mathcal{L}_{t,f}^k \phi(\mathbf{x})$ to be a element of $\mathbb{R}[\mathbf{x}, t]$, we call:

$$\mathbf{I}_k \doteq \langle \mathcal{L}_{t,f}^0 \phi(\mathbf{x}), \mathcal{L}_{t,f}^1 \phi(\mathbf{x}), \dots, \mathcal{L}_{t,f}^k \phi(\mathbf{x}) \rangle$$

as k -th total ideal generated by Lie derivatives. Also call:

$$\mathbf{I} \doteq \bigcup_k \mathbf{I}_k$$

as total ideal generated by Lie derivatives.

We see that total ideal is just to consider t as a variable rather than a parameter.

Now we are ready to prove the main theorem of this subsection:

Theorem 3.7 *There exists $N \in \mathbb{N}$ such that for every $t \in \mathbb{R}$, $\mathbf{I}^{(t)} = \mathbf{I}_N^{(t)}$.*

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

$$\mathbf{I}_0 \subseteq \mathbf{I}_1 \subseteq \cdots \subseteq \mathbf{I}_k \subseteq \cdots$$

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

Now for arbitrary $t \in \mathbb{R}$, we claim that $\mathbf{I}_N^{(t)} = \mathbf{I}^{(t)}$. To see that, let's first explore the total ideal chain. To say $\mathbf{I}_N = \mathbf{I}$ is to say for any $l \geq N$, $\mathcal{L}_{t,f}^l \phi \in \langle \mathcal{L}_{t,f}^0 \phi, \mathcal{L}_{t,f}^1 \phi, \dots, \mathcal{L}_{t,f}^N \phi \rangle$, which means there exists $\{g_j\} \in \mathbb{R}[\mathbf{x}, t]$ such that:

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

Substitute the value of t into it, we get:

$$\mathcal{L}_{t,f}^l \phi = \sum_{0 \leq j \leq N} g'_j \mathcal{L}_{t,f}^j \phi$$

where $\{g'_j\} \in \mathbb{R}[\mathbf{x}]$, this equation holds in $\mathbb{R}[\mathbf{x}]$.

So we have: for any $t \in \mathbb{R}$, any $l \geq N$, $\mathcal{L}_{t,f}^l \phi \in \mathbf{I}_N^{(t)}$ holds. That immediately leads us to our final conclusion. ■

Corollary 3.8 *There exists $N \in \mathbb{N}$ which is independent to \mathbf{x} and t such that $\gamma_{f,\phi}(\mathbf{x}, t) < \infty$ if and only if $\gamma_{f,\phi}(\mathbf{x}, t) \leq N$.*

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

Take the N in Theorem 3.7, suppose $\gamma_{f,\phi}(\mathbf{x}, t) > N$, since $\mathbf{I}_N = \mathbf{I}$, we have: $\mathcal{L}_{t,f}^{\gamma_{f,\phi}(\mathbf{x}, t)} p(\mathbf{x}) \in \mathbf{I}_N$. So:

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

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

But with $\gamma_{f,\phi}(\mathbf{x}, t) > N$, for $0 \leq j \leq N$ we have: $\mathcal{L}_{t,f}^j p(\mathbf{x}) = 0$. Substitute it into last equation we get: $\mathcal{L}_{t,f}^{\gamma_{f,\phi}(\mathbf{x}, t)} p(\mathbf{x}) = 0$, which contradicts Definition 2.9. ■

Next we will present an algorithm to find such a $N \in \mathbb{N}$. First we give a lemma:

Lemma 3.9 (Fixed Point Theorem) *If for some i we have:*

$$\mathcal{L}_{t,f}^{i+1} \phi \in \mathbf{I}_i^{(t)}$$

then for any $m > i$:

$$\mathcal{L}_{t,f}^m \phi \in \mathbf{I}_i^{(t)}$$

Proof To prove this lemma, we use induction. The situation $m = i + 1$ is exactly what we already have. Suppose we have proved for some $k > i$, the lemma holds. Now we consider the situation $m = k + 1$. From induction hypothesis, we know:

$$\mathcal{L}_{t,f}^k \phi \in \langle \mathcal{L}_{t,f}^0 \phi, \mathcal{L}_{t,f}^1 \phi, \dots, \mathcal{L}_{t,f}^i \phi \rangle$$

that is, for $0 \leq j \leq i$, there exists $\{g_j\} \in \mathbb{R}[\mathbf{x}]$ such that:

$$\mathcal{L}_{t,f}^k \phi = \sum_{0 \leq j \leq i} g_j \mathcal{L}_{t,f}^j \phi$$

Now for $\mathcal{L}_{t,f}^{k+1} \phi$, By Definition 2.7:

$$\begin{aligned} \mathcal{L}_{t,f}^{k+1} \phi &= \left\langle \frac{\partial}{\partial \mathbf{x}} \mathcal{L}_{t,f}^k \phi \cdot \mathbf{f} \right\rangle \\ &= \left\langle \frac{\partial}{\partial \mathbf{x}} \sum_{0 \leq j \leq i} g_j \mathcal{L}_{t,f}^j \phi \cdot \mathbf{f} \right\rangle \\ &= \sum_{0 \leq j \leq i} \langle \mathcal{L}_{t,f}^j \phi \frac{\partial}{\partial \mathbf{x}} g_j \cdot \mathbf{f} \rangle + \sum_{0 \leq j \leq i} \langle g_j \frac{\partial}{\partial \mathbf{x}} \mathcal{L}_{t,f}^j \phi \cdot \mathbf{f} \rangle \\ &= \sum_{0 \leq j \leq i} \left\langle \frac{\partial}{\partial \mathbf{x}} g_j \cdot \mathbf{f} \right\rangle \mathcal{L}_{t,f}^j \phi + \sum_{0 \leq j \leq i} g_j \mathcal{L}_{t,f}^{j+1} \phi \\ &= \sum_{0 \leq j \leq i} \left\langle \frac{\partial}{\partial \mathbf{x}} g_j \cdot \mathbf{f} \right\rangle \mathcal{L}_{t,f}^j \phi + \sum_{1 \leq j \leq i} g_j \mathcal{L}_{t,f}^j \phi + g_i \mathcal{L}_{t,f}^{i+1} \phi \end{aligned}$$

