

Article

Approximate Completed Trace Equivalence of ILAHSs Based on SAS Solving

Honghui He ^{1,2,3,4} , Jinzhao Wu ^{1,2,3,*}, Juxia Xiong ^{1,2,3}

¹ Chengdu Institute of Computer Application, Chinese Academy of Sciences, Chengdu 610041, China; HongHuiHe@vip.henu.edu.cn (H.H.); xiongjuxia@gxun.edu.cn (J.X.)

² University of Chinese Academy of Sciences, Beijing 100049, China

³ Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, Guangxi University for Nationalities, Nanning 530006, China

⁴ School of Computer Software, Henan University, Kaifeng 475001, China

* Correspondence: wujinzhao@gxun.edu.cn

Received: 6 September 2019; Accepted: 23 October 2019; Published: 1 November 2019



Abstract: The ILAHS (inhomogeneous linear algebraic hybrid system) is a kind of classic hybrid system. For the purpose of optimizing the design of ILAHS, one important strategy is to introduce equivalence to reduce the states. Recent advances in the hybrid system indicate that approximate trace equivalence can further simplify the design of ILAHS. To address this issue, the paper first introduces the trajectory metric d_{trj} for measuring the deviation of two hybrid systems' behaviors. Given a deviation $\varepsilon \geq 0$, the original ILAHS of \mathcal{H}_1 can be transformed to the approximate ILAHS of \mathcal{H}_2 , then in trace equivalence semantics, \mathcal{H}_2 is further reduced to \mathcal{H}_3 with the same functions, and hence \mathcal{H}_1 is ε -approximate trace equivalent to \mathcal{H}_3 . In particular, $\varepsilon = 0$ is a traditional trace equivalence. We implement an approach based on *RealRootClassification* to determine the approximation between the ILAHSs. The paper also shows that the existing approaches are only special cases of our method. Finally, we illustrate the effectiveness and practicality of our method on an example.

Keywords: approximate equivalence; hybrid system; inhomogeneous linear algebraic hybrid systems; RealRootClassification

1. Introduction

In the real world, there exist many systems exhibiting a hybrid discrete–continuous behavior, and the notion of hybrid automaton [1] was introduced to model these hybrid systems. For example, embedded systems are often modeled as hybrid systems due to their involvement of both digital control software and analog plants, which physical process is often specified in the form of differential equations. Infinite states (or state explosion) due to the continuous behavior of the system are among the most challenging problems in verifying hybrid systems. In the traditional functional analysis domain, an efficient classical technique to tackle state explosion is equivalence [2,3]. In brief, the systems with the same functional behaviors are called equivalence. Functional equivalence simplifies the system by removing the duplicate branches. The papers by Glabbeek [2,3] proposed fourteen kinds of linear time-branching time equivalence relations, and the completed trace equivalence is a basic state-space equivalence relation, which can be used to further reduce the states of hybrid systems [4,5].

However, traditional equivalence techniques are based on absolute precision, which only gives the answer “yes” or “no”, not how much simulation. There is an example that illustrates that it is meaningful to introduce the deviation analysis to equivalence. Considering a microwave oven, if the microwave oven is required to heat to 100 °C within 60 s but does so within 60.5 s (this is very common

in life), and if a 0.5 s deviation is allowable, then the two systems are equivalent, or the two states can be merged. Otherwise, the two systems are not equivalent, or the two states cannot be merged. Especially in safety-critical cases, any deviation greater than zero seconds can be viewed as the presence of a bug. In reality, the parameters of hybrid systems are obtained by all kinds of measurements, which generally have a small number of errors. Since errors are inevitable, approximation analysis between the original system and the approximate system is very significant for the given permission precision.

In order to achieve this goal, this paper proceeds by defining two types of metrics, state metric d_{sta} and trajectory metric d_{trj} , to check how much a hybrid system \mathcal{H}_1 conforms to another hybrid system \mathcal{H}_2 . The d_{sta} measures the distance of the states, while the d_{trj} specifies the deviation of the system behaviors. Given a deviation $\varepsilon \geq 0$, the original ILAHS of \mathcal{H}_1 can be transformed to the approximate ILAHS of \mathcal{H}_2 . Then in trace equivalence semantics, \mathcal{H}_2 is further reduced to \mathcal{H}_3 with the same functions, and hence \mathcal{H}_1 is ε -approximate trace equivalent to \mathcal{H}_3 . In particular, $\varepsilon = 0$ is a traditional trace equivalence. At last, the decision problem that \mathcal{H}_1 is ε -approximate to \mathcal{H}_2 can be reduced to semi-algebraic system solving (SAS solving) [6–11].

There are several choices for metrics, such as bisimulation metrics [12,13], Lyapunov-like functions [13], directed metrics [14], and pseudo-ultrametrics [15], most of which are required to satisfy intricate properties of simulation or bisimulation relations. In contrast, state metric d_{sta} and trajectory metric d_{trj} in this paper are based on Euclidean distance, which is an intuitive metric.

In this paper, we focus on a specific class of hybrid systems, namely, inhomogeneous linear algebraic hybrid systems (ILAHSs for short). It is well known that the most important analysis question for hybrid systems is the problem of reachability, which is computationally hard and undecidable for the general case and intractable even for the simplest subclasses [1]. So it is doomed to be impossible to find a universal approach for this question without any simplifications or restrictions. ILAHSs, which are ubiquitous in reality, are a kind of classical hybrid systems with some simplifications.

Note that our approximation technology differs from those proposed in the papers [16–20], which aim to construct an approximately reachable set. Most of the differential equations of hybrid systems, as we all know, cannot be solved analytically, and, therefore, the exact reachable set cannot be computed symbolically, so many papers developed a variety of methods that over-approximate or under-approximate the reachable set using varieties of set representations such as polyhedra [21], zonotopes [22], level sets, or ellipsoids [23]. Our approximation technology is, however, to identify the approximate behavior equivalence among the hybrid systems. So far, there exist few papers focused on the approximation analysis of the ILAHS' behavior equivalence [4,5,24]. Unfortunately, their methods are based on *Matrix Jordan Standard Type*, which only apply to special cases and cannot deal with the infinite trajectory condition, whereas our method can solve this problem.

The main contributions of this paper are as follows. First, we propose a new method for identifying equivalence, which applies to more general ILAHS. Second, not only is the finite trajectory condition defined, but the infinite trajectory condition is also defined. Third, compared to existing approaches, our method is based on Euclidean distance, which is an intuitive metric.

The remainder of this paper is organized as follows. Section 2 recalls the preliminaries of our method, including hybrid system, inhomogeneous linear algebraic hybrid systems, SAS, and so on. We define the approximate completed trace equivalence of ILAHS in Section 3. Section 4 presents a case study. Conclusions are given in Section 5.

2. Preliminaries

In this section, we recall some concepts used throughout the paper. We first clarify some notation conventions. We use bold uppercase letters such as A to denote matrices and A^T, \tilde{A} denote the transpose and approximate matrix of A , respectively. We use $\vec{b} = (b_1, \dots, b_n)$ to denote vectors. We denote $(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)$ as an assignment of variables $\vec{x} = (x_1, \dots, x_n)$.

