A Formal Verification Method for Hybrid Systems

Zhao Peng

Jiangsu Second Normal University
Jiangsu Nanjing 210013, China
roc_glacier@163.com

***Abstract* — A hybrid system is a dynamical system with both discrete and continuous state variables, and it has been widely used in many application fields. In this paper we propose a formal verification method for hybrid systems. Particularly, as embedded systems form part of hybrid systems, we use embedded system to test the effectiveness of our proposed algorithm. The design of embedded system is mainly composed of five levels: system level, transactional level, behavioral level, RT level, and gate level. In order to effectively tackle the task of embedded systems' formal verification, 'priced timed' automata is utilized to add cost variables on edges and locations. Then the proposed verification algorithm is designed via relaxing the timing constraints, and an iterative mode is utilized to implement the process of embedded system's formal verification. In particular, the proposed algorithm is repeated until the ending condition is satisfied. A case study is used to evaluate the proposed method. In this case study, verification time of the ring configuration sub-systems under different methods is used as the performance criteria. Experimental results demonstrate that compared with other schemes our proposed algorithm can effectively reduce the formal verification time.**

Keywords -- Formal verification*; *Hybrid system*; *Embedded system*; *Timed automata.

I. INTRODUCTION

As is well known that a hybrid system refers to a dynamical system with both discrete and continuous state varying, which is broadly exploited to complex engineering, such as air-traffic control, automotive control, robotics, manned space flight, and embedded systems[1]. Particularly, the high-confidence and safety-critical feature of the application areas has fostered a large and growing body of work on formal approaches for hybrid systems. Furthermore, there are mainly two kinds of formal verification methods: theorem proving and model checking[2].

Considering the infinite state-spaces of hybrid system, more attentions have been paid on the analysis of hybrid system. Hybrid system also can be regarded as a state-transition system, which is made up of a non-trivial mixture of continuous activities and discrete events[3][4]. Moreover, the hybrid system plays an important role in process control systems, in which errors are highly undesirable. Therefore, it is of great importance to utilize the formal verification technology in hybrid system.

In recent years, with the evolution of electronic device, embedded systems have been attracted by more and more researchers, which are computer related technologies involving in each field of engineering, widely and deeply[5]. As a typical hybrid system, embedded system is characterized by both discrete and continuous behaviors. Particularly, the formal approaches of hybrid system are composed of 1) formal analysis, 2) formal verification and 3) validation[6]. In particular, the design of embedded system requires high quality and reliability, hence, the formal verification for both the hardware and software module in embedded system is very crucial.

In recent years, as a powerful tool, formal verification has been applied in many applications. Petros proposed a computer-based method for the formal verification of collaboration patterns in healthcare teams[7]. Eldib et al. proposed the first SMT-solver-based approach for formally verifying the security of a masking countermeasure against network attacks[8]. Woo-Sik utilized the formal verification technology in RFID Authentication Protocol design which is widely used in many industries[9]. Corno et al. tried to introduce correctness, reliability, safety, and security in the design process of SmE and their related components through proposing a design time modeling and formal verification technology[10]. Bolton et al. proposed an approach for automatically obtaining specification properties from task models which enables analysts to utilize formal verification to check for system HAI problems[11]. Furthermore, some other domains have used the formal verification technology, such as flexible manufacturing system[12], monitoring system in aircraft[13], real-time communication protocol[14], model-driven engineering[15], pervasive messaging system[16], Grid computing[17], Information-Based Access Control[18], SysML activity diagram[19], artificial Intelligence[20], and online registration protocol[21].

The rest of this paper is organized as follows. The next section provides the statement of the embedded system design. Section 3 illustrates the proposed formal verification method for the hybrid system based on the timed automata. Based on the above analysis, formal verification case studies are conducted in section 4. Then, the conclusions are illustrated in the final section.

