Approximate completed trace equivalence of real-time linear algebraic Hybrid Automata

Zhi Liu*

Sichuan College of Architectural Technology, China

Received 1 August 2014, www.cmnt.lv

Abstract

In allusion to design simpler software system, the paper proposes approximate completed trace equivalence of real-time linear algebraic Hybrid Automata. Firstly, it pulls real-time linear algebraic program into Hybrid Automaton and establishes real-time linear algebraic Hybrid Automaton. Next, it uses matrix Frobenius norm to analyse approximation of real-time linear algebraic Hybrid Automata. Afterwards, it gets approximate completed trace equivalence of real-time linear algebraic Hybrid Automata. Finally, the Email virus spreading automata example shows that approximate completed trace equivalence of real-time algebraic Hybrid Automata can simplify automaton.

Keywords: Hybrid Automata, approximate, completed trace equivalence, algebraic program

1 Introduction

Hybrid system [1-4] has continuous variables processes and discrete event processes. They exchange information with each other. Most existing control system can be seen as Hybrid Systems. For example, in the level, control system water level is a continuous quantity and valve switch is discrete quantity. In the room temperature control system, room temperature is a continuous quantity and air conditioning switch is discrete quantity. Completed trace equivalence [5] can effectively reduce states of automaton. Approximate completed trace equivalence theory analysis the approximation of automata and gets completed trace equivalence automaton of approximate automaton. For real-time linear algebraic Hybrid Automata, there is no corresponding approximate completed trace equivalence theory.

This paper is organized as follows. In section 2, it establishes real-time linear algebraic Hybrid Automaton which uses real-time linear algebraic programs to describe actions. In section 3, it uses Frobenius norm to analyse approximation of real-time linear algebraic Hybrid Automata. In section 4, it gets approximate completed trace equivalence of real-time linear algebraic Hybrid Automata. In section 5, the Email virus spreading automata example shows that approximate completed trace equivalence of real-time linear algebraic Hybrid Automata can optimize real-time linear algebraic Hybrid Automaton.

2 Real-time linear algebraic Hybrid Automaton

Definition 1 (Real-Time Linear Algebraic Program): let R be the set of real numbers, $x_i(i=1,\cdots,n)$ and $x_i'(i=1,\cdots,n)$ be the real variables, $t\in \mathrm{R}$ be the time variable. A real-time linear algebraic program is an algebraic program likes $X'=X+(AX+b)t$, where $X'=(x_1',\cdots,x_n')^{\mathrm{T}}$ is the post-state value of real-time linear algebraic program transition, $X=(x_1,\cdots,x_n)^{\mathrm{T}}$ is the pre-state value of real-time linear algebraic program transition.

$$A=\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \text{ and } A\neq 0,\ b=(b_1,b_2,\cdots,b_n)^{\mathrm{T}},$$

is n dimensional column vector.

In the real-time linear algebraic program $X'=X+(AX+b)t$, when $X=(x_1,\cdots,x_n)^{\mathrm{T}}$ is given, $AX+b$ is a n dimensional column vector. $X'=(x_1',\cdots,x_n')^{\mathrm{T}}$ changed monotonically along with the time *t*.

Definition 2 (Real-Time Linear Algebraic Hybrid Automaton): a real-time linear algebraic Hybrid Automaton is a tuple *H*=<*Q*, *V*, *HX*, *Init*, *Lab*, *E*, *Inv*, *F*, *R* >, where:

1) *Q* is a set of system discrete locations.
2) *V* is a set of system continuous variables.
3) *HX* is a set of system continuous variables values.
4) $Init\in Q\times HX$ is a set of system initial states. $Init=\{<q_0,X_0>\}$. It has only one initial state in the real-time linear algebraic Hybrid Automaton.

***Corresponding author's** e-mail: 1157713577@qq.com

5) *Lab* is a set of discrete transition programs. Discrete transition programs are real number assignment programs $X' = C$.

6) *E* is a set of discrete transitions.

7) *Inv* is a set of continuous variables invariants.

8) *F* is a set of real-time linear algebraic programs $X' = X + (AX + b)t$ which describe system continuous variables dynamic processes.

9) *R* is a set of discrete location transition conditions.

In the each discrete location of real-time linear algebraic Hybrid Automata, every element of X changed monotonically along with the time *t*. Because each discrete transition program is a real number assignment program, continuous variables value after discrete transition is independent with continuous variables value before discrete transition.