Hybrid automata, first proposed by Rajeev Alur et al. [25], are a mathematical model to describe a system containing continuous and discrete components. Many other models for hybrid systems can

be found in [26–29]. In this paper, we adopt the model proposed in [1] as our modeling framework. A hybrid system can then be defined as:

Definition 1 (Hybrid System). A hybrid system $\mathcal{H} : \langle L, V, \mathcal{T}, F, \ell_0, \Theta \rangle$ consists of the following components:

- L : a finite set of locations (or modes).
- V : a set of real-valued system variables. The hybrid state space is denoted by $\Sigma = L \times \mathbb{R}^{|V|}$, a state is denoted by $(\ell, \vec{s}) \in \Sigma$, and $\vec{s} \in \mathbb{R}^{|V|}$ is a continuous state of the variables over the real numbers.
- $\mathcal{T} \subseteq L \times L \times 2^V \times (V' \mapsto V)$: a set of discrete transitions. A discrete transition $\tau = \langle \ell_1, \ell_2, \rho_\tau, \alpha_\tau \rangle$ consists of the pre- and post-locations of the transition ℓ_1, ℓ_2 , a guard ρ_τ , which is a boolean function of the variables V , and an action α_τ , which is an assignment over the variables V and V' . V denotes the current-state variables, and V' denotes the next-state variables.
- $F: L \mapsto (V \mapsto V)$: continuous evolution, a map that maps each location $\ell \in L$ to a differential rule $F(\ell)$, of the form $\dot{v}_i = f_i(\vec{v})$. The differential rule $F(\ell)$ specifies how the system variables evolve at the location ℓ , which is also known as a vector field or a flow field.
- $I: L \mapsto 2^V$: a map that maps each location $\ell \in L$ to a location condition (location invariant) that is an assertion over V .
- $\ell_0 \in L$: the initial location.
- Θ : an assertion specifying the initial condition.

The behaviors of hybrid systems are expressed as trajectories.

Definition 2 (Trajectory). A trajectory of a hybrid system \mathcal{H} is an (in)finite sequence of states $\langle \ell, \hat{v} \rangle \in L \times \mathbb{R}^{|V|}$ of the form

$$\langle \ell_0, \hat{v}_0 \rangle, \langle \ell_1, \hat{v}_1 \rangle, \langle \ell_2, \hat{v}_2 \rangle, \dots,$$

such that,

Initiation: $\hat{v}_0 \models \Theta$ specifies an initial state. Furthermore, for each consecutive state pair $\langle (\ell_i, \hat{v}_i), (\ell_{i+1}, \hat{v}_{i+1}) \rangle$, one of the two evolution conditions below is satisfied:

Discrete evolution: There exists a transition $\tau : \langle \ell_1, \ell_2, \rho_\tau, \alpha_\tau \rangle \in \mathcal{T}$ such that τ is enabled, i.e., $\rho_\tau(\hat{v}_i) = \text{true}$ and $\hat{v}_{i+1} = \alpha_\tau(\hat{v}_i)$.

Continuous evolution: $\ell_i = \ell_{i+1} = \ell$, and there exists a time interval $\delta > 0$, $\hat{v}_i \xrightarrow{\delta} \hat{v}_{i+1}$, along with a smooth (continuous and differentiable to all orders) function $g : [0, \delta] \mapsto \mathbb{R}^{|V|}$ such that g evolves from \hat{v}_i to \hat{v}_{i+1} according to the differential rule at location ℓ , while satisfying the location condition $I(\ell)$. Formally,

1. $g(0) = \hat{v}_i, g(\delta) = \hat{v}_{i+1}$ and $\forall t \in [0, \delta], g(t) \models I(\ell)$;
2. $\forall t \in [0, \delta], \langle g(t), \dot{g}(t) \rangle \models F(\ell)$.

A state $\langle \ell, \hat{v} \rangle$ is reachable if it appears in some trajectory of \mathcal{H} . The set of all reachable states of \mathcal{H} is denoted by $\text{Reach}(\mathcal{H})$.

With some restrictions, a hybrid system can be specialized into an inhomogeneous linear algebraic hybrid system, which is ubiquitous in reality [4,5,24].

Definition 3 (Inhomogeneous linear algebraic hybrid system, ILAHS). ILAHS is a kind of hybrid system with simplifications: for continuous evolution, $F(\ell) : \vec{v}' = \mathbf{A}\vec{v}$, where $\vec{v}' \xrightarrow{\delta} \vec{v}$.

Example 1 (A thermostat). Consider a room being heated by a radiator controlled by a thermostat, one of the typical introductory examples of hybrid systems. This system has both a continuous state and two discrete

states. The continuous state is the temperature in the room $x \in \mathbb{R}$. The discrete states, $L = \{on, off\}$, reflect whether the radiator is on or off. The evolution of x is governed by a differential equation, while the evolution of L is through jumps. It is very convenient to compactly describe such hybrid systems by mixing the differential equation with the directed graph notation (shown in Figure 1).

Computers are used as control systems for a wide variety of industrial and consumer devices. However, the working principle of computers is fundamentally discrete rather than continuous. When the time x is discretized, the general hybrid system (Figure 1) can be turned into an ILAHS (Figure 2), where

$$\begin{cases} L = \{on, off\}, \\ V = \{x\}, \\ \mathcal{T} = \{\langle on, \alpha_1, off \rangle, \langle off, \alpha_2, on \rangle\}, \alpha_{1,2} = \{x' = x\}, \\ F(on) = \{x' = (1 + K \cdot h \cdot \Delta t)x\}, \\ F(off) = \{x' = (1 - K \cdot \Delta t)x\}. \end{cases} \quad (1)$$

The ILAHS' behaviors are expressed as trajectories. For instance, we assume that $m = 15$, $m_{on} = 20$, $M = 32$, $M_{off} = 30$, $\Delta t = 0.1$, $K = 0.2$, and $h = 5$. If the initial temperature of the room is 25°C , then one possible trajectory is as follows (Figure 3):

$$\langle off, 25 \rangle \rightarrow \langle off, 24.5 \rangle \rightarrow \langle off, 24.01 \rangle \rightarrow \dots \rightarrow \langle on, 15.09 \rangle \rightarrow \langle on, 16.60 \rangle \rightarrow \dots$$

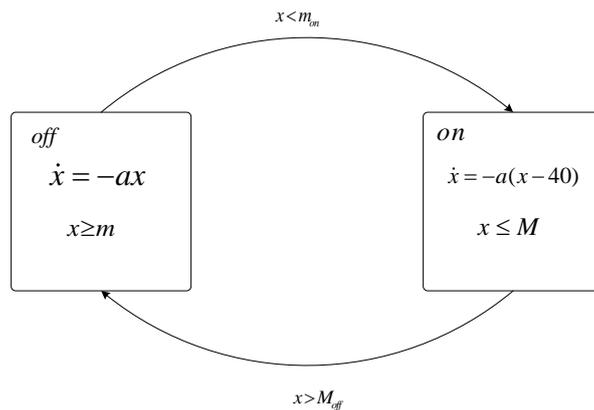


Figure 1. Automaton of a thermostat.

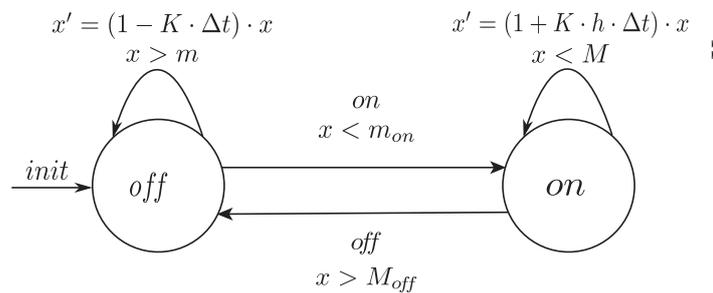


Figure 2. Inhomogeneous linear algebraic hybrid system (ILAHS) of a thermostat.