Recall the condition of this lemma, we have: $\mathcal{L}_{t,f}^{i+1} \phi \in \langle \mathcal{L}_{t,f}^0 \phi, \mathcal{L}_{t,f}^1 \phi, \dots, \mathcal{L}_{t,f}^i \phi \rangle$. So:

$$\mathcal{L}_{t,f}^{k+1} \in \langle \mathcal{L}_{t,f}^0 \phi, \mathcal{L}_{t,f}^1 \phi, \dots, \mathcal{L}_{t,f}^i \phi \rangle = \mathbf{I}_i^{(t)}$$

That concludes our proof for this lemma. ■

The basic frame of our algorithm is:

COMPUTING UPPER BOUND N: ORIGINAL

```

1   $i = 0, \mathbf{B} = \{\mathcal{L}_{t,f}^0\}$ 
2  while TRUE
3      Compute  $\mathcal{L}_{t,f}^{i+1}$  from  $\mathcal{L}_{t,f}^i$ 
4      Generate ideal  $\mathbf{I}$  from basis  $\mathbf{B}$ 
5      Ask whether  $\mathcal{L}_{t,f}^{i+1} \in \mathbf{I}$  using Gröbner basis
6      if  $\mathcal{L}_{t,f}^{i+1} \notin \mathbf{I}$ 
7           $\mathbf{B} = \mathbf{B} \cup \{\mathcal{L}_{t,f}^{i+1}\}$ 
8           $i = i + 1$ 
9      else
10         break
11   $N = i$ 
    
```

This is just a naive implement of Lemma 3.9. Next subsection we will give an alternative algorithm to compute the upper bound N .

3.4 Alternative Algorithm to Compute N

The algorithm we present in last subsection computes N using Gröbner basis, which is a rather “expensive” operation. Here we present an algorithm that avoids Gröbner basis, but may give a larger N comparing to the original algorithm. To achieve that goal, let’s first consider monomial ideals:

Definition 3.10 A monomial ideal of $\mathbb{R}[\mathbf{x}]$ is an ideal that can be generated by a set of monomials(maybe infinite). More specifically, \mathbf{I} is a monomial ideal of $\mathbb{R}[\mathbf{x}]$ if and only if:

$$\mathbf{I} = \langle m_1, m_2, \dots, m_k \dots \rangle$$

where m_i is a monomial in $\mathbb{R}[\mathbf{x}]$.

We use $m(p)$ to represent the leading term of polynomial p i.e. term that has the largest order. For an ideal \mathbf{I} , we define $m(\mathbf{I}) \doteq \langle \{m(p) \mid p \in \mathbf{I}\} \rangle$. When order monomials, we use the *degree-lexicographic order*, which is a total ordering compatible with the natural graduation by total degree.

We start with a lemma:

Lemma 3.11 Given polynomials p_0, p_1, \dots, p_n . Let $\mathbf{I}_i = \langle p_0, \dots, p_i \rangle$. if:

$$\mathbf{I}_0 \subsetneq \mathbf{I}_1 \subsetneq \dots \subsetneq \mathbf{I}_n$$

then:

$$m(\mathbf{I}_0) \subsetneq m(\mathbf{I}_1) \subsetneq \dots \subsetneq m(\mathbf{I}_n)$$

Proof We prove this lemma step by step. For \mathbf{I}_i , since $\mathbf{I}_i \subsetneq \mathbf{I}_{i+1}$, we know that $p_{i+1} \notin \mathbf{I}_i$. If $m(p_{i+1}) \notin m(\mathbf{I}_i)$, then $m(\mathbf{I}_i) \subsetneq m(\mathbf{I}_{i+1})$. Otherwise there is some $j \leq i$ such that $m(p_j)$ divides $m(p_{i+1})$. In that case, substitute p_{i+1} by the non-zero rest p'_{i+1} of ordered division of p_{i+1} by p_j , note \mathbf{I}_k and $m(\mathbf{I}_k)$ stay unchanged. As the order of p_{i+1} drops, the process will end in finite steps, so we also have $m(\mathbf{I}_i) \subsetneq m(\mathbf{I}_{i+1})$.

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

Now we can give the alternative algorithm to compute N :

COMPUTING UPPER BOUND N: ALTERNATIVE

```

1   $i = 0, p_0 = \mathcal{L}_{t,f}^0, \mathbf{B} = \{m(p_0)\}$ 
2  while TRUE
3      Compute  $\mathcal{L}_{t,f}^{i+1}$  from  $\mathcal{L}_{t,f}^i, p_{i+1} = \mathcal{L}_{t,f}^{i+1}$ 
4      while  $p_{i+1} \neq 0$  and there exists  $m(p_j) \in \mathbf{B}$  divides  $m(p_{i+1})$ 
5           $p_{i+1} =$  rest of ordered division of  $p_{i+1}$  by  $p_j$ 
6      if  $p_{i+1}$  equals to 0
7          break
8       $\mathbf{B} = \mathbf{B} \cup \{m(p_{i+1})\}$ 
9       $i = i + 1$ 
10  $N = i$ 
```

When looping, we always have $\mathbf{I}_i = \langle p_0, \dots, p_i \rangle$. When loop terminates, we have $p_{i+1} \in \langle p_0, \dots, p_i \rangle$, which suggests $\mathbf{I}_i = \mathbf{I}$ by Lemma 3.9. That guarantees the correctness of our algorithm.

Every time the inside while loop executed, the degree of p_{i+1} strictly drops. So the inside while loop can only be executed finite times. As for the outer loop, it's easy to prove \mathbf{B} of each loop execution generates a strictly ascending ideal chain, so by Theorem 2.13 it also ends after finite times of execution.