Example 1 (Cattle and sheep breeding law automaton): let x_1, x_2 respectively be cattle number and sheep number, vector $X = (x_1, x_2)^{\mathrm{T}}$ be total number.

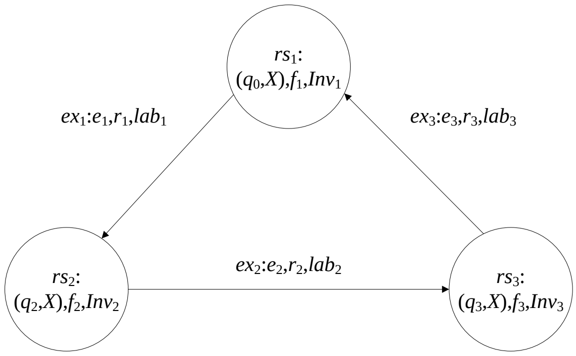

FIGURE 1 Cattle and sheep breeding law automaton

In Figure 1, the system initial state is $< q_0, X_0 >$, where $X_0 = (50,50)^{\mathrm{T}}$. In the discrete locations q_0, q_2, q_3, time $t \in [0,3]$. In the discrete location q_0 the fodder is short, cattle and sheep breeding capacity is weak, real-time linear algebraic program f_1 is $\begin{cases} x_1' = x_1 + (0.2x_1 + 0.1x_2)t \\ x_2' = x_2 + (0.2x_2 + 2)t \end{cases}$. In the discrete location q_2 the fodder is moderate, cattle and sheep breeding capacity is moderate, real-time linear algebraic program f_2 is $\begin{cases} x_1' = x_1 + (0.5x_1 + 0.2x_2 + 1)t \\ x_2' = x_2 + (0.6x_2 + 3)t \end{cases}$. In the discrete location q_3 the fodder is abundant, cattle and sheep breeding capacity is strong, real-time linear algebraic program f_3 is $\begin{cases} x_1' = x_1 + (x_1 + 0.3x_2 + 2)t \\ x_2' = x_2 + (x_2 + 4)t \end{cases}$. Discrete transition programs $lab_i (i = 1,2,3)$ are $\begin{cases} x_1' = 50 \\ x_2' = 50 \end{cases}$.

3 Approximation of real-time linear algebraic Hybrid Automata

In the real life, matrix *A* and vector *b* are the result of tool measure. They often have some inevitable errors. Let *A* be the approximate matrix, *b* be the approximate vector, $A + \delta A$ be the actual matrix, $b + \delta b$ be the actual vector, $X' = X + [(A + \delta A)X + (b + \delta b)]t$ be the actual real-time linear algebraic program, $X' = X + (AX + b)t$ be the approximate real-time linear algebraic program. We use Frobenius norm to study approximation of real-time linear algebraic Hybrid Automata.

Matrix $A \in \mathrm{R}^{m \times n}$, the Frobenius norm of matrix *A* is $\|A\|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{\frac{1}{2}}$.

For the actual real-time linear algebraic program $X' = X + [(A + \delta A)X + (b + \delta b)]t$ and approximate real-time linear algebraic program $X' = X + (AX + b)t$, they can be respectively written as $X' = X + \begin{pmatrix} A + \delta A & b + \delta b \end{pmatrix} \begin{pmatrix} X \\ 1 \end{pmatrix} t$ and $X' = X + \begin{pmatrix} A & b \end{pmatrix} \begin{pmatrix} X \\ 1 \end{pmatrix} t$. Approximation of actual real-time linear algebraic program and approximate real-time linear algebraic program can be converted to approximation of matrix $\begin{pmatrix} A + \delta A & b + \delta b \end{pmatrix}$ and matrix $\begin{pmatrix} A & b \end{pmatrix}$.

Let $B + \delta B = \begin{pmatrix} A + \delta A & b + \delta b \end{pmatrix}$, $B = \begin{pmatrix} A & b \end{pmatrix}$, $\delta B = \begin{pmatrix} \delta A & \delta b \end{pmatrix}$. We use matrix Frobenius norm to analysis approximation of matrix $B + \delta B$ and matrix *B*.

$$\frac{\|(B + \delta B) - B\|_F}{\|B\|_F} = \frac{\|\delta B\|_F}{\|B\|_F} = \frac{\left(\sum_{i=1}^{m} \sum_{j=1}^{n} |\delta b_{ij}|^2 \right)^{\frac{1}{2}}}{\left(\sum_{i=1}^{m} \sum_{j=1}^{n} |b_{ij}|^2 \right)^{\frac{1}{2}}} = W.$$

For a given positive number ε, if it has $W < \varepsilon$, then matrix $\begin{pmatrix} A + \delta A & b + \delta b \end{pmatrix}$ and matrix $\begin{pmatrix} A & b \end{pmatrix}$ are approximate about ε. For a given positive number ε, if it has $W \geq \varepsilon$, then matrix $\begin{pmatrix} A + \delta A & b + \delta b \end{pmatrix}$ and matrix $\begin{pmatrix} A & b \end{pmatrix}$ are not approximate about ε.