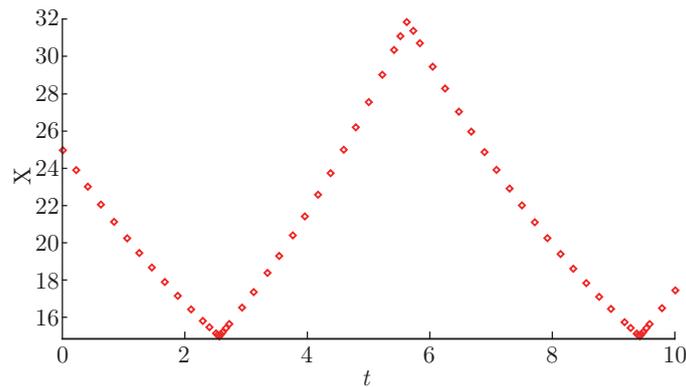


Figure 3. An ILAHS trajectory.

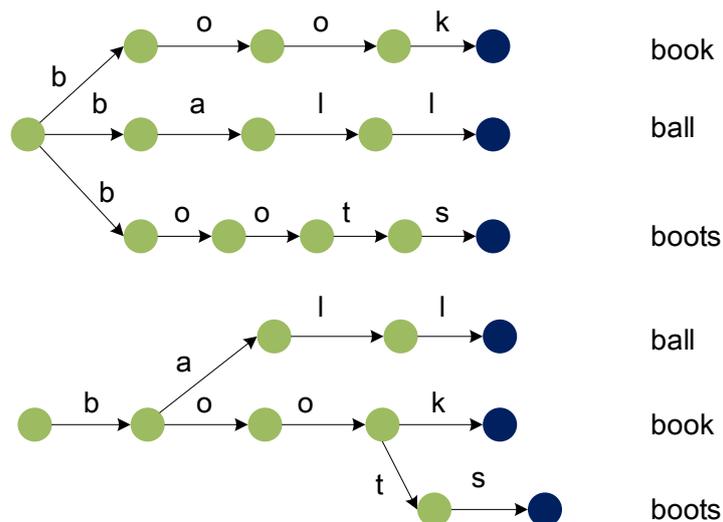
Definition 4 (Metric). For a set X , a metric on X is a function $d_X : X \times X \mapsto \mathbb{R}^+$ such that

1. for all $x, y \in X, d_X(x, y) = 0$ if and only if $x = y$;
2. $d_X(x, y) = d_X(y, x)$, for all $x, y \in X$; and
3. $d_X(x, z) \leq d_X(x, y) + d_X(y, z)$ for all $x, y, z \in X$.

Note that the condition (1) expresses that $d_X(x, y) = 0$ implies $x = y$. If $d_X(x, y) = 0$ does not necessarily mean that $x = y$, the metric is called a pseudo-metric [12].

Definition 5 (Trace Equivalence). $\delta \in Act^*$ is a trace of a process p , if there exists a process q , such that $p \xrightarrow{\delta} q$. Let $T(p)$ denote the set of traces of p . Two processes p and q are trace equivalent if $T(p) = T(q)$. In trace semantics, two processes are identified iff they are trace equivalent.

Trace semantics is based on the idea that two processes are to be identified if they allow the same set of observations, where an observation simply consists of a sequence of actions performed by the process in succession (Figure 4). More details can be found in the paper [2].



Automata in language processing and behavioral equivalence

Figure 4. A trace equivalence example.

However, *Act* consists of abstract actions, which are unable to express the exchange details of the data stream. In 2008, Sankaranarayanan et al. [30] defined a hybrid system program model with polynomial equations that took the place of abstract actions and opened up a new method of hybrid system design and verification based on polynomial algebra. For example, the transition $l_0 \xrightarrow{\alpha} l_1$ with $\alpha \equiv (x - 1)^2 + (x' - 2)^2 + (y' - 3)^2 = 0$ indicates the relation $\{ \langle (l_0, s_0), (l_1, s_1) \rangle \mid s_0 = (1, y) \wedge s_1 = (2, 3), y \in \mathbb{R} \}$, the variable x refers to the current state, and x' to the next state of a transition.

Definition 6 (Completed Trace Equivalence). $\delta \in Act^*$ is a completed trace of a process p , if there exists a process q , such that $p \xrightarrow{\delta} q$ and $I(q) = \emptyset$. Let $CT(p)$ denote the set of completed traces of p . Two processes p and q are completed trace equivalent if $T(p) = T(q)$ and $CT(p) = CT(q)$. In completed trace semantics, two processes are identified iff they are completed trace equivalent (Figure 5).

Obviously, the trace equivalence can be regarded as a simpler version of the completed trace equivalence.

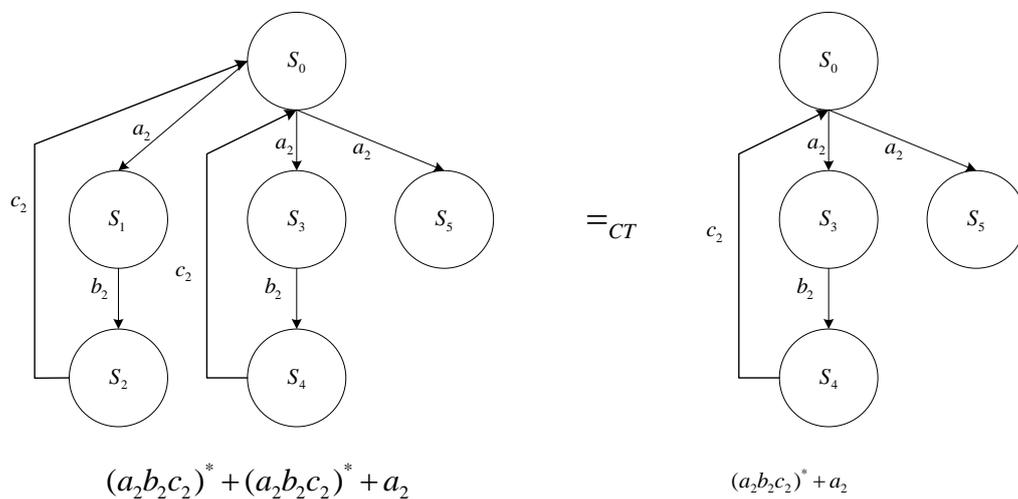


Figure 5. A completed trace equivalence example.

Definition 7 (Semi-Algebraic System). A semi-algebraic system (SAS for short) is a conjunctive polynomial formula of the following form:

$$\begin{cases} p_1(\vec{u}, \vec{x}) = 0, \dots, p_r(\vec{u}, \vec{x}) = 0, \\ g_1(\vec{u}, \vec{x}) \leq 0, \dots, g_k(\vec{u}, \vec{x}) \leq 0, \\ g_{k+1}(\vec{u}, \vec{x}) > 0, \dots, g_l(\vec{u}, \vec{x}) > 0, \\ h_1(\vec{u}, \vec{x}) \neq 0, \dots, h_m(\vec{u}, \vec{x}) \neq 0, \end{cases} \quad (2)$$

where $r \geq 1, l \geq k \geq 0, m \geq 0, \vec{u} = (u_1, \dots, u_t), \vec{x} = (x_1, \dots, x_r)$, and p_i, g_i, h_i are all in $\mathbb{Q}[\vec{u}, \vec{x}] \setminus \mathbb{Q}$. An SAS is usually denoted by a quadruple $[\mathbb{P}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{H}]$, where $\mathbb{P} = [p_1, \dots, p_r], \mathbb{G}_1 = [g_1, \dots, g_k], \mathbb{G}_2 = [g_{k+1}, \dots, g_l]$ and $\mathbb{H} = [h_1, \dots, h_m]$. An SAS is called parametric if $t \neq 0$, otherwise it is constant.

For SAS, what we are concerned about is the real solutions of the equations(\mathbb{P}) under the constraints ($\mathbb{G}_1, \mathbb{G}_2, \mathbb{H}$). More specifically, assuming S is a constant SAS, the interesting questions are how to compute the number of real solutions of S , and if the number is finite, how to compute these real solutions. Assuming S is parametric SAS, the interesting problem is the so-called real solution classification, which is to determine the condition on the parameters such that the system has the prescribed number of distinct real solutions, which is possibly infinite.