When the program terminates, we have $m(\mathbf{I}_{i+1}) = m(\mathbf{I}_i)$. By Lemma 3.11, i satisfies the condition $\mathbf{I}_{i+1} = \mathbf{I}_i$, so it is a valid upper bound N .

3.5 Verify Invariants of Hybrid System: Simple Case

Now, with previous discussions, we are ready to give the algorithm of verifying invariant of hybrid system. To do this, first we translate the conditions of Theorem 3.4 into first order logic formula of finite length, then use quantifier elimination to get the equivalent quantifier-free formula. Since all variables are bounded, the resulting quantifier-free formula will be either *TRUE* or *FALSE*. *TRUE* means the possible invariant is indeed an invariant of our hybrid system, while *FALSE* means it is not an invariant of our hybrid system. First let's see how to do the translation:

Theorem 3.12 *Given a hybrid system $(\mathbf{H}, \mathbf{I}, \mathbf{f})$ and a polynomial $\phi(\mathbf{x})$. We can construct a finite length formula $\theta(\phi, \mathbf{H}, \mathbf{f}, \mathbf{x}, t)$ such that $\phi(\mathbf{x}) \geq 0$ is an invariant if and only if formula $(\mathbf{I}(\mathbf{x}) \geq 0) \rightarrow (\phi(\mathbf{x}) \geq 0)$ and formula $\theta(\phi, \mathbf{H}, \mathbf{f}, \mathbf{x}, t)$ all satisfy.*

Proof Recall Theorem 3.4. Obviously, $\mathbf{I} \subseteq S(\phi(\mathbf{x}) \geq 0)$ if and only if formula

$$(\mathbf{I}(\mathbf{x}) \geq 0) \rightarrow (\phi(\mathbf{x}) \geq 0)$$

satisfies.

Now by Definition 3.3, $\forall t \geq 0, \mathbf{x} \in (\text{Trans}_{f \uparrow \phi}^{(t)})^c \cup \text{Trans}_{f \uparrow H}^{(t)}$ if and only if:

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

Use Corollary 3.8, it's equivalent to:

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

for a computable natural number N .

If we take:

$$\begin{aligned} \pi^{(0)}(\phi, \mathbf{f}, \mathbf{x}, t) &\doteq \phi(\mathbf{x}) < 0 \\ \pi^{(i)}(\phi, \mathbf{f}, \mathbf{x}, t) &\doteq \left(\bigwedge_{0 \leq j < i} \mathcal{L}_{t, f}^j \phi(\mathbf{x}) = 0 \right) \wedge \mathcal{L}_{t, f}^i \phi(\mathbf{x}) < 0 \quad i > 0 \end{aligned}$$

and let:

$$\pi(\phi, \mathbf{f}, \mathbf{x}, t) \doteq \bigvee_{0 \leq i \leq N} \pi^{(i)}(\phi, \mathbf{f}, \mathbf{x}, t)$$

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

$$\forall t((t \geq 0) \rightarrow (\neg \pi(\phi, \mathbf{f}, \mathbf{x}, t) \vee \pi(\mathbf{H}, \mathbf{f}, \mathbf{x}, t)))$$

which is equivalent to formula:

$$\forall t((t < 0) \vee (\pi(\phi, \mathbf{f}, \mathbf{x}, t) \wedge \neg \pi(\mathbf{H}, \mathbf{f}, \mathbf{x}, t)))$$

satisfies.

Now the second condition of Theorem 3.4 can be translated into:

$$\theta(\phi, \mathbf{H}, \mathbf{f}, \mathbf{x}, t) \doteq (\phi(\mathbf{x}) = 0) \rightarrow (\forall t((t < 0) \vee (\pi(\phi, \mathbf{f}, \mathbf{x}, t) \wedge \neg \pi(\mathbf{H}, \mathbf{f}, \mathbf{x}, t))))$$

■

The basic structure of our verifying algorithm is as follows:

- (i) Use the original or alternative algorithm to compute the upper bound N for ϕ and \mathbf{f} .
- (ii) Construct $\theta(\phi, \mathbf{H}, \mathbf{f}, \mathbf{x}, t)$ as in the proof of Theorem 3.12.
- (iii) Apply quantifier elimination algorithms on formula $\forall \mathbf{x}(\theta(\phi, \mathbf{H}, \mathbf{f}, \mathbf{x}, t) \wedge ((\mathbf{I}(\mathbf{x}) \geq 0) \rightarrow (\phi(\mathbf{x}) \geq 0)))$.
- (iv) If the result is *TRUE*, then $\phi(\mathbf{x}) \geq 0$ is an invariant; if the result is *FALSE*, then $\phi(\mathbf{x}) \geq 0$ is not an invariant.

3.6 Generate Invariants of Hybrid System: Simple Case

Using the notion of *template*, we can derive an algorithm to generate invariants of hybrid system from the algorithm we presented in last subsection. A template is a parametrized polynomial $p(\mathbf{u}, \mathbf{x}) \in \mathbb{R}[u_1, u_2, \dots, u_m, x_1, x_2, \dots, x_n]$ where u_1, u_2, \dots, u_m are parameters. Our target is to find an invariant $\phi(\mathbf{x})$ such that for some $\mathbf{u}_0 \in \mathbb{R}^m$, $p(\mathbf{u}_0, \mathbf{x}) = \phi(\mathbf{x})$, or report an error if no such invariant exists.

First we notice that Theorem 3.7 and related corollaries can be extended to situation when the number of parameters is more than one:

Lemma 3.13 *There exists $N \in \mathbb{N}$ which is independent to \mathbf{x} and t and \mathbf{u} such that $\gamma_{f, \phi}(\mathbf{x}, t) < \infty$ if and only if $\gamma_{f, \phi}(\mathbf{x}, t) \leq N$.*

Also, the original and alternative algorithm both work for multiple parameters. Now we give the structure of our generating algorithm:

- (i) Set a template $p(\mathbf{u}, \mathbf{x}) \in \mathbb{R}[\mathbf{u}, \mathbf{x}]$.
- (ii) Use the original or alternative algorithm to compute the upper bound N for p and \mathbf{f} .
- (iii) Construct $\theta(p, \mathbf{H}, \mathbf{f}, \mathbf{u}, \mathbf{x}, t)$ as in the proof of Theorem 3.12.