II. STATEMENT OF THE EMBEDDED SYSTEM DESIGN

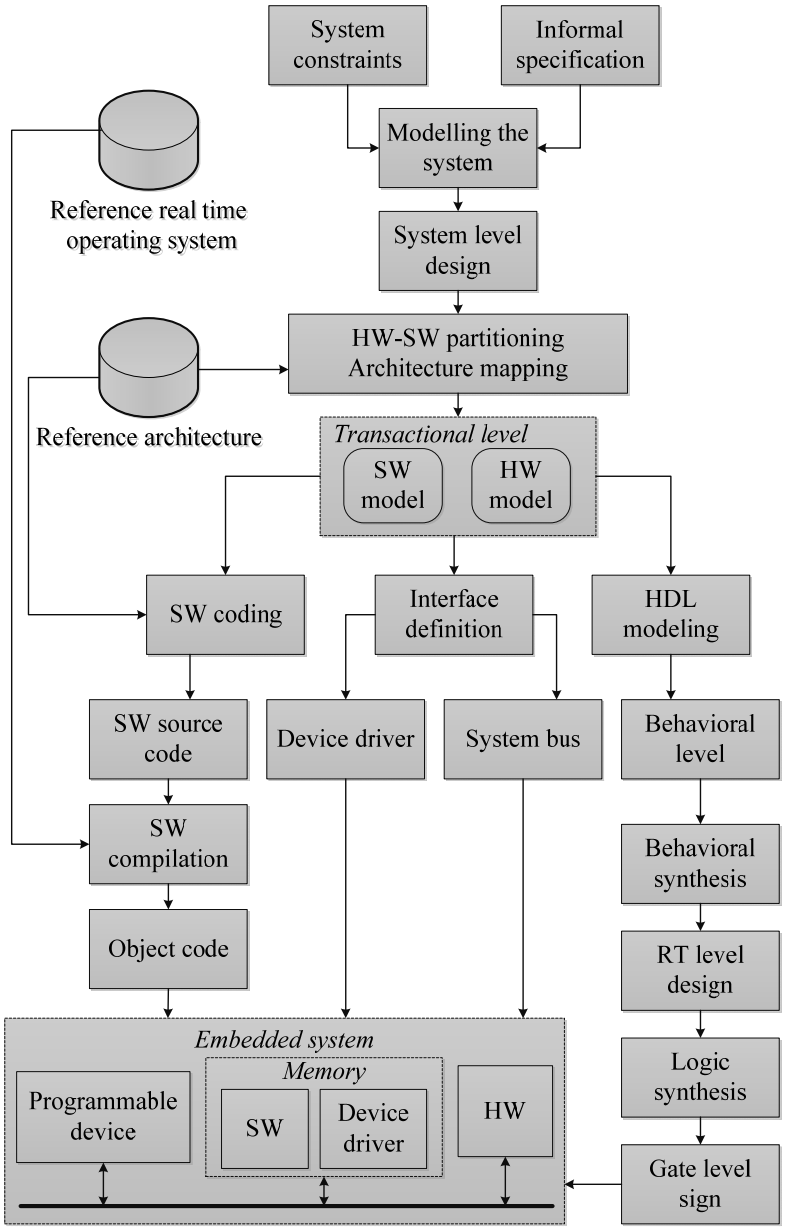

Fig. 1 The process of embedded systems design

An embedded system is defined as a computer system which is a component in a large system and that depends on its own micro-processor[22][23]. Hence, the embedded system can also be regarded as a mix of cooperating hardware and software modules, and it has the ability to provide a wider and more adaptable set of complex functionalities using ASIC and ASIP. For example, typical embedded systems contain controllers for industrial processes, automotive appliances, automatic controlling system, multimedia portable systems, data acquisition systems, and so on. In particularly, the main feature embedded systems lie in that it should continuously react to asynchronous input events.

That is, embedded systems are particularly located in real-time contexts, in which tasks should be scheduled be the pre-defined deadline.

As is shown in Fig.1, the process of embedded systems design is illustrated. Particularly, in this paper, we aim to propose an effective formal verification method and exploit it in the hybrid systems. As a typical hybrid system, embedded system is considered in our method. From Fig.1, we can see that the design of embedded system contains several levels, such as system level, transactional level, behavioral level, RT level, and gate level. In the next section, we will discuss how to design a proposed formal verification method for the hybrid system.