To address this issue, Yang et al. [6] first defined the key concept of border polynomial(BP), based on which an algorithm is proposed. For more details, please refer to [7–11]. This algorithm has been improved and implemented by Xia as the Maple package DISCOVERER [31]. Since 2009, the main functions of DISCOVERER have been integrated into the RegularChains library of Maple. Since then, the implementation has been improved by Chen et al. [32–34]. Thus, the experiment in this paper requires a version of Maple higher than Maple 13.

Example 2. Prove that $f \geq 0$ under the constraints $a \geq 0, b \geq 0, c \geq 0, abc - 1 = 0$, i.e.,

$$\left. \begin{array}{l} a \geq 0 \\ b \geq 0 \\ c \geq 0 \\ abc - 1 = 0 \end{array} \right\} \models f \geq 0. \tag{3}$$

In other words,

$$\left\{ \begin{array}{l} a \geq 0 \\ b \geq 0 \\ c \geq 0 \\ abc - 1 = 0 \\ f < 0 \end{array} \right. \text{ has no solution} \tag{4}$$

where $f = 2b^4c^4 + 2b^3c^4a + 2b^4c^3a + 2a^3c^3b^2 + 2a^4c^3b + 2a^3c^4b + 2a^4c^4 + 2a^3b^4c + 2a^4b^4 + 2a^3b^3c^2 + 2a^4b^3c - 3b^5c^4a_6^3b^4c^4a^4 - 3b^5c^3a^4 - 3b^4c^3a^5 - 3b^4c^5a^3 - 3b^3c^5a^4 - 3b^3c^4a^5$.

In order to prove (4) with Maple, we first start Maple and load two relative packages of RegularChains as follows.

```
> with(RegularChains);
> with(ParametricSystemTools);
> with(SemiAlgebraicSetTools);
```

Then we define an order of the unknowns:

```
> R := PolynomialRing([a, b, c]);
```

Then, by calling

```
> RealRootClassification([abc-1],[a,b,c], [-f], [ ], 2, 0, R);
```

we will know at once that the inequality holds.

3. ϵ -Approximate Completed Trace Equivalence of ILAHS

In this section, we define ϵ -approximate completed trace equivalence of ILAHS and propose the discriminant conditions based on SAS. Assuming \mathcal{H} is an ILAHS and \mathfrak{T} is a trajectory of \mathcal{H} ,

$$\langle \ell_0, \hat{v}_{00} \rangle \xrightarrow{A_0} \langle \ell_0, \hat{v}_{01} \rangle \xrightarrow{A_0} \dots \langle \ell_0, \hat{v}_{0m_0} \rangle \xrightarrow{lab_{01}} \langle \ell_1, \hat{v}_{10} \rangle, \dots, \xrightarrow{A_n} \langle \ell_n, \hat{v}_{n0} \rangle, \dots$$

A_i is an original (or theoretical) transition matrix, lab_{01} is discrete evolution, for simplicity, $lab_{01} : v'_i = v_i$, and so on. \tilde{A}_i is an approximate (or actual) matrix with respect to A_i , so we have another trajectory,

$$\langle \ell_0, \tilde{v}_{00} \rangle \xrightarrow{\tilde{A}_0} \langle \ell_0, \tilde{v}_{01} \rangle \xrightarrow{\tilde{A}_0} \dots \langle \ell_0, \tilde{v}_{0m_0} \rangle \xrightarrow{lab_{01}} \langle \ell_1, \tilde{v}_{10} \rangle, \dots, \xrightarrow{\tilde{A}_n} \langle \ell_n, \tilde{v}_{n0} \rangle, \dots$$

If a given derivation ϵ is allowable, we aim to check whether two trajectories are identical; if every trajectory of two ILAHSs is identical, the two hybrid systems are identical with respect to derivation ϵ .

To measure the behavior of the hybrid system, this paper defines two types of metrics based on the Euclidean metric: the state metric d_{sta} and the trajectory metric d_{trj} . d_{sta} measures the distance of states, while d_{trj} specifies the similarity of two systems' behaviors.

Definition 8 (State Metric). State metric d_{sta} is Euclidean metric : $\mathbb{R}^n \times \mathbb{R}^n \rightarrow [0, 1]$ such that

$$d_{sta} \left(\langle l, \hat{v} \rangle, \langle l', \tilde{v} \rangle \right) = \begin{cases} 1 & \text{if } l \neq l', \\ d_e \left(\hat{v}, \tilde{v} \right) & \text{if } l = l', \end{cases}$$

$$\text{where } d_e \left(\hat{v}, \tilde{v} \right) = \frac{\sqrt{\sum_{i=1}^n (\hat{v}_i - \tilde{v}_i)^2}}{1 + \sqrt{\sum_{i=1}^n (\hat{v}_i - \tilde{v}_i)^2}}. \tag{5}$$

If $l = l'$, $d_{sta} \left(\langle l, \hat{v} \rangle, \langle l', \tilde{v} \rangle \right)$ can be abbreviated as $d_{sta} \left(\hat{v}, \tilde{v} \right)$.

Definition 9 (Trajectory Metric). Trajectory metric d_{trj} is defined as the minimum distance of relevant states, i.e.,

$$d_{trj}(\mathfrak{T}_1, \mathfrak{T}_2) = \min \left(d_{sta} \left(\langle l, \hat{v}_i \rangle, \langle l', \tilde{v}_i \rangle \right) \right). \tag{6}$$

Definition 10 (ϵ -Approximate States). Given a deviation ϵ , $\langle l, \hat{v} \rangle$ is ϵ -approximate to $\langle l', \tilde{v} \rangle$ iff $d_{sta} \left(\langle l, \hat{v} \rangle, \langle l', \tilde{v} \rangle \right) \leq \epsilon$.

Definition 11 (ϵ -Approximate Trajectories). Given a deviation ϵ and two trajectories \mathfrak{T}_1 and \mathfrak{T}_2 , \mathfrak{T}_1 is ϵ -approximate to \mathfrak{T}_2 iff $d_{trj}(\mathfrak{T}_1, \mathfrak{T}_2) \leq \epsilon$.

Definition 12 (ϵ -Approximate Completed Trace Equivalence of ILAHS). Given a deviation ϵ , if all completed traces of two ILAHSs are ϵ -approximate trajectories, then the two ILAHSs are ϵ -approximate completed trace equivalence.

In Figure 6, \mathcal{H}_1 is an original ILAHS, given a deviation ϵ , \mathcal{H}_1 and \mathcal{H}_3 are completed trace equivalence, in the semantics of completed trace equivalence, \mathcal{H}_3 is identical to \mathcal{H}_1 .

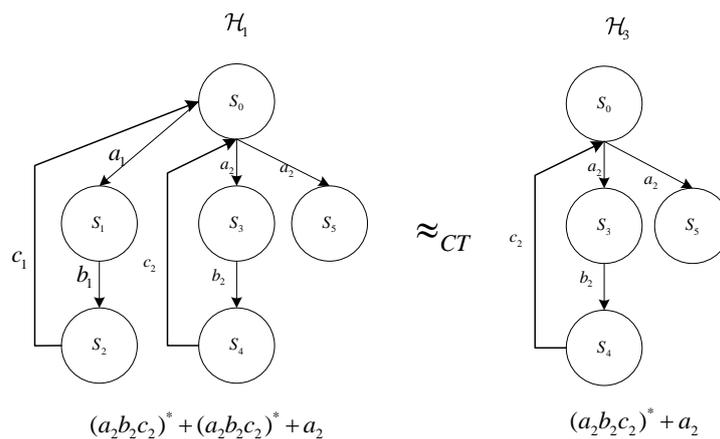


Figure 6. An approximate completed trace equivalence example.