- (iv) Apply quantifier elimination algorithms on formula $\forall \mathbf{x}(\theta(p, \mathbf{H}, \mathbf{f}, \mathbf{u}, \mathbf{x}, t) \wedge ((\mathbf{I}(\mathbf{x}) \geq 0) \rightarrow (p(\mathbf{u}, \mathbf{x}) \geq 0)))$.
- (v) Consider the obtained quantifier-free formula $\Xi(\mathbf{u})$ as a constrain on \mathbf{u} . If $\Xi(\mathbf{u}) = \text{FALSE}$ then report an error. Otherwise use a constrain solver, such as DISCOVERER[23], to find a solution \mathbf{u}_0 .
- (vi) Let $\phi(\mathbf{x}) = p(\mathbf{u}_0, \mathbf{x})$, then $\phi(\mathbf{x}) \geq 0$ is an invariant.

4 General Case

In this section, we consider invariant verification and generation for general polynomial hybrid system. The domain and initial set will be:

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

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

and the possible invariant takes the form:

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

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

The basic idea is almost the same as that of last section. However, the complex boundaries of invariants, domain and initial set require some extra discussion. Strict proofs are often omitted as they are often repeats of what we did in last section.

4.1 Inward Set and Inverse Inward Set: Definitions and Properties

We define inward set as:

Definition 4.1 (Inward Set) Given a hybrid system $(\mathbf{H}, \mathbf{I}, \mathbf{f})$, the inward set of a set A is defined as:

$$\text{In}_f^{(t_0)}(A) \doteq \{\mathbf{x}_0 \in \mathbb{R}^n \mid \forall \mathbf{x} : \mathbf{x}(t_0) = \mathbf{x}_0 \wedge \exists \varepsilon > 0 \forall t \in (t_0, t_0 + \varepsilon), \mathbf{x}(t) \in A\}$$

where \mathbf{x} is a trajectory of the hybrid system.

Similarly, define inverse inward set as:

Definition 4.2 (Inverse Inward Set) Given a hybrid system $(\mathbf{H}, \mathbf{I}, \mathbf{f})$, the inverse inward set of a set A is defined as:

$$\text{IvIn}_f^{(t_0)}(A) \doteq \{\mathbf{x}_0 \in \mathbb{R}^n \mid \exists \mathbf{x} : \mathbf{x}(t_0) = \mathbf{x}_0 \wedge \exists \varepsilon > 0 \forall t \in (t_0 - \varepsilon, t_0), \mathbf{x}(t) \in A\}$$

where \mathbf{x} is a trajectory of the hybrid system.

Intuitively, $\mathbf{x}_0 \in \text{In}_f^{(t_0)}(A)$ means any trajectories that pass \mathbf{x}_0 at t_0 must stay in A for a positive time. While $\mathbf{x}_0 \in \text{IvIn}_f^{(t_0)}(A)$ means there exists at least one trajectory that passes \mathbf{x}_0 at t_0 and come from A .

The inward set and inverse inward set are useful when describing general continuous invariants:

Theorem 4.3 *Given a hybrid system $(\mathbf{H}, \mathbf{I}, \mathbf{f})$, a set Ψ is an invariant if and only if:*

- (i) $\mathbf{I} \subseteq \Psi$;
- (ii) $\forall t \geq 0, \forall \mathbf{x}(\mathbf{x} \in \Psi \cap \mathbf{H} \cap \text{In}_f^{(t)}(\mathbf{H}) \Rightarrow \mathbf{x} \in \text{In}_f^{(t)}(\Psi))$;
- (iii) $\forall t \geq 0, \forall \mathbf{x}(\mathbf{x} \in \Psi^c \cap \mathbf{H} \cap \text{IvIn}_f^{(t)}(\mathbf{H}) \Rightarrow \mathbf{x} \in (\text{IvIn}_f^{(t)}(\Psi))^c)$.

Proof The first condition of Definition 2.4 and the first condition of this theorem coincides, so we only need to consider the rest.

First let's consider the sufficient part. Recall Definition 2.4, if Ψ is not an invariant, then we have: there exists $t_0 \geq 0$, $\mathbf{x}_0 \in \Psi$ and $T_0 \geq 0$ such that:

$$(\forall t \in [t_0, t_0 + T_0], \mathbf{x}(t) \in \mathbf{H}) \not\Rightarrow (\forall t \in [t_0, t_0 + T_0], \mathbf{x}(t) \in \Psi)$$

which is equivalent to:

$$(\forall t \in [t_0, t_0 + T_0], \mathbf{x}(t) \in \mathbf{H}) \wedge (\exists t \in [t_0, t_0 + T_0], \mathbf{x}(t) \notin \Psi)$$

Take $t_z \in [t_0, t_0 + T_0]$ such that $\mathbf{x}(t_z) \notin \Psi$, and let

$$t_m = \inf\{t \in [t_0, t_z] \mid \mathbf{x}(t) \notin \Psi\}$$

By definition of t_m , $\mathbf{x}([t_0, t_m]) \subseteq \Psi$. Now consider $\mathbf{x}_m \doteq \mathbf{x}(t_m)$.

If $\mathbf{x}_m \notin \Psi$, then \mathbf{x}_m satisfies:

$$\mathbf{x}_m \in \Psi^c \cap \mathbf{H} \cap \text{IvIn}_f^{(t_m)}(\mathbf{H}) \cap \text{IvIn}_f^{(t_m)}(\Psi)$$

Which contradicts the third condition.

If $\mathbf{x}_m \in \Psi$, then $t_m < t_0 + T_0$, \mathbf{x}_m satisfies:

$$\mathbf{x}_m \in \Psi \cap \mathbf{H} \cap \text{In}_f^{(t_m)}(\mathbf{H}) \cap (\text{In}_f^{(t_m)}(\Psi))^c$$

Which contradicts the second condition.