Definition 3: for a given positive number ε, if matrix $\begin{pmatrix} A + \delta A & b + \delta b \end{pmatrix}$ and matrix $\begin{pmatrix} A & b \end{pmatrix}$ are approximate about ε, then actual real-time linear algebraic program $X' = X + [(A + \delta A)X + (b + \delta b)]t$ and approximate real-time linear algebraic program $X' = X + (AX + b)t$ are approximate. For a given positive number ε, if matrix

$\begin{pmatrix}A+\delta A & b+\delta b\end{pmatrix}$ and matrix $\begin{pmatrix}A & b\end{pmatrix}$ are not approximate about ε, then actual real-time linear algebraic program $X' = X + [(A+\delta A)X + (b+\delta b)]t$ and approximate real-time linear algebraic program $X' = X + (AX + b)t$ are not approximate.

Definition 4: if all corresponding real-time linear algebraic programs in two corresponding traces of two real-time linear algebraic Hybrid Automata are approximate, then two traces are approximate.

Definition 5: if all corresponding traces of two real-time linear algebraic Hybrid Automata are approximate, then two real-time linear algebraic Hybrid Automata are approximate.

When real-time linear algebraic programs of real-time linear algebraic Hybrid Automata are approximate, continuous variables values are also accordingly changed. The continuous variables invariants and discrete transition condition in each discrete location also can occurrence change. Approximation of real-time linear algebraic Hybrid Automata can optimize real-time linear algebraic programs and simplify real-time linear algebraic Hybrid Automaton. It also can increase mathematical computation speed. It gets the approximate automaton of actual real-time linear algebraic Hybrid Automaton by approximation of real-time linear algebraic Hybrid Automata.

4 Approximate completed trace equivalence of real-time linear algebraic Hybrid Automata

It is known that an actual real-time linear algebraic Hybrid Automaton H_1. We can get approximate real-time linear algebraic Hybrid Automaton H_2 of actual real-time linear algebraic Hybrid Automaton H_1 by approximation of real-time linear algebraic Hybrid Automata. And then, we get completed trace equivalence Automaton H_3 of approximate real-time linear algebraic Hybrid Automaton H_2 by completed trace equivalence theory. H_1 and H_3 are approximate completed trace equivalence. It uses H_3 to replace H_1.

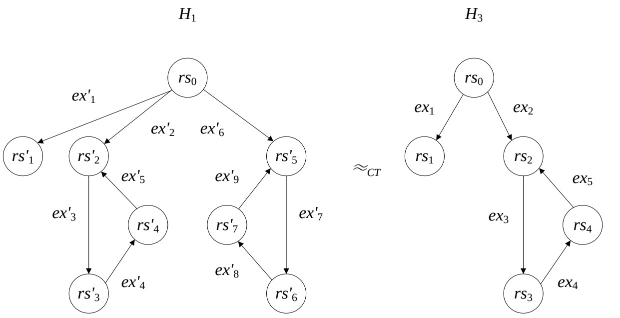

FIGURE 2 An approximate completed trace equivalence of real-time linear algebraic Hybrid Automata example

In Figure 2, we can use approximate method and completed trace equivalence theory to get approximate completed trace equivalence of real-time linear algebraic Hybrid Automata. Approximate completed trace equivalence of real-time linear algebraic Hybrid Automata can optimize real-time linear algebraic programs and reduce states of real-time linear algebraic Hybrid Automaton.

5 Experiments

In the Email of computer Internet, some Emails often have viruses. They infect other computers and increases virus number. Let Email group have four computer, where three computers send Email with each other, one computer sends Email to other three computer and it can not receive Email. We can only consider the influence of communication frequency high, medium, low three cases and open Email number. Three computers Email virus number respectively be x_1, x_2, x_3. It uses $X = (x_1, x_2, x_3)^{\mathrm{T}}$ to express three computers Email virus total number. The other one computer Email virus number is always invariable. It sets 15 days as a cycle, previous 5 days communication frequency is low, middle 5 days communication frequency is medium, latter 5 days communication frequency is high. In the end of each 5 days disinfect three computers and the leaving virus number is the initial virus number. In the Email virus spreading automaton open Email number respectively are four times and five times.