Obviously, \mathcal{H}_3 involves fewer states than \mathcal{H}_1 ; in other words, by approximate completed trace equivalence, the research on hybrid systems can be simplified.

Since there exist two categories of trajectories in completed trace equivalence (Figure 6), the infinite trajectory $(a_2b_2c_2)^*$ and the finite trajectory a_2 , there are two corresponding decision conditions, the finite trajectory condition and the infinite trajectory condition. For simplicity, assume discrete evolution such as $lab_{01} : v'_i = v_i$.

Definition 13 (Finite Trajectory Condition). For a finite trajectory \mathfrak{T}_1

$$\langle \ell_0, \hat{v}_{00} \rangle \xrightarrow{A_0} \langle \ell_0, \hat{v}_{01} \rangle \xrightarrow{A_0} \dots \langle \ell_0, \hat{v}_{0m_0} \rangle \xrightarrow{lab_{01}} \langle \ell_1, \hat{v}_{10} \rangle, \dots, \xrightarrow{A_n} \langle \ell_n, \hat{v}_{nm_n} \rangle.$$

such that

$$\begin{cases} \hat{v}_{00} & \in \Theta \\ \vec{v} & = \hat{v}_{00} \\ \tilde{v} & = \hat{v}_{00} \\ \vec{v}' & = (A_0^{m_0} A_1^{m_1} \dots A_i^{m_i}) \vec{v} \\ \tilde{v}' & = (\tilde{A}_0^{m_0} \tilde{A}_1^{m_1} \dots \tilde{A}_i^{m_i}) \tilde{v} \\ d_{sta}(\vec{v}', \tilde{v}') & > \varepsilon \end{cases} \text{ have no solution for every } 0 \leq i \leq n. \tag{7}$$

The infinite trajectory condition is defined by the inductive method.

Definition 14 (Infinite Trajectory Condition). For an infinite trajectory \mathfrak{T}_2

$$\langle \ell_0, \hat{v}_{00} \rangle \xrightarrow{A_0} \langle \ell_0, \hat{v}_{01} \rangle \xrightarrow{A_0} \dots \langle \ell_0, \hat{v}_{0m_0} \rangle \xrightarrow{lab_{01}} \langle \ell_1, \hat{v}_{10} \rangle, \dots, \xrightarrow{A_n} \langle \ell_n, \hat{v}_{nm_n} \rangle \xrightarrow{lab_{n0}} \langle \ell_0, \hat{v}_{00'} \rangle \xrightarrow{A_0} \dots,$$

such that

$$\text{Initiation : } \begin{cases} \vec{v}, \tilde{v} \in \Theta \\ d_{sta}(\vec{v}, \tilde{v}) > \varepsilon \end{cases} \tag{8}$$

$$\text{Consecution : } \begin{cases} \vec{v}' & = (A_0^{m_0} A_1^{m_1} \dots A_i^{m_i}) \vec{v} \\ \tilde{v}' & = (\tilde{A}_0^{m_0} \tilde{A}_1^{m_1} \dots \tilde{A}_i^{m_i}) \tilde{v} \\ d_{sta}(\vec{v}', \tilde{v}') & > \varepsilon \end{cases} \tag{9}$$

for every $0 \leq i \leq n$, (8) and (9) have no solution.

The condition *Initiation* shows that the deviation of two states is initially less than ε ; the condition *Consecution* shows that deviation less than ε is preserved by the loop.

Complexity Analysis: Take (9) for example. Assuming $\vec{v} = \{v_1, v_2, \dots, v_n\}$, $deg(v_i) \leq 2, 1 \leq i \leq n$, our method consists of three main steps. In the first step, we transform the equations of (9) into triangular sets (i.e., equations in the triangular form) by Ritt–Wu’s method. By [35], the complexity of this step is $\mathcal{O}((2n)^{(2n)^{\mathcal{O}(1)}} (2+1)^{\mathcal{O}(2n)^3})$. The second step is to compute a border polynomial (BP) from the triangularized systems through resultant computation. By [35], the complexity of computing the BP is at most $(2n+2)\mathcal{O}((2n)^{6n+2n^2} (2+1)^{2n^3})$. Finally, we use the PCAD (partial cylindrical algebraic decomposition) algorithm with the BP to obtain the real solution classification; the complexity of this step is at most $\mathcal{O}(2D^{2^{4n+8}})$, where $D = \mathcal{O}((2n)^{4n^2+8n^3} (2+1)^{4n^3})$, the highest degree of BP.

A Special Case: The approaches proposed in papers [4,5,24] use the Frobenius norm to study approximate equivalence of real-time linear algebraic hybrid automaton. Their approaches can only apply to a class of special matrices. We now prove that their approaches are a special case of our approach.

We first introduce the conclusions in their papers [4,5,24].

For A_j and its approximate matrix $\tilde{A}_j = A + \delta_{A_j}$, which are $n \times n$ nonsingular matrices, the eigenvalues of matrix A are $\lambda_i (i = 1, \dots, n) \in \mathbb{R}$, and there exists an orthogonal matrix U :

$$\begin{aligned}
 A_k &= U^T \begin{bmatrix} \lambda_{k1} & & * \\ & \ddots & \\ & & \lambda_{kn} \end{bmatrix} U, \\
 A_k + \delta_{A_k} &= U^T \begin{bmatrix} \lambda_{k1} + \delta_{\lambda_{k1}} & & * \\ & \ddots & \\ & & \lambda_{kn} + \delta_{\lambda_{kn}} \end{bmatrix} U, \\
 A_k A_{k-1} \dots A_1 &= U^T \begin{bmatrix} \lambda_{11} \dots \lambda_{k1} & & * \\ & \ddots & \\ & & \lambda_{1n} \dots \lambda_{kn} \end{bmatrix} U, \\
 (A_k + \delta_{A_k})(A_{k-1} + \delta_{A_{k-1}}) \dots (A_1 + \delta_{A_1}) &= U^T \begin{bmatrix} (\lambda_{11} + \delta_{\lambda_{11}}) \dots (\lambda_{k1} + \delta_{\lambda_{k1}}) & & * \\ & \ddots & \\ & & (\lambda_{1n} + \delta_{\lambda_{1n}}) \dots (\lambda_{kn} + \delta_{\lambda_{kn}}) \end{bmatrix} U,
 \end{aligned}$$

$$\begin{aligned}
 \delta_{A_j} &= \tilde{A}_j - A_j, \|\delta_{A_j}\|_2 = r_j \|A_j\|_2, \\
 \delta_k &= (A_k + \delta_{A_k})(A_{k-1} + \delta_{A_{k-1}}) \dots (A_1 + \delta_{A_1}) - A_k A_{k-1} \dots A_1 \\
 &\leq ((1 + r_1) \dots (1 + r_k) - 1) \|A_1\|_2 \dots \|A_k\|_2 \\
 &\leq (e^{r_1 + r_2 + \dots + r_k} - 1) \|A_1\|_2 \|A_2\|_2 \dots \|A_k\|_2 = W_k.
 \end{aligned}$$

Then, for a deviation ω and $\frac{W_k}{\|A_k A_{k-1} \dots A_1\|_2} \leq \omega$, if $A_k A_{k-1} \dots A_1$ are ω -approximate to $(A_k + \delta_{A_k})(A_{k-1} + \delta_{A_{k-1}}) \dots (A_1 + \delta_{A_1})$, $\forall h \in N^+$ and $1 \leq h < k$, $A_h A_{h-1} \dots A_1$ are ω -approximate to $(A_h + \delta_{A_h})(A_{h-1} + \delta_{A_{h-1}}) \dots (A_1 + \delta_{A_1})$.