III. THE PROPOSED FORMAL VERIFICATION METHOD FOR THE HYBRID SYSTEM

The main innovations of this paper lie in that we introduce the timed automata technology is our design. Timed automata has been widely used in modelling real-time specifications. Note that timed automata can provide an effective way to extend finite state machines via real valued clocks to model real time processes on continuous time.

As is well known that, most embedded real-time systems are made up of many concurrent components running at various speeds. Hence, formal verification of embedded systems should efficiently solve many states and large ratios of timing constants. We propose a formal verification algorithm for embedded systems using modified timed automata, in which a finite state system is augmented with time measuring devices (that is also named as timers). Afterwards, the definition of timed automata is given as follows.

A. Introduction of the price timed automata

Definition 1 (Timed automata) Supposing that the symbol C refers to a set of valued variables (denoted as clocks), and $|C|$ is satisfied. Let the symbol $G(C)$ be the set of guards on clocks as the conjunctions of the constraints of the form ∞x, in which $c \in C$, $x \in N$ and $\infty \in \{\le, <, >, \ge, =\}$. Moreover, let the symbol $u(C)$ be the clock valuation function: $C \to \left[R \ge 0 \cup \{\infty\}\right]^n$. Then a timed automata is defined as a tuple (L, l_0, C, A, E, I) as follows.

1) L: A set of locations which indicate the system state when a transition has been done.

2) l_0: it refers to an initial location, and it is belonged to the set L.

3) C: it means a set of clocks, and all clocks in C are set to 0 at the location l_0. After running a transition, all the clocks should be reset.

4) A: it denotes a set of actions.

5) $I: L \to G(C)$: it is an invariant, that can allocate guards to specific locations.

6) $E \subseteq L \times A \times G(C) \times 2^C \times L$: it refers to a set of transition with a specific action, a give guard, and a set of clocks

Based on the above definition, a transition in the timed automata can be represented as follows.

$$l \xrightarrow{(a,g,r)} l^* \tag{1}$$

where symbol l and l^* denote the source location and the destination location respectively. a is the action which triggers the transition, g is the clock guard and r refers to the subset of clocks. Particularly, in this paper, by introducing the factor of cost, a modified timed automata (denoted as Priced timed automata) is utilized. In the new version of timed automata, the standard timed automata is revised by adding cost variables on edges and locations.

Assuming that $B(X)$ refers to the formula which represents the conjunctions of atomic constraints of the form $x_i \infty n$, where $x_i \in X$, $n \in N$, then, the definition of priced time automata is given as follows.

Definition 2 (Priced timed automata) We use a five-tuple (L, l_0, E, I, P) to represent the priced timed automata on clocks X, where the symbol L refers to a finite set of locations, $I: L \to B^*(X)$ allocates invariants to locations, and $P: (L \cup E) \to N$ allocate cost to edges and locations. Other symbols are just the same to definition 1.

B. The proposed formal verification algorithm

Supposing that $V = \{x_1, x_2, \cdots, x_n\}$ is a set of timer variables, and Ψ denotes the set of for mulation of the condition $x \le a$, $x \ge b$ or $x\text{-}y \le c$, where x, y are timer variables, and a, b, c mean the bounds. Furthermore, the timer valuation $\tau: V \to R$ is defined to allocate a real value to each timer variable.

Next, to validate the embedded systems, we define an eight tuple $(\sum, Q, I, TR, V, TO, TM, F)$, in which the symbols are defined as follows:

1) $\sum$ denotes some finite set of the I/O value.

2) Q is finite set of states.

3) I refers to an initial state set, and it is belonged to Q.

4) TR is a transition relation, and it is belonged to $Q \times \Sigma \times Q$.

5) TO denotes a timing obligation.

6) TM means a timer modifier.

7) F is a set of the final states, and it is belonged to Q.