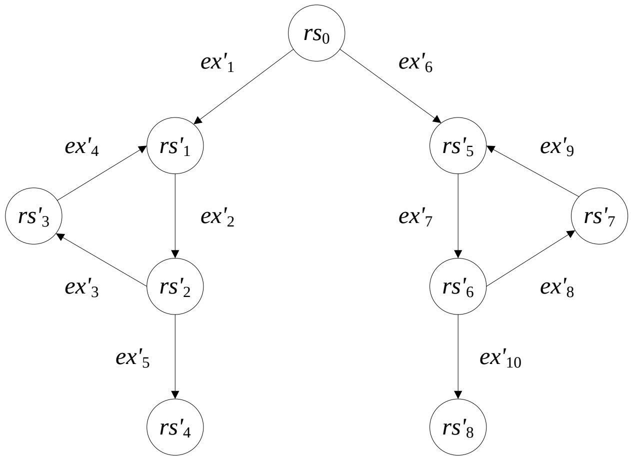

FIGURE 3 Actual Email virus spreading automaton

In Figure 3, left branch is open Email four times and right branch is open Email five times. Initial Email virus number of Email virus spreading automaton is $X_0 = (2,2,2)^{\mathrm{T}}$.

The actual real-time linear algebraic program, which describe system variables dynamic process

$$f'_1 \text{ is } X' = X + \left[\begin{pmatrix}1.95 & 0.97 & 0.98\\ 0.97 & 1.96 & 0.98\\ 0.99 & 0.97 & 1.98\end{pmatrix} X + \begin{pmatrix}0.97\\ 0.97\\ 0.97\end{pmatrix}\right] t,$$

$$f'_2 \text{ is } X' = X + \left[\begin{pmatrix}2.96 & 1.97 & 1.96\\ 1.99 & 2.98 & 1.98\\ 1.94 & 1.99 & 2.97\end{pmatrix} X + \begin{pmatrix}1.98\\ 1.98\\ 1.98\end{pmatrix}\right] t,$$

f'_3 and f'_4 are

$$X' = X + \left[\begin{pmatrix} 3.94 & 2.97 & 2.99 \\ 2.98 & 3.96 & 2.96 \\ 2.97 & 2.98 & 3.97 \end{pmatrix} X + \begin{pmatrix} 2.98 \\ 2.98 \\ 2.98 \end{pmatrix}\right] t ,$$

f'_5 is $X' = X + \left[\begin{pmatrix} 2.02 & 1.02 & 1.02 \\ 1.03 & 2.01 & 1.02 \\ 1.04 & 1.01 & 2.03 \end{pmatrix} X + \begin{pmatrix} 1.01 \\ 1.01 \\ 1.01 \end{pmatrix}\right] t ,$

f'_6 is $X' = X + \left[\begin{pmatrix} 3.02 & 2.01 & 2.02 \\ 2.03 & 3.05 & 2.02 \\ 2.04 & 2.01 & 3.04 \end{pmatrix} X + \begin{pmatrix} 2.03 \\ 2.03 \\ 2.03 \end{pmatrix}\right] t ,$

f'_7 and f'_8 are

$$X' = X + \left[\begin{pmatrix} 4.04 & 3.02 & 3.06 \\ 3.02 & 4.01 & 3.03 \\ 3.03 & 3.03 & 4.02 \end{pmatrix} X + \begin{pmatrix} 3.03 \\ 3.03 \\ 3.03 \end{pmatrix}\right] t .$$

Discrete transition programs $lab'_i (i = 1, 2, \cdots, 10)$ is $X' = (2, 2, 2)^{\mathrm{T}}$.

The approximate real-time linear algebraic program which describe system variables dynamic process

f_1 and f_5 are $X' = X + \left[\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} X + \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}\right] t ,$

f_2 and f_6 are $X' = X + \left[\begin{pmatrix} 3 & 2 & 2 \\ 2 & 3 & 2 \\ 2 & 2 & 3 \end{pmatrix} X + \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}\right] t ,$

f_3, f_4, f_7 and f_8 are $X' = X + \left[\begin{pmatrix} 4 & 3 & 3 \\ 3 & 4 & 3 \\ 3 & 3 & 4 \end{pmatrix} X + \begin{pmatrix} 3 \\ 3 \\ 3 \end{pmatrix}\right] t .$

Discrete transition programs $lab_i (i = 1, 2, \cdots, 10)$ is $X' = (2, 2, 2)^{\mathrm{T}}$.