This conclusion is equivalent to (11) \models (13) for an existing constant ε .

Proof.

$$\begin{aligned}
 \because d_{sta}^{(k)}(\vec{v}', \vec{v}') &= \|(\tilde{A}_1 \tilde{A}_2 \dots \tilde{A}_k - A_1 A_2 \dots A_k) \hat{v}_0\|_2 \\
 &= \|\delta_k \hat{v}_0\|_2 \leq \|\delta_k\|_2 \|\hat{v}_0\|_2 \leq W_k \|\hat{v}_0\|_2 \\
 &\leq \omega \|A_k A_{k-1} \dots A_1\|_2 \|\hat{v}_0\|_2 = \varepsilon,
 \end{aligned} \tag{10}$$

$$\therefore \begin{cases} \hat{v}_0 & \in \Theta \\ \vec{v} & = \hat{v}_0 \\ \tilde{\vec{v}} & = \hat{v}_0 \\ \vec{v}' & = (A_1 A_2 \dots A_k) \vec{v} \\ \tilde{\vec{v}}' & = (\tilde{A}_1 \tilde{A}_2 \dots \tilde{A}_k) \tilde{\vec{v}} \\ d_{sta}^{(k)}(\vec{v}', \tilde{\vec{v}}') & > \varepsilon \end{cases} \text{ has no solution,} \tag{11}$$

$$\begin{aligned}
 \text{and } \because d_{sta}^{(h)}(\vec{v}', \tilde{\vec{v}}') &= \|(\tilde{A}_1 \tilde{A}_2 \dots \tilde{A}_h - A_1 A_2 \dots A_h) \hat{v}_0\|_2 \\
 &= \|\delta_h \hat{v}_0\|_2 \leq \|\delta_h\|_2 \|\hat{v}_0\|_2 \\
 &\leq ((1 + r_1)(1 + r_2) \dots (1 + r_h) - 1) \|A_1\|_2 \|A_2\|_2 \dots \|A_h\|_2 \|\hat{v}_0\|_2 \\
 &\leq ((1 + r_1)(1 + r_2) \dots (1 + r_h) \dots (1 + r_k) - 1) \|A_1\|_2 \|A_2\|_2 \dots \|A_k\|_2 \|\hat{v}_0\|_2 \\
 &\leq W_k \|\hat{v}_0\|_2 \leq \omega \|A_k A_{k-1} \dots A_1\|_2 \|\hat{v}_0\|_2 = \varepsilon,
 \end{aligned} \tag{12}$$

i.e., for every $1 \leq h < k$,

$$\left\{ \begin{array}{l} \hat{v}_0 \in \Theta \\ \vec{v} = \hat{v}_0 \\ \tilde{v} = \hat{v}_0 \\ \vec{v}' = (A_1 A_2 \dots A_h) \vec{v} \\ \tilde{v}' = (\tilde{A}_1 \tilde{A}_2 \dots \tilde{A}_h) \tilde{v} \\ d_{sta}^{(h)}(\vec{v}', \tilde{v}') > \varepsilon \end{array} \right. \text{ has no solution,} \tag{13}$$

in other words, (11) \models (13). \square

4. Experiments

In this section, we present an example to demonstrate the application of our method to simplify the ILAHS in approximate completed trace equivalence semantics. Figure 7 is a boiler feed-water control system, which has both continuous and discrete states and can be modeled as a hybrid system. For simplicity, we assume that there exist nine discrete locations that are divided according to the combination of fuel supply (low, high) and outdoor temperature (low, medium, high). rs_0 is the initial location of the system, which does not involve continuous evolution and directly jumps to a different location by τ_i . The continuous variables $X = (x_1, x_2)^T$; x_1 and x_2 stand for water level and water supply, respectively. The fuel supply and outdoor temperature have an effect on x_1 and x_2 , and the regulation rules of x_1 and x_2 conform to their respective differential equations at different locations $rs'_i (i = 1, \dots, 8)$. The initial state of the system $X_0 = (100, 10)^T$, and for τ_i , the discrete transition program $\alpha_i : x'_1 = x_1 \wedge x'_2 = x_2$, labeled by *lab*. Assuming the duration time at every location $rs'_i (i = 1, \dots, 8)$ is 10 s and the discrete time of continuous evolution is 5 s, Figure 8 is the corresponding ILAHS.

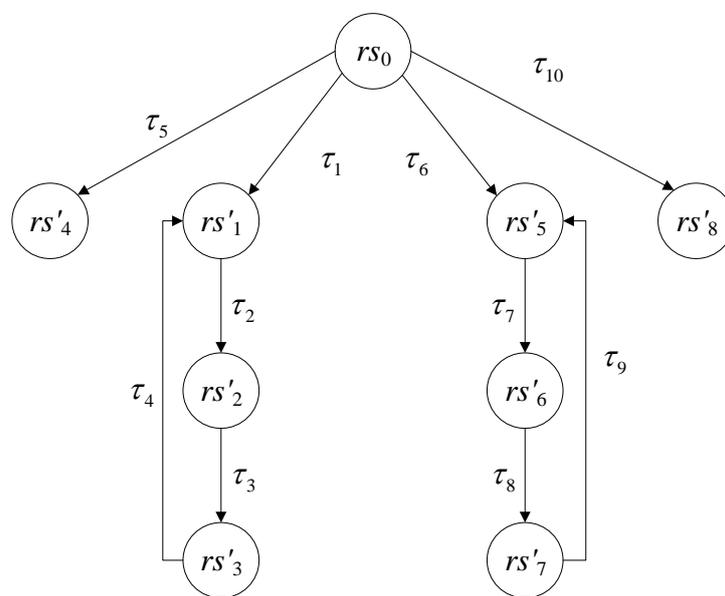


Figure 7. Automaton of feed-water control system.

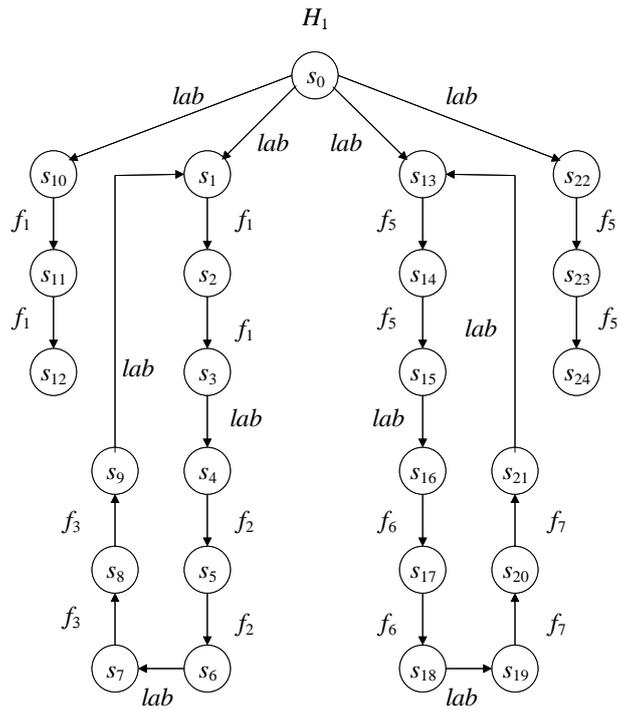


Figure 8. ILAHS of feed-water control system.