Next we consider the necessary part. If the second condition doesn't hold, which is:

$$\exists t_0 \geq 0, \exists \mathbf{x}_0 \in \Psi \cap \mathbf{H} \cap \text{In}_f^{(t_0)}(\mathbf{H}) \cap (\text{In}_f^{(t_0)}(\Psi))^c$$

Recall Definition 4.1, that is to say: there exists a trajectory \mathbf{x} satisfying $\mathbf{x}(t_0) = \mathbf{x}_0$ such that: $\forall \varepsilon > 0 \exists t_\varepsilon \in (t_0, t_0 + \varepsilon) : \mathbf{x}(t_\varepsilon) \notin \Psi$, which contradicts the second condition of Definition 2.4.

If the third condition doesn't hold, which is:

$$\exists t_0 \geq 0, \exists \mathbf{x}_0 \in \Psi^c \cap \mathbf{H} \cap \text{IvIn}_f^{(t_0)}(\mathbf{H}) \cap \text{IvIn}_f^{(t_0)}(\Psi)$$

Recall Definition 4.2, that is to say: there exists a trajectory \mathbf{x} satisfying $\mathbf{x}(t_0) = \mathbf{x}_0$, $\exists \varepsilon > 0 \forall t \in (t_0 - \varepsilon, t_0) : \mathbf{x}(t) \in \Psi$, which also contradicts the second condition of Definition 2.4. ■

Inward set and inverse inward set have the following properties:

Theorem 4.4 For any $A, B \subseteq \mathbb{R}^n$,

- (i) If A is a open set, then: $A \subseteq \text{In}_f^{(t)}(A)$ and $A \subseteq \text{IvIn}_f^{(t)}(A)$ for any $t \in \mathbb{R}$.
- (ii) If $A \cap B = \emptyset$, then: $\text{In}_f^{(t)}(A) \cap \text{In}_f^{(t)}(B) = \emptyset$ and $\text{IvIn}_f^{(t)}(A) \cap \text{IvIn}_f^{(t)}(B) = \emptyset$ for any $t \in \mathbb{R}$.

Proof Proof can be obtained by checking definitions. Detailed proof is omitted. ■

4.2 Inward Set and Inverse Inward Set: Lie Derivatives

We want to use Lie derivatives to describe inward set and inverse inward set. To do this, Theorem 3.1 should be extended to include the inverse part of trajectories:

Theorem 4.5 Given polynomial ϕ and hybrid system $(\mathbf{H}, \mathbf{I}, \mathbf{f})$, $\gamma_{f,\phi}(\mathbf{x}_0, t) \neq 0$ if and only if $\mathbf{x}_0 \in S(\phi(\mathbf{x}) = 0)$, and if we take $\mathbf{x}(t_0) = \mathbf{x}_0$, then it follows that:

- (i) if $\gamma_{f,\phi}(\mathbf{x}_0, t_0) < \infty$ and $\mathcal{L}_{t_0, f}^{\gamma_{f,\phi}(\mathbf{x}_0, t_0)} \phi(\mathbf{x}_0) > 0$, then:

$$\exists \varepsilon > 0, \forall t \in (t_0, t_0 + \varepsilon), \phi(\mathbf{x}(t)) > 0$$

- (ii) if $\gamma_{f,\phi}(\mathbf{x}_0, t_0) < \infty$ and $\mathcal{L}_{t_0, f}^{\gamma_{f,\phi}(\mathbf{x}_0, t_0)} \phi(\mathbf{x}_0) < 0$, then:

$$\exists \varepsilon > 0, \forall t \in (t_0, t_0 + \varepsilon), \phi(\mathbf{x}(t)) < 0$$

- (iii) If $\gamma_{f,\phi}(\mathbf{x}_0, t_0) < \infty$ and $(-1)^{\gamma_{f,\phi}(\mathbf{x}_0, t_0)} \mathcal{L}_{t_0, f}^{\gamma_{f,\phi}(\mathbf{x}_0, t_0)} \phi(\mathbf{x}_0) > 0$, then:

$$\exists \varepsilon > 0, \forall t \in (t_0 - \varepsilon, t_0), \phi(\mathbf{x}(t)) > 0$$

- (iv) If $\gamma_{f,\phi}(\mathbf{x}_0, t_0) < \infty$ and $(-1)^{\gamma_{f,\phi}(\mathbf{x}_0, t_0)} \mathcal{L}_{t_0, f}^{\gamma_{f,\phi}(\mathbf{x}_0, t_0)} \phi(\mathbf{x}_0) < 0$, then:

$$\exists \varepsilon > 0, \forall t \in (t_0 - \varepsilon, t_0), \phi(\mathbf{x}(t)) < 0$$

- (v) if $\gamma_{f,\phi}(\mathbf{x}_0, t_0) = \infty$, then:

$$\exists \varepsilon > 0, \forall t \in (t_0 - \varepsilon, t_0 + \varepsilon), \phi(\mathbf{x}(t)) = 0$$

Proof The proof is exactly the same as Theorem 3.1. ■

Then, let's consider inward set and inverse inward set of sets defined by a single polynomial. For convenience, we write $\text{In}_f^{(t)}(\phi \triangleright 0)$, $\text{IvIn}_f^{(t)}(\phi \triangleright 0)$ to denote $\text{In}_f^{(t)}(S(\phi \triangleright 0))$ and $\text{IvIn}_f^{(t)}(S(\phi \triangleright 0))$. We have:

Theorem 4.6 Given a hybrid system $(\mathbf{H}, \mathbf{I}, \mathbf{f})$ and a polynomial ϕ , we have:

- $\text{In}_f^{(t)}(\phi > 0) = \Theta_+^{(t)}(\phi, \mathbf{f})$
- $\text{In}_f^{(t)}(\phi \geq 0) = \Theta_+^{(t)}(\phi, \mathbf{f}) \cup \Theta_0^{(t)}(\phi, \mathbf{f})$
- $\text{IvIn}_f^{(t)}(\phi > 0) = \Theta_-^{(t)}(\phi, \mathbf{f})$
- $\text{IvIn}_f^{(t)}(\phi \geq 0) = \Theta_-^{(t)}(\phi, \mathbf{f}) \cup \Theta_0^{(t)}(\phi, \mathbf{f})$

for any $t \in \mathbb{R}$.

