Available online at www.ijpe-online.com

International Journal of Performability Engineering

Vol. 13, No. 6, October 2017, pp. 886-896
DOI: 10.23940/ijpe.17.06.p10.886896

The Quantitative Analysis of Approximate Correctness for Real-Time Systems

Yanfang Ma[a,*], Liang Chen[b], Haiyu Pan[c]

[a]*School of Computer and Science, Huaibei Normal University, Huaibei,China, 235000*
[b]*School of Mathematical Science, Huaibei Normal University, Huaibei,China, 235000*
[c]*Taizhou University, Taizhou,China, 235000*

Abstract

In order to formalize the correctness of real-time systems, strong timed bisimulation in TCCS has been proposed to characterize the relation between implementation and specification. The usual action and time delay must be the same in strong timed bisimulation. However, in some real situations, many real-timed systems can not satisfy the exact match. In this paper, in order to characterize the approximate usual action and time delay, the strong timed bisimulation in TCCS is generalized to numerical version. Firstly, the definition of global timed bisimulation index of a binary relation is established to describe the relation between implementation and specification. Secondly, in order to quantify the approximate degree between implementation and specification, the global timed λ-bisimulation is defined. Finally, the congruence of the global timed λ-bisimulation is proven to guarantee the modular development and hierarchic design methods which are used in the real software development.

Keywords: correctness; metric; formalization; real-time system

(Submitted on July 25, 2017; Revised on August 30, 2017; Accepted on September 15, 2017)

(This paper was presented at the Third International Symposium on System and Software Reliability.)

1. Introduction

The real-time system exists in the ordiray life. How to gurantee the correctness of the system is an important research area. For a real-time system, the difference from a classical system is that some critical time requirements must be satisfied in the implementation. For example, in the aircraft and spacecraft flight controls systems, there exists a number of real-time programs. In these real-time programs, many time requirements should be satisfied. Similar to the classical software system, some formal models have been proposed to abstract the real-time system. For example, timed automata[9,10], timed CCS(Communication and Concurrency System, abbreviated TCCS) [18], real-time CSP(Communicating Sequential Processes) [3] and timed extension of LOTOS and so on. In these extensions of process calculi, one way is to describe the time directly with actions, e.g. [6,8], and another way is to use a special action to characterize the passage of time, e.g. [12,22,24]. And just like the classical processes calculus, strong and weak timed bisimulation equivalences are presented to formalize the relation between implementation and its specification. If two processes are strong timed bisimulation, then they should perform the same actions and have same time requirement for the same action. These bisimulation relations provide some formal methods to verify the correctness of a real-time system. If a certain behavior equivalence can be established to describe the relation between specification and implementation for a real-time system, then the system will be treated as correctness.

However, in real situations, such as mobile phones or network applications, when softwares are running on the physical equipment, the hardware of the system and external environment may lead to the prerequisites of software that can not be

* Corresponding author.
E-mail address: clmyf@163.com.

holden. For example, a video software may not meet the time requirement, but it can be played at some extent. There are many theories to extend the bisimulation in processes calculus to quantify the degree that implementation approximates specification. In [17], a parameter is used in the timed automata semantics to express the implementation approximates its specification. In [19], the author extended the traces semantics of timed automaton to quantitative case. A metric is put on the set of traces. A trace is robustly accepted if and only if a tube around that trace is classically accepted. In [20], the authors proposed several metrics that may be used to measure the timing performance of a design. These metrics were analyzed using workloads from both real-world task systems and randomly generated task systems. In [7], the authors modeled and analyzed stochastic hybrid systems under quantitative aspects. In [2], based on the probabilistic of the literature[7], the authors added a probabilistic for non probabilistic timed systems in order to establish linear-time properties assuming fairness on actions and delays. In [13], based on the model of partially-observable and non-deterministic timed automata, a new framework of black-box conformance testing for a real-time system was proposed. In [1], the distance function on the set of actions is established by the amount of time to which event structures do 'agree'. A generalization to equivalence classes of timed event structures was proposed. In [11], based on Bertoni's studies, the authors expanded a statistical framework for the synthesis of succinct quantum finite automata, and discussed its adaptation to the case of multiperiodic events and languages. In [4], the authors used formalization method to establish a model for the specification of PALS. In 2001, based on the metric on the time field, Ying [14] extended the classical strong timed bisimulation of real-time system to the numberable version and defined strong timed bisimulation indexes. According to the strong time bisimulation indexes, the degree to which a binary relation on timed proccess is strong timed bisimulation is computed. Noticing that, in this bisimulation index, the common observable action must be exactly matched by itself, but the action that represents time delay may be performed by other approximate time delay.