$$\begin{aligned}
 f_1, f_4 : X' &= \begin{bmatrix} \frac{111}{100} & \frac{2}{5} \\ 0 & \frac{151}{100} \end{bmatrix} X & f_8, f_5 : X' &= \begin{bmatrix} \frac{109}{100} & \frac{2}{5} \\ 0 & \frac{149}{100} \end{bmatrix} X \\
 f_2 : X' &= \begin{bmatrix} \frac{111}{100} & \frac{11}{111} \\ 0 & 1 \end{bmatrix} X & f_3 : X' &= \begin{bmatrix} 1 & -\frac{51}{151} \\ 0 & \frac{100}{151} \end{bmatrix} X \\
 f_6 : X' &= \begin{bmatrix} \frac{100}{109} & \frac{9}{109} \\ 0 & 1 \end{bmatrix} X & f_7 : X' &= \begin{bmatrix} 1 & -\frac{49}{149} \\ 0 & \frac{100}{149} \end{bmatrix} X
 \end{aligned}$$

The original invariant set is

- $Inv(rs'_1) : \{(100 \leq x_1 \leq 133.69) \wedge (10 \leq x_2 \leq 22.801)\},$
- $Inv(rs'_2) : \{(112.801 \leq x_1 \leq 133.69) \wedge (x_2 = 22.801)\},$
- $Inv(rs'_3) : \{(100 \leq x_1 \leq 112.801) \wedge (10 \leq x_2 \leq 22.801)\},$
- $Inv(rs'_4) : \{(100 \leq x_1 \leq 133.69) \wedge (10 \leq x_2 \leq 22.801)\},$
- $Inv(rs'_5) : \{(100 \leq x_1 \leq 129.13) \wedge (10 \leq x_2 \leq 22.201)\},$
- $Inv(rs'_6) : \{(112.801 \leq x_1 \leq 129.13) \wedge (x_2 = 22.201)\},$
- $Inv(rs'_7) : \{(100 \leq x_1 \leq 112.201) \wedge (10 \leq x_2 \leq 22.201)\},$
- $Inv(rs'_8) : \{(100 \leq x_1 \leq 129.13) \wedge (10 \leq x_2 \leq 22.201)\}.$

Assume Figure 9 is the approximate ILAHS, where

$$\begin{aligned}
 \tilde{f}_1, \tilde{f}_4, \tilde{f}_5, \tilde{f}_8 : \tilde{X}' &= \begin{bmatrix} \frac{11}{10} & \frac{2}{5} \\ 0 & \frac{5}{3} \end{bmatrix} X, \\
 \tilde{f}_2, \tilde{f}_6 : \tilde{X}' &= \begin{bmatrix} \frac{10}{11} & \frac{1}{11} \\ 0 & 1 \end{bmatrix} X, \\
 \tilde{f}_3, \tilde{f}_7 : \tilde{X}' &= \begin{bmatrix} 1 & -\frac{1}{3} \\ 0 & \frac{2}{3} \end{bmatrix} X.
 \end{aligned}$$

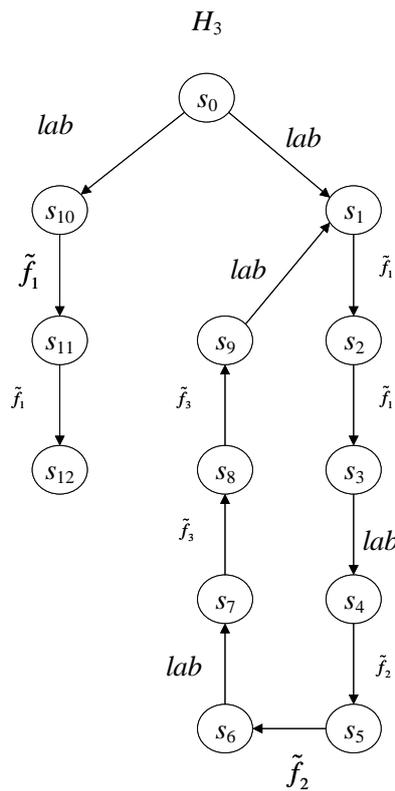


Figure 9. Approximate ILAHS.

The approximate invariant set is

$$\begin{aligned} \widetilde{Inv}(rs'_1), \widetilde{Inv}(rs'_4) &: \{(100 \leq x_1 \leq 131.4) \wedge (10 \leq x_2 \leq 22.5)\}, \\ \widetilde{Inv}(rs'_5), \widetilde{Inv}(rs'_8) &: \{(100 \leq x_1 \leq 131.4) \wedge (10 \leq x_2 \leq 22.5)\}, \\ \widetilde{Inv}(rs'_2), \widetilde{Inv}(rs'_6) &: \{(112.5 \leq x_1 \leq 131.4) \wedge (x_2 = 22.5)\}, \\ \widetilde{Inv}(rs'_3), \widetilde{Inv}(rs'_7) &: \{(100 \leq x_1 \leq 112.5) \wedge (10 \leq x_2 \leq 22.5)\}. \end{aligned}$$

The approximate discrete transition program is $\tilde{a}_i : x'_1 = x_1 \wedge x'_2 = x_2$, labeled by *lab*.

Given an error derivation $\epsilon = 0.1$, \mathcal{H}_1 is ϵ -approximate completed trace equivalent to \mathcal{H}_3 (Figure 9), i.e., in the semantics of completed trace equivalence, \mathcal{H}_1 is identical to \mathcal{H}_3 , which involves fewer states. Therefore, the original hybrid automaton (Figure 7) can be reduced to an equivalent but simpler hybrid automaton (Figure 10), and the design of the hybrid system is optimized with the theory of approximate completed trace equivalence.

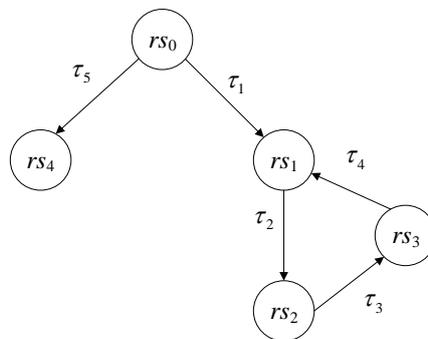


Figure 10. Reduced automaton.

5. Conclusions

With some restrictions, a general hybrid system can be specialized into an inhomogeneous linear algebraic hybrid system (ILAHS), which is ubiquitous in reality. Since errors are inevitable in engineering technology, it is meaningful to introduce deviation to the analysis of hybrid systems. Given an error derivation, this paper first proposes the theory of approximate completed trace equivalence. Then we come up with a new approach (based on SAS solving) to approximate completed trace equivalence, and the computation complexity is also analyzed. Finally, a case study is presented to illustrate the practicality of our method. Compared to the well-established approaches in this field, our method applies to more general ILAHS. Moreover, this technology is promising for solving parametric approximate completed trace equivalence. We will address this issue in future work.

Author Contributions: Conceptualization, investigation, supervision, and writing, H.H.; review and editing, J.X.; review, J.W.

Funding: This work was partly supported by the National Natural Science Foundation of China under Grant No. 61772006, the Science and Technology Program of Guangxi under Grant No. AB17129012, the Science and Technology Major Project of Guangxi under Grant No. AA17204096, the Special Fund for Scientific and Technological Bases and Talents of Guangxi under Grant No. AD16380076, and the Special Fund for Bagui Scholars of Guangxi.