For the given allowable error $\varepsilon = 0.03$, we get approximate matrix Frobenius norm max relative error $W_{\max} \approx 0.02267 < \varepsilon$. Because corresponding traces of H_1 and H_2 are approximate, H_1 and H_2 are approximate. In the error allowable range, it can use H_2 to replace H_1. Approximation of Email virus spreading automata can optimize real-time linear algebraic programs.

It gets the approximate completed trace equivalence automaton of actual Email virus spreading automaton by approximate completed trace equivalence model of actual Email virus spreading automaton state transition model (Figure 4).

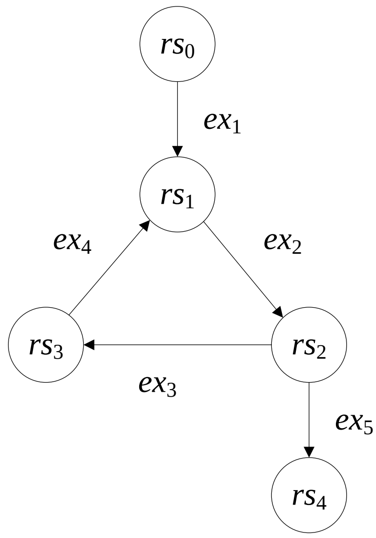

FIGURE 4 Approximate completed trace equivalence automaton of actual Email virus spreading automaton

In the error allowable range, approximate completed trace equivalence automaton of actual Email virus spreading automaton can replace actual Email virus spreading automaton to study virus number change of Email virus spreading model. Approximate completed trace equivalence of real-time linear algebraic Hybrid Automata can optimize real-time linear algebraic programs of actual Email virus spreading automaton and eliminate states of actual Email virus spreading automaton.

6 Conclusion

In this paper, approximate completed trace equivalence of real-time linear algebraic Hybrid Automata is proposed. It can simplify real-time linear algebraic Hybrid Automaton. In the future work, we will use approximate completed trace equivalence of real-time linear algebraic Hybrid Automata to develop software.

Acknowledgments

This work was supported in part by the Guangxi Key Laboratory of Hybrid Computational and IC Design Analysis Open Fund HCIC201303.

References

[1] Mansour C, Clodic D 2012 Dynamic modeling of the electro-mechanical configuration of the Toyota Hybrid System series/parallel power train *International Journal of Automotive Technology* **13**(1) 143-66

[2] Suzuki T 2010 Advanced motion as a hybrid system *Electronics and Communications in Japan* **93**(12) 35-43

[3] Allibert B 1999 Analytic controllability of a linear hybrid system *SIAM Journal on Control and Optimization* **37**(3) 844-68

[4] Lauer F, Bloch G, Vidal R 2011 A continuous optimization framework for hybrid system identification *Automatica* **47**(3) 608-13

[5] van Glabbeek R J 1990 The Linear Time – Branching Time Spectrum *CONCUR '90 Theories of Concurrency: Unification and Extension Lectures Notes in Computer Science* Springer-Verlag **458** 278-97
[6] Bauer S S, Fahrenberg U, Juhl L, Larsen K G, Kegay A, Thrane C 2013 Weighted modal transition systems *Formal Methods in System Design* **42**(2) 193-220
[7] Shen M 2013 H∞ filtering of continuous Markov jump linear system with partly known Markov modes and transition probabilities *Journal of the Franklin Institute* **350**(10) 3384-99
[8] Fu J, Tanner H, Heinz J and Chandlee J 2014 *IEEE Transactions on Automatic Control* **59**(2) 505-11
[9] Bernardo M, Nicola N D and Loreti M 2014 Revisiting trace and testing equivalences for nondeterministic and probabilistic processes *Logical Methods in Computer Science* **10**(1:16) 1-42
[10] Cheval V, Cortier V and Delaune S 2013 Deciding equivalence-based properties using constraint solving *Theoretical Computer Science* **492** 1-39
[11] André P 2010 Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics *Springer*

Authors

Zhi Liu, China

Current position, grades: Master's degree.
Scientific interests: symbolic computation.