where:

- $\Theta_+^{(t)}(\phi, \mathbf{f}) \doteq \{\mathbf{x} \in \mathbb{R}^n \mid \gamma_{\phi, \mathbf{f}}(\mathbf{x}, t) < \infty \wedge \mathcal{L}_{t, \mathbf{f}}^{\gamma_{\phi, \mathbf{f}}(\mathbf{x}, t)} \phi(\mathbf{x}) > 0\}$
- $\Theta_0^{(t)}(\phi, \mathbf{f}) \doteq \{\mathbf{x} \in \mathbb{R}^n \mid \gamma_{\phi, \mathbf{f}}(\mathbf{x}, t) = \infty\}$
- $\Theta_-^{(t)}(\phi, \mathbf{f}) \doteq \{\mathbf{x} \in \mathbb{R}^n \mid \gamma_{\phi, \mathbf{f}}(\mathbf{x}, t) < \infty \wedge (-1)^{\gamma_{\phi, \mathbf{f}}(\mathbf{x}, t)} \mathcal{L}_{t, \mathbf{f}}^{\gamma_{\phi, \mathbf{f}}(\mathbf{x}, t)} \phi(\mathbf{x}) > 0\}$

Proof Combine Theorem 4.5, Definition 4.1 and Definition 4.2, then apply the same technique used in the proof of Theorem 3.4. ■

For general semi-algebraic set

$$A \doteq \bigvee_{k=1}^K \bigwedge_{j=1}^{J_k} S(p_{kj} \triangleright 0)$$

we have:

Theorem 4.7

$$\begin{aligned} \text{In}_f^{(t)}(A) &= \bigvee_{k=1}^K \bigwedge_{j=1}^{J_k} \text{In}_f^{(t)}(p_{kj} \triangleright 0) \\ \text{IvIn}_f^{(t)}(A) &= \bigvee_{k=1}^K \bigwedge_{j=1}^{J_k} \text{IvIn}_f^{(t)}(p_{kj} \triangleright 0) \end{aligned}$$

for any $t \in \mathbb{R}$.

Proof Use Theorem 4.4. ■

Now we are ready to give the Lie derivative description of inward set and inverse inward set:

Theorem 4.8 Given a hybrid system $(\mathbf{H}, \mathbf{I}, \mathbf{f})$ and semi-algebraic set A :

$$A = \bigvee_{k=1}^K \left(\bigwedge_{j=1}^{j_k} S(p_{kj} \geq 0) \wedge \bigwedge_{j=j_k+1}^{J_k} S(p_{kj} > 0) \right)$$

we have:

$$\text{In}_f^{(t)}(A) = \bigvee_{k=1}^K \left(\bigwedge_{j=1}^{j_k} (\Theta_+^{(t)}(p_{kj}, \mathbf{f}) \cup \Theta_0^{(t)}(\phi, \mathbf{f})) \wedge \bigwedge_{j=j_k+1}^{J_k} \Theta_+^{(t)}(\phi, \mathbf{f}) \right)$$

and:

$$\text{IvIn}_f^{(t)}(A) = \bigvee_{k=1}^K \left(\bigwedge_{j=1}^{j_k} (\Theta_-^{(t)}(p_{kj}, \mathbf{f}) \cup \Theta_0^{(t)}(\phi, \mathbf{f})) \wedge \bigwedge_{j=j_k+1}^{J_k} \Theta_-^{(t)}(\phi, \mathbf{f}) \right)$$

for any $t \in \mathbb{R}$.

The definition of Θ_+ , Θ_- and Θ_0 can be found in Theorem 4.6.

Proof Combine Theorem 4.6 and Theorem 4.7. ■

4.3 Verify Invariants of Polynomial Hybrid System: General Case

In this subsection, we extend our verifying algorithm presented in last section to general semi-algebraic case. First we consider the situation when the possible invariant is a polynomial, note that we can still use algorithms of 3.3 and 3.4 to compute the upper bound N despite the complexity of boundaries. We have the following theorem:

Theorem 4.9 *Let:*

$$\begin{aligned} \pi_+^{(0)}(\phi, \mathbf{f}, \mathbf{x}, t) &\doteq \phi(\mathbf{x}) > 0 \\ \pi_+^{(i)}(\phi, \mathbf{f}, \mathbf{x}, t) &\doteq \left(\bigwedge_{0 \leq j < i} \mathcal{L}_{t,f}^j \phi(\mathbf{x}) = 0 \right) \wedge \mathcal{L}_{t,f}^i \phi(\mathbf{x}) > 0 \quad i > 0 \\ \pi_+(\phi, \mathbf{f}, \mathbf{x}, t) &\doteq \bigvee_{0 \leq i \leq N} \pi_+^{(i)}(\phi, \mathbf{f}, \mathbf{x}, t) \\ \pi_0(\phi, \mathbf{f}, \mathbf{x}, t) &\doteq \bigwedge_{0 \leq i \leq N} \mathcal{L}_{t,f}^i \phi(\mathbf{x}) = 0 \\ \pi_-^{(0)}(\phi, \mathbf{f}, \mathbf{x}, t) &\doteq \phi(\mathbf{x}) < 0 \\ \pi_-^{(i)}(\phi, \mathbf{f}, \mathbf{x}, t) &\doteq \left(\bigwedge_{0 \leq j < i} \mathcal{L}_{t,f}^j \phi(\mathbf{x}) = 0 \right) \wedge \mathcal{L}_{t,f}^i \phi(\mathbf{x}) < 0 \quad i > 0 \\ \pi_-(\phi, \mathbf{f}, \mathbf{x}, t) &\doteq \bigvee_{0 \leq i \leq N} \pi_-^{(i)}(\phi, \mathbf{f}, \mathbf{x}, t) \end{aligned}$$