Based on the above definitions, we design the verification algorithm via relaxing the timing constraints. We propose an iterative verification algorithm to complete the task of embedded system's formal verification, and the algorithm is repeated until the ending condition is satisfied.

The aim of this algorithm is to effectively lower the time cost and space cost, and the proposed formal verification algorithm is illustrated as follows.

Algorithm 1: the formal verification algorithm based on timed automata

Input: An eight tuple $\left(\sum ,Q,I,TR,V,TO,TM,F\right)$

(1) $T=\left(\sum ,Q,I,TR,F\right)$

(2) **While** the ending condition for verification is not satisfied

(3) **If** $\left(r=verify\left(T\right)\right)=\varnothing$

(4) **Then** return $\varnothing$

(5) **End If**

(6) **If** $\left(G=analyze\left(T,r\right)\right)=\varnothing$

(7) **Then** return r

(8) **End If**

(9) T=modify$\left(G,T\right)$

(10) **End while**

To demonstrate the effectiveness our proposed algorithm, a case study is provided in the next section.

IV. FORMAL VERIFICATION CASE STUDIES

To make performance evaluation, in this section, we provide a case study for formal verification of the embedded system. In this case study, a ring configuration system is illustrated, in which a number of *n* processing sub-systems are allocated in a ring configuration. Particularly, the structure of a sub-system used in this experiment is given in Fig.3 as follows.

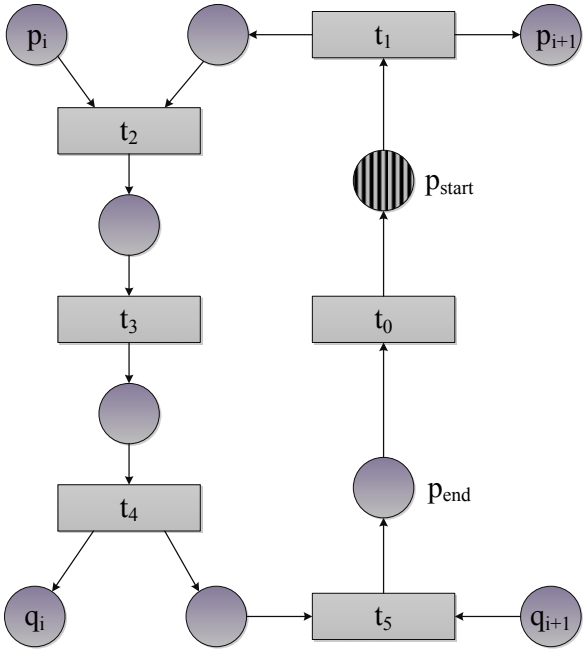

Fig. 2 Structure of a sub-system in the ring configuration system.

In the structure of Fig. 2, each element of the given *n* sub-system should finish before a pre-defined deadline. Particularly, the start of processing for a sub-system is represented as p_{start}, similarly, the end is denoted as p_{end}. Afterwards, the UPPALL tool is used to check the time requirements of the given system. To compare the performance with our method, two approaches are utilized, that is, 1) Naive, and 2) Transformations. The Naive method means the naive translation of the Petri net based Representation for Embedded Systems into timed automata. On the other hand, the second method (named Transformations) represents the embedded system which supports a transformation process. Time of formal verification using different methods are listed in Table.1 as follows.

TABLE. 1 THE RESULTS OF THE FORMAL VERIFICATION FOR THE GIVEN RING CONFIGURATION EXAMPLE

Number of sub-systems	Time of the formal verification process (s)		
	Naive	Transformations	Our method
2	0.129	0.084	0.148
3	2.438	0.613	0.217
4	48.135	8.631	0.675

5	792.36	118.33	6.123
6	13507.97	1237.08	58.42
7	Out of time	19124.62	476.57
8	Out of time	Out of time	3869.43
9	Out of time	Out of time	29574.18

In particular, to make the experimental results more clearly, the data in Table. 1 can also be visualized in Fig. 3 as follows.