For a real-time system, strong timed bisimulation is often chosen to evaluate the correctness of the system. The conditions owning the same common actions and time delay are sometimes unpractical. In some situations, we often meet two processes that can not own the same common action and time delay. However, these processes are more or less strong timed bisimulation. That means when a process can run a usual observable action, then another process may execute one action that is highly similar to this observable action that the previous object executed. On the other hand, when the process has a time delay, then another process can own an approximate time delay. In this paper, the author will try to build mathematical tools to describe this kind of approximate bisimulation. Since our aim is to get approximate strong timed bisimulation from the common action and time delay, we call our approximate bisimulation global in order to distinguish from Ying's timed λ-bisimulation.

As opposed to the literature [15,21], there are two kinds of metric in our studies. In order to measure the similarity between usual actions, we put a metric on the set of Γ. On the other hand, since the time domain is $\Theta = [0,\infty)$, in order to characterize the approximate time delay, the metric on the time field is built on the natural number metric. The global timed bisimulation index can be obtained from two cases. One case is to establish the approximate degree that two processes can perform the similar action. Another case is to obtain the approximate degree that they perform similar time delay. Furthermore, the global timed λ-bisimulation is defined. In order to describe the modular and hierarchic development methods in designing softwares, the substitutivity laws of global timed λ-bisimulation is stated.

This paper is organized as follows: the syntax and semantics of TCCS are reviewed in Section 2. The global timed bisimulation index is established in Section 3. And, in this section, some important algebraical properties of the global timed bisimulation index are researched. The substitutivity laws of global timed λ-bisimulation under various combinators is proven in Section 4. In Section 5, we present the conclusions and future work.

2. Preliminaries

In this section, the syntax and semantics of timed CCS are recalled. They are very similar to CCS's syntax and semantics. The detailing information about TCCS can be found in the literature of Ying [14].

Generally, we suppose that the time domain is the set of nonnegative reals, written as $\Theta = [0,\infty]$. The elements of Θ are represented by $t, u, \cdots$. Suppose $\Gamma = \Delta \cup \overline{\Delta}$ be the set of labels, where Δ is the set of action names and $\overline{\Delta} = \{\overline{a} : a \in \Delta\}$ is the set of conames of actions. Let $Act_t = \Gamma \cup \delta_\Theta$ be the set of all actions, where $\delta_\Theta = \{\delta(t) : t \in \Theta\}$ is the set of time delays. For the convenience, the elements of Δ are written by $a, b, \cdots$. The elements of Γ are ranged by $l, l', \cdots$ and the elements of Act_t are represented by $\alpha, \beta, \cdots$. If there is a function $f : \Gamma \to \Gamma$ such that $f(\bar{l}) = \overline{f(l)}$

for every $l \in \Gamma$, then f is called relabeling function. Naturally, if $f(\tau)=\tau$ and $f(\delta(t))=\delta(t)$ for any $t \in \Theta$, then f is an extension mapping from Act_t to itself. Let $\aleph$ be the set of variables, K be the set of constants. If an agent can be described by the following syntax in Definition 1, then the agent is called time process expression. The set of all process expressions is written as ∂_t. The elements of ∂_t is ranged over by $E, F, \cdots$.

Definition 1 (Syntax of TCCS) [18] *The set of all timed processes expressions* ∂_t *including* $\aleph, K$ *is the smallest set that has following expressions:*

$$E ::= \alpha.E \mid \sum_{i \in I} E_i \mid E_1 \mid E_2 \mid E \setminus L \mid E[f].$$

The set of all process expressions without process variables are called processes. We use $\mathcal{P}_t$ to denote the set of all processes, the elements of $\mathcal{P}_t$ is represented by $P, Q, \cdots, W, V, \cdots$ .For any $A \in K$, we assume that there is a defining equation $A \overset{def}{=} P_A$, such as $A = P_A = l.A \in \mathcal{P}_t$.