Conflicts of Interest: We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

1. Henzinger, T.A. The theory of hybrid automata. In *Verification of Digital and Hybrid Systems*; Springer: Berlin, Germany, 2000; pp. 265–292.
2. van Glabbeek, R.J. The linear time-branching time spectrum. In Proceedings of the 1st International Conference on Concurrency Theory (CONCUR '90), Amsterdam, The Netherlands, 27–30 August 1990; pp. 278–297.
3. van Glabbeek, R.J. The linear time-branching time spectrum II. In Proceedings of the 4th International Conference on Concurrency Theory (CONCUR'93), Hildesheim, Germany, 23–26 August 1993; pp. 66–81.
4. Yang, H.; Wu, J.; Zhang, Z.; Liu, Y. Approximate completed trace equivalence of three dimensional t-model nonlinear algebraic hybrid systems. *Appl. Math. Inf. Sci.* **2013**, *7*, 1693–1697, [[CrossRef](#)]
5. Yang, H.; Wu, J.; Zhang, Z. Approximate completed trace equivalence of homogeneous linear algebraic hybrid systems. *J. Conver. Inf. Technol.* **2013**, *8*, 520–527.
6. Yang, L.; Hou, X.; Xia, B. A complete discrimination system for polynomials. *Sci. China Ser. E Technol. Sci.* **1996**, *39*, 628–646.
7. Chen, Y.; Xia, B.; Yang, L.; Zhan, N. Generating polynomial invariants with DISCOVERER and QEPCAD. In *Formal Methods and Hybrid Real-Time Systems*; Springer: Berlin, Germany, 2007; pp. 67–82.
8. Xia, B.; Yang, L. An algorithm for isolating the real solutions of semi-algebraic systems. *J. Symb. Comput.* **2002**, *34*, 461–477. [[CrossRef](#)]
9. Xia, B.; Zhang, T. Real solution isolation using interval arithmetic. *Comput. Math. Appl.* **2006**, *52*, 853–860. [[CrossRef](#)]
10. Yang, L. Recent advances on determining the number of real roots of parametric polynomials. *J. Symb. Comput.* **1999**, *28*, 225–242. [[CrossRef](#)]
11. Yang, L.; Hou, X.; Xia, B. A complete algorithm for automated discovering of a class of inequality-type theorems. *Sci. China Ser. F Inf. Sci.* **2001**, *44*, 33–49. [[CrossRef](#)]
12. Girard, A.; Zheng, G. Verification of safety and liveness properties of metric transition systems. *ACM Trans. Embed. Comput. Syst. (TECS)* **2012**, *11*, 1–23. [[CrossRef](#)]
13. Girard, A.; Pappas, G.J. Approximation metrics for discrete and continuous systems. *IEEE Trans. Autom. Control.* **2007**, *52*, 782–798. [[CrossRef](#)]
14. Alfaro, L.D.; Faella, M.; Stoelinga, M. Linear and branching system metrics. *IEEE Trans. Softw. Eng.* **2008**, *35*, 258–273. [[CrossRef](#)]
15. Cao, Y.; Sun, S.X.; Wang, H.; Chen, G. A behavioral distance for fuzzy-transition systems. *IEEE Trans. Fuzzy Syst.* **2013**, *21*, 735–747. [[CrossRef](#)]

16. Fan, C.; Kapinski, J.; Jin, X.; Mitra, S. Locally optimal reach set over-approximation for nonlinear systems. In Proceedings of the 2016 International Conference Embedded Software, Pittsburgh, PA, USA, 2–7 October 2016; pp. 1–10.
17. Kendra, L.; Meeko, O. Finite state approximation for verification of partially observable stochastic hybrid systems. In Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control, Seattle, WA, USA, 14–16 April 2015; pp. 159–168.
18. Yang, Y.; Wang, Y.; Wu, Q.M.J.; Lin, X.; Liu, M. Progressive learning machine: A new approach for general hybrid system approximation. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 1855–1874. [[CrossRef](#)] [[PubMed](#)]
19. Fan, C.; Qi, B.; Mitra, S.; Viswanathan, M.; Duggirala, P.S. Automatic reachability analysis for nonlinear hybrid models with C2E2. In Proceedings of the 28th International Conference Comput. Aided Verification, Toronto, ON, Canada, 17–23 July 2016; pp. 531–538.
20. Xiang, W.; Tran, H.D.; Johnson, T.T. Output reachable set estimation for switched linear systems and its application in safety verification. *IEEE Trans. Autom. Control* **2017**, *62*, 5380–5387. [[CrossRef](#)]
21. Bemporad, A.; Torrisi, F.D.; Morari, M. Optimization-based verification and stability characterization of piecewise affine and hybrid systems. In Proceedings of the 3rd International Conference on Hybrid Systems: Computation and Control (HSCC 2000), Pittsburgh, PA, USA, 23–25 March 2000; pp. 45–58.
22. Girard, A. Reachability of uncertain linear systems using zonotopes. In Proceedings of the 8th International Conference on Hybrid Systems: Computation and Control (HSCC 2005), Zurich, Switzerland, 9–11 March 2005; pp. 291–305.
23. Kurzthanski, A.B.; Varaiya, P. Ellipsoidal techniques for reachability analysis: Internal approximation. *Syst. Control Lett.* **2000**, *41*, 201–211. [[CrossRef](#)]
24. Yang, H. Over approximate flow pipeline volume algorithm for three variables real-time linear algebraic hybrid automaton. In Proceedings of the 7th International Conference on Management, Education, Information and Control (MEICI 2017), Shenyang, China, 15–17 September 2017. Available online: <https://download.atlantis-press.com/article/25885913.pdf> (accessed on 25 October 2019).
25. Alur, R.; Courcoubetis, C.; Halbwachs, N.; Henzinger, T.A.; Ho, P.-H. *Hybrid Automata: An Algorithmic Approach to the Specification and Verification of Hybrid Systems*; Springer: Berlin, Germany, 1993; pp. 209–229.
26. Alur, R.; Courcoubetis, C.; Halbwachs, N.; Henzinger, T.A.; Ho, P.-H.; Nicollin, X.; Olivero, A.; Sifakis, J.; Yovine, S. The algorithmic analysis of hybrid systems. *Theor. Comput. Sci.* **1995**, *138*, 3–34. [[CrossRef](#)]
27. Maler, O.; Manna, Z.; Pnueli, A. From timed to hybrid systems. In Proceedings of the REX Workshop 1991, Mook, The Netherlands, 3–7 June 1991; pp. 447–484.
28. Alur, R.; Henzinger, T.A. Modularity for timed and hybrid systems. In Proceedings of the 8th International Conference on Concurrency Theory (CONCUR '97), Warsaw, Poland, 1–4 July 1997; pp. 74–88.
29. Lygeros, J.; Tomlin, C.; Sastry, S. Controllers for reachability specifications for hybrid systems. *Automatica* **1999**, *35*, 349–370. [[CrossRef](#)]
30. Sankaranarayanan, S.; Sipma, H.B.; Manna, Z. Constructing invariants for hybrid systems. *Form. Methods Syst. Des.* **2008**, *32*, 25–55. [[CrossRef](#)]
31. Xia, B. Discoverer: A tool for solving semi-algebraic systems. *ACM Commun. Comput. Algebra* **2007**, *41*, 102–103. [[CrossRef](#)]
32. Chen, C.; Davenport, J.H.; Lemaire, F.; Maza, M.M.; Xia, B.; Xiao, R.; Xie, Y. Computing the real solutions of polynomial systems with the regular chains library in maple. *ACM Commun. Comput. Algebra* **2011**, *45*, 166–168. [[CrossRef](#)]
33. Chen, C.; Davenport, J.H.; May, J.P.; Maza, M.M.; Xia, B.; Xiao, R. Triangular decomposition of semi-algebraic systems. *J. Symb. Comput.* **2013**, *49*, 3–26. [[CrossRef](#)]
34. Chen, C.; Davenport, J.H.; Maza, M.M.; Xia, B.; Xiao, R. Computing with semi-algebraic sets: relaxation techniques and effective boundaries. *J. Symb. Comput.* **2013**, *52*, 72–96. [[CrossRef](#)]
35. Gallo, G.; Mishra, B. Efficient Algorithms and Bounds for Wu-Ritt Characteristic Sets. In Proceedings of the MEGA'90, Livorno, Italy, 17–21 April 1990; pp. 119–142.