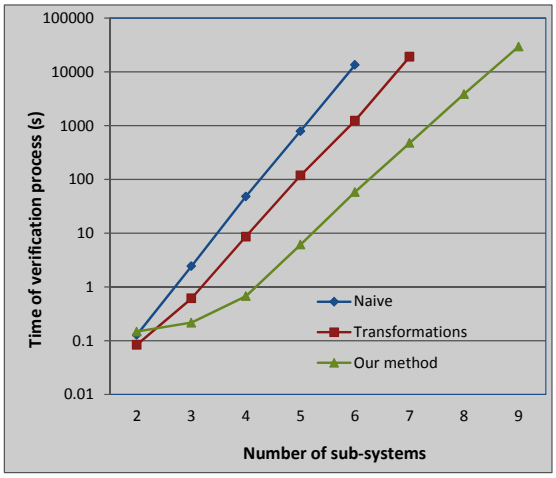

Fig. 3 Verification time of the ring configuration sub-systems for different methods.

Combining all the experimental results together, the conclusions can be drawn that our proposed can effectively save the verification time for embedded systems than other existing methods. As embedded systems are belonged the hybrid systems, our proposed method can be applied in hybrid systems as well.

V. CONCLUSIONS

This paper proposed a formal verification method for the hybrid system. As the embedded system is a type of hybrid systems, we exploit the embedded system to test the effectiveness of our proposed algorithm. The main innovations of this paper lie in that we use the priced timed automata to solve this problem by adding cost variables on edges and locations. A case study is conducted to verify the effectiveness.

In the future, we will extend this work by the following aspects:

(1) We will utilize other types of hybrid system to test this algorithm, such as, air-traffic control, automotive control, robotics, manned space flight and so on.

(2) More influencing factors in the hybrid system design will be considered, such as cost of the space.

ACKNOWLEDGEMENT

Fund Project : The open project on Provincial Department altogether constructing the laboratory base （9011311）;Jiangsu Second Normal University about the twelfth five-year-plan outline for national economic and social development of the People's Republic of China （JSNU-Z-4464）

REFERENCES

[1] Vercelli B., Angella G., Virgili T., Lopez I. Suarez, Pasini M., “Photo-Physical Behaviour of CdSe Nanocrystals/Bis(dithiocarbamate) Linker Multilayered Hybrid Systems”, Journal of Nanoscience and Nanotechnology, vol. 15 , No. 5, pp. 3540-3544, 2015.

[2] Vento Jorge, Blesa Joaquim, Puig Vicenc, Sarrate Ramon, “Set-membership parity space hybrid system diagnosis”, International Journal of Systems Science, vol. 46 , No. 5 , pp. 790-807, 2015.

[3] Park Kwan-Soon, Ok Seung-Yong, “Optimal design of hybrid control system for new and old neighboring buildings”, Journal of Sound and Vibration, vol.33 , No. 336: , pp. 16-31, 2015,.

[4] Torreglosa Juan P., Garcia Pablo, Fernandez Luis M., Jurado Francisco, “Energy dispatching based on predictive controller of an off-grid wind turbine/photovoltaic/hydrogen/battery hybrid system”, Renewable Energy , vol.74, pp.326-336, 2015.

[5] Yuan Jinlong, Zhang Xu, Zhu Xi, Feng Enmin, Yin Hongchao, Xiu Zhilong, “Pathway identification using parallel optimization for a nonlinear hybrid system in batch culture”, Nonlinear Analysis-hybrid Systems, vol. 14, No. 15, pp. 112-131, 2015.

[6] Mirko Loghi, Tiziana Margaria, Graziano Pravadelli, Bernhard Steffen, “Dynamic and FormalVerification of Embedded Systems:A Comparative Survey”, International Journal of Parallel Programming, vol. 33 , No. 6 , pp. 585-611, 2005.

[7] Papapanagiotou Petros, Fleuriot Jacques D., “Formal verification of collaboration patterns in healthcare”, Behaviour & Information Technology, vol.33, no,12, pp.1278-1293, 2014.