Then:

- $\mathbf{x} \in \text{In}_f^{(t)}(\phi > 0)$ if and only if formula $\pi_+(\phi, \mathbf{f}, \mathbf{x}, t)$ satisfies.
- $\mathbf{x} \in \text{In}_f^{(t)}(\phi \geq 0)$ if and only if formula $\pi_{+,0}(\phi, \mathbf{f}, \mathbf{x}, t) \doteq \pi_+(\phi, \mathbf{f}, \mathbf{x}, t) \vee \pi_0(\phi, \mathbf{f}, \mathbf{x}, t)$ satisfies.
- $\mathbf{x} \in \text{IvIn}_f^{(t)}(\phi > 0)$ if and only if formula $\pi_-(\phi, \mathbf{f}, \mathbf{x}, t)$ satisfies.
- $\mathbf{x} \in \text{IvIn}_f^{(t)}(\phi \geq 0)$ if and only if formula $\pi_{-,0}(\phi, \mathbf{f}, \mathbf{x}, t) \doteq \pi_-(\phi, \mathbf{f}, \mathbf{x}, t) \vee \pi_0(\phi, \mathbf{f}, \mathbf{x}, t)$ satisfies.

Proof Use the same technique as Theorem 3.12 to translate the conclusion of Theorem 4.6 into finite length formula. ■

Now for general semi-algebraic invariants:

Theorem 4.10 *Given a hybrid system $(\mathbf{H}, \mathbf{I}, \mathbf{f})$ and semi-algebraic set Ψ where:*

$$\begin{aligned}\mathbf{H} &= \bigvee_{k=1}^K \left(\bigwedge_{j=1}^{j_k} (p_{kj} \geq 0) \wedge \bigwedge_{j=j_k+1}^{J_k} (p_{kj} > 0) \right) \\ \Psi &= \bigvee_{m=1}^M \left(\bigwedge_{l=1}^{l_m} (p_{ml} \geq 0) \wedge \bigwedge_{l=l_m+1}^{L_m} (p_{ml} > 0) \right)\end{aligned}$$

and let $\mathcal{I}, \mathcal{H}, \Psi$ denote the formula that defines set $\mathbf{I}, \mathbf{H}, \Psi$ respectively. By our assumption, $\mathbf{I}, \mathbf{H}, \Psi$ are semi-algebraic sets, so $\mathcal{I}, \mathcal{H}, \Psi$ are boolean combinations of polynomial formulas.

We have: Ψ is an invariant of the hybrid system if and only if formula:

$$\begin{aligned}\forall \mathbf{x} : \mathcal{I} &\rightarrow \Psi \\ \forall t : (t < 0) &\vee (\forall \mathbf{x} : \Psi \wedge \mathcal{H} \wedge \xi_{+,H} \rightarrow \xi_{+,\Psi}) \\ \forall t : (t < 0) &\vee (\forall \mathbf{x} : \neg \Psi \wedge \mathcal{H} \wedge \xi_{-,H} \rightarrow \neg \xi_{-,\Psi})\end{aligned}$$

all satisfy.

Where:

$$\begin{aligned}\xi_{+,H} &\doteq \bigvee_{k=1}^K \left(\bigwedge_{j=1}^{j_k} \pi_{+,0}(p_{kj}, \mathbf{f}, \mathbf{x}, t) \wedge \bigwedge_{j=j_k+1}^{J_k} \pi_{+}(p_{kj}, \mathbf{f}, \mathbf{x}, t) \right) \\ \xi_{+,\Psi} &\doteq \bigvee_{m=1}^M \left(\bigwedge_{l=1}^{l_m} \pi_{+,0}(p_{ml}, \mathbf{f}, \mathbf{x}, t) \wedge \bigwedge_{l=l_m+1}^{L_m} \pi_{+}(p_{ml}, \mathbf{f}, \mathbf{x}, t) \right) \\ \xi_{-,H} &\doteq \bigvee_{k=1}^K \left(\bigwedge_{j=1}^{j_k} \pi_{-,0}(p_{kj}, \mathbf{f}, \mathbf{x}, t) \wedge \bigwedge_{j=j_k+1}^{J_k} \pi_{-}(p_{kj}, \mathbf{f}, \mathbf{x}, t) \right) \\ \xi_{-,\Psi} &\doteq \bigvee_{m=1}^M \left(\bigwedge_{l=1}^{l_m} \pi_{-,0}(p_{ml}, \mathbf{f}, \mathbf{x}, t) \wedge \bigwedge_{l=l_m+1}^{L_m} \pi_{-}(p_{ml}, \mathbf{f}, \mathbf{x}, t) \right)\end{aligned}$$

Proof Use the conclusion of Theorem 4.10 and Theorem 4.7 to translate the conclusion of Theorem 4.3. ■

Now we can give the basic structure of the verifying algorithm:

- (i) Use the original or alternative algorithm to compute the upper bound N for every p_{ml} and p_{kj} .
- (ii) Construct the formula in Theorem 4.10.
- (iii) Apply quantifier elimination algorithms on that formula.

- (iv) If the result is *TRUE*, then Ψ is an invariant; if the result is *FALSE*, then Ψ is not an invariant.

4.4 Generate Invariants of Hybrid System: General Case

We also give the generating algorithm for general case:

- (i) Set semi-algebraic invariant template:

$$\Psi(\mathbf{u}, \mathbf{x}) = \bigvee_{m=1}^M \left(\bigwedge_{l=1}^{l_m} (p_{ml}(\mathbf{u}, \mathbf{x}) \geq 0) \wedge \bigwedge_{l=l_m+1}^{L_m} (p_{ml}(\mathbf{u}, \mathbf{x}) > 0) \right)$$

for some $p_{ml} \in \mathbb{R}[\mathbf{u}, \mathbf{x}]$.

- (ii) Use the original or alternative algorithm to compute the upper bound N for every p_{ml} and p_{kj} .
- (iii) Construct the formula in Theorem 4.10.
- (iv) Apply quantifier elimination algorithms on that formula.
- (v) Consider the obtained quantifier-free formula $\Xi(\mathbf{u})$ as a constrain on \mathbf{u} . If $\Xi(\mathbf{u}) = \text{FALSE}$ then report an error. Otherwise use a constrain solver, such as DISCOVERER[23], to find a solution \mathbf{u}_0 .
- (vi) Let $\Psi_0(\mathbf{x}) = \Psi(\mathbf{u}_0, \mathbf{x})$, then $\Psi_0(\mathbf{x})$ is an invariant.

5 Conclusion

In this article we propose a sound and complete algorithm to verify invariants of polynomial non-autonomous hybrid systems. From it we derive a sound and relatively complete algorithm to automatically generate invariants for such systems. “Relatively” means invariant generation needs a pre-defined template.

Quantifier elimination in real closed field play a critical role in our approach. Since the problem has a double exponential time nature[3], time consumption of our algorithm grows very quickly as the number of variables grows. Still, besides its theoretical meanings, our approach has been proved to be a possible choice for automatic invariant generation in small systems.

The chain of ideals generated by Lie derivatives has finite length. A primitive recursive upper bound of that length can be found[7, 21]. This upper-bound is useful in constructive proof and time complexity analysis, but it’s often too large to be practical comparing to the upper bound computed by the original and alternative algorithm presented in 3.3 and 3.4. Thus it’s not included in previous sections.

Further problems: the use of heuristic strategy in choosing template and verifying invariants; finding sub-problems where our approach can be more competitive.

References

- [1] Rajeev Alur, Costas Courcoubetis, Thomas A. Henzinger, and Pei-Hsin Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid Systems*, pages 209–229. Springer, 1993.
- [2] Michael S. Branicky. General hybrid dynamical systems: Modeling, analysis, and control. In Rajeev Alur, Thomas A. Henzinger, and Eduardo D. Sontag, editors, *Hybrid Systems III*, volume 1066 of *LNCS*, pages 186–200. Springer Berlin Heidelberg, 1996.
- [3] Christopher W Brown and James H Davenport. The complexity of quantifier elimination and cylindrical algebraic decomposition. In *Proceedings of the 2007 international symposium on Symbolic and algebraic computation*, pages 54–60. ACM, 2007.
- [4] Edmund Clarke, Ansgar Fehnker, Zhi Han, Bruce Krogh, Joël Ouaknine, Olaf Stursberg, and Michael Theobald. Abstraction and counterexample-guided refinement in model checking of hybrid systems. *International Journal of Foundations of Computer Science*, 14(04):583–604, 2003.
- [5] David Cox, John Little, and Donal O’shea. *Ideals, varieties, and algorithms*, volume 3. Springer, 1992.
- [6] J.M. Davoren and Anil Nerode. Logics for hybrid systems. In *Proc. of the IEEE*, volume 88, pages 985–1010. Springer-Verlag, 2000.
- [7] Diego Figueira, Santiago Figueira, Sylvain Schmitz, and Philippe Schnoebelen. Ackermannian and primitive-recursive bounds with dickson’s lemma. In *Logic in Computer Science (LICS), 2011 26th Annual IEEE Symposium on*, pages 269–278. IEEE, 2011.
- [8] T.A. Henzinger. The theory of hybrid automata. In *LICS’96*, pages 278–292, July 1996.
- [9] Thomas A. Henzinger and Joseph Sifakis. The embedded systems design challenge. In *FM’06*, pages 1–15. Springer, 2006.
- [10] Hui Kong, Sergiy Bogomolov, Christian Schilling, Yu Jiang, and Thomas A Henzinger. Invariant clusters for hybrid systems. *arXiv preprint arXiv:1605.01450*, 2016.
- [11] Edward A. Lee. What’s ahead for embedded software? *IEEE Computer*, 33(9):18–26, 2000.
- [12] Jiang Liu, Jidong Lv, Zhao Quan, Naijun Zhan, Hengjun Zhao, Chaochen Zhou, and Liang Zou. A calculus for hybrid CSP. In *APLAS’10*, volume 6461 of *LNCS*, pages 1–15. Springer, 2010.
- [13] Jiang Liu, Naijun Zhan, and Hengjun Zhao. Computing semi-algebraic invariants for polynomial dynamical systems. In *Proceedings of the ninth ACM international conference on Embedded software*, pages 97–106. ACM, 2011.
- [14] Zohar Manna and Amir Pnueli. Verifying hybrid systems. In Robert Grossman, Anil Nerode, Anders Ravn, and Hans Rischel, editors, *Hybrid Systems*, volume 736 of *LNCS*, pages 4–35. Springer, 1993.
- [15] A. Platzer. Differential-algebraic dynamic logic for differential-algebraic programs. *J. Log. and Comput.*, 20(1):309–352, February 2010.
- [16] André Platzer. Differential dynamic logic for hybrid systems. *J. Autom. Reason.*, 41(2):143–189, 2008.
- [17] André Platzer. *Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics*. Springer, Heidelberg, 2010.
- [18] André Platzer. The complete proof theory of hybrid systems. In *LICS’12*, pages 541–550, Wash-

- ington, DC, USA, 2012. IEEE Computer Society.
- [19] Stephen Prajna and Ali Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *Hybrid Systems: Computation and Control*, pages 477–492. Springer, 2004.
 - [20] Sriram Sankaranarayanan, Henny B Sipma, and Zohar Manna. Constructing invariants for hybrid systems. In *Hybrid Systems: Computation and Control*, pages 539–554. Springer, 2004.
 - [21] Guillermo Moreno Socias. Length of polynomial ascending chains and primitive recursiveness. *Mathematica Scandinavica*, 71:181–205, 1992.
 - [22] Morris Tenenbaum and Harry Pollard. *Ordinary differential equations: An elementary textbook for students of mathematics, engineering, and the sciences*, chapter 9, pages 562–563. Courier Corporation, 1963.
 - [23] Bican Xia. Discoverer: a tool for solving semi-algebraic systems. *ACM Communications in Computer Algebra*, 41(3):102–103, 2007.