[8] Eldib Hassan, Wang Chao, Schaumont Patrick, “Formal Verification of Software Countermeasures against Side-Channel Attacks”, ACM Transactions on Software Engineering and Methodology, vol.24, no.2, pp.30-37, 2014.

[9] Woo-Sik Bae, “Formal Verification of an RFID Authentication Protocol Based on Hash Function and Secret Code”, Wireless Personal Communications, vol. 79 , No. 4 , pp. 2595-2609, 2014.

[10] Corno Fulvio, Sanaullah Muhammad, "Modeling and formal verification of smart environments", Security and Communication Networks, vol. 7 , No. 10 , pp. 1582-1598, 2014.
[11] Bolton Matthew L., Jimenez Noelia, van Paassen Marinus M., Trujillo Maite, "Automatically Generating Specification Properties from Task Models for the Formal Verification of Human-Automation Interaction", IEEE Transactions on Human-machine Systems, vol. 44 , No. 5 , pp. 561-575, 2014.
[12] Carpanzano E., Ferrucci L., Mandrioli D., Mazzolini M., Morzenti A., Rossi M., "Automated formal verification for flexible manufacturing systems", Journal of Intelligent Manufacturing, vol. 25 , No. 5 , pp. 1181-1195, 2014.
[13] Kim Seonmo, Nam Wonhong, Kil Hyunyoung, Park Myunghwan, "Formal Verification of a Gravity-induced Loss-of-Consciousness Monitoring System for Aircraft", Computing in Science & Engineering, vol. 16 , No. 5, pp. 96-103, 2014.
[14] Zhou Rui, Li Chanjuan, Min Rong, Yu Qi, et al., "On design and formal verification of SNSP: a novel real-time communication protocol for safety-critical applications", Journal of Supercomputing, vol. 69 , No. 3 , pp. 1254-1283, 2014.
[15] Gonzalez Carlos A., Cabot Jordi, "Formal verification of static software models in MDE: A systematic Review", Information and Software Technology, vol. 56 , No. 8 , pp. 821-838, 2014.
[16] Konur Savas, Fisher Michael, Dobson Simon, Knox Stephen, "Formal verification of a pervasive messaging system", Formal Aspects of Computing, vol.26, no.4, pp. 677-689, 2014
[17] Souri Alireza, Navimipour Nima Jafari, "Behavioral modeling and formal verification of a resource discovery approach in Grid computing", Expert Systems with Applications, vol. 41 , No. 8 , pp. 3831-3849, 2014.
[18] Lamilla Alvarez Pablo, Takata Yoshiaki, "A Formal Verification of a Subset of Information-Based Access Control Based on Extended Weighted Pushdown System", IEICE Transactions on Information and Systems, vol. 97 , No. 5 , pp. 1149-1159, 2014.
[19] Ouchani Samir, Mohamed Otmane Ait, Debbabi Mourad, "A formal verification framework for SysML activity diagrams", Expert Systems with Applications, vol. 41 , No. 6 , pp. 2713-2728, 2014.
[20] Velev Miroslav N., Franco John, "Application of constraints to formal verification and artificial Intelligence", Annals of Mathematics and Artificial Intelligence, vol. 4 , No. 70 , pp. 313-314, 2014.
[21] Diaz Jesus, Arroyo David, Rodriguez Francisco B., "On securing online registration protocols: Formal verification of a new proposal", Knowledge-based Systems, vol. 16, No. 59 , pp. 149-158, 2014.
[22] Jung YoungHoon, Carloni Luca P., Petracca Michele, "Cloud-Aided Design for Distributed Embedded Systems", IEEE Design & Test, vol. 31 , No. 3 , pp. 32-40, 2014.
[23] Huang Jia, Barner Simon, Raabe Andreas, Buck Christian, Knoll Alois, "A framework for reliability-aware embedded system design on multiprocessor platforms", Microprocessors and Microsystems, vol. 38 , No. 6 , pp. 539-551, 2014.