From the definition of syntax of TCCS, we notice that the only difference between the syntaxes of CCS and TCCS is that time delays $\delta(t)$ are added in the syntax of TCCS, just like $\delta(t).E$. Intuitively, $\delta(t).E$ means that a process will delay t time units, then it executes like E at time $r+t$ when E consumes at time r .

The semantics of TCCS is also given based on structural operational semantics that is proposed by G.D.Plotkin [5]. The detailing definition can refer to the literature [15].

Definition 2 (Semantics of TCCS)[15] *Suppose* $\overset{\alpha}{\rightarrow} \subseteq \partial_t \times \partial_t (\alpha \in Act_t)$ *be the transition relations over the timed labeled transition system* $(\partial_t, Act_t, \{\overset{\alpha}{\rightarrow} : \alpha \in Act_t\})$. *The transition rules are defined as follows:*

$$\text{Null delay}\ \frac{}{P \overset{\delta(0)}{\rightarrow} P}$$

$$\text{Prefix}\ \frac{P \overset{\mu}{\rightarrow} P'}{\delta(0).P \overset{\mu}{\rightarrow} P'} \qquad \frac{}{\delta(t+u).P \overset{\delta(t)}{\rightarrow} \delta(u)P}$$

$$\frac{P \overset{\delta(t)}{\rightarrow} P'}{\delta(u).P \overset{\delta(t+u)}{\rightarrow} P'} \qquad \frac{}{\alpha.E \overset{\alpha}{\rightarrow} E}$$

$$\frac{}{l.P \overset{\delta(t)}{\rightarrow} l.P}$$

$$Sum_j\ \frac{E_j \overset{\alpha}{\rightarrow} E_j'}{\sum_{i \in I} E_i \overset{\alpha}{\rightarrow} E_j'} \qquad \frac{E_j \overset{\delta(t)}{\rightarrow} E_j'\ \text{for each}\ i \in I}{\sum_{i \in I} E_i \overset{\delta(t)}{\rightarrow} E_j'}$$

$$Com_1\ \frac{E \overset{\mu}{\rightarrow} E'}{E \mid F \overset{\mu}{\rightarrow} E' \mid F} \qquad Com_2\ \frac{F \overset{\mu}{\rightarrow} F'}{E \mid F \overset{\mu}{\rightarrow} E \mid F'}$$

$$Com_3\ \frac{E \overset{l}{\rightarrow} E' F \overset{\bar{l}}{\rightarrow} F'}{E \mid F \overset{\tau}{\rightarrow} E' \mid F'} \qquad \frac{E \overset{\delta(t)}{\rightarrow} E' F \overset{\delta(t)}{\rightarrow} F'}{E \mid F \overset{\delta(t)}{\rightarrow} E' \mid F'}$$

$$Res\ \frac{E \overset{\alpha}{\rightarrow} E'}{E \setminus L \overset{\alpha}{\rightarrow} E' \setminus L} \qquad (\alpha, \bar{\alpha} \notin L)$$

$$Rel\ \frac{E \overset{\alpha}{\rightarrow} E'}{E[f] \overset{f(\alpha)}{\rightarrow} E'[f]}$$

$$Con\ \frac{P \overset{\alpha}{\rightarrow} P'}{A \overset{\alpha}{\rightarrow} P'} (A \overset{def}{=} P)$$

In the subsequent sections, we mainly focus on the restriction $(\mathcal{P}_t, Act_t, \{\overset{\alpha}{\rightarrow}|_{\mathcal{P}_t} : \alpha \in Act_t\})$ of $(\partial_t, Act_t, \{\overset{\alpha}{\rightarrow} : \alpha \in Act_t\})$ on $\mathcal{P}_t$, where for each $\alpha \in Act_t$, $\overset{\alpha}{\rightarrow}|_{\mathcal{P}_t} = \overset{\alpha}{\rightarrow} \cap (\mathcal{P}_t \times \mathcal{P}_t)$ is restriction of $\overset{\alpha}{\rightarrow}$ on $\mathcal{P}_t$. For convenience, $\overset{\alpha}{\rightarrow}|_{\mathcal{P}_t}$ can be represented as $\overset{\alpha}{\rightarrow}$.

In [23], the author define the notion of bisimulation on the timed CCS in the standard way [18], called strong timed bisimulation. According to the definition of strong timed bisimulation, a usual action must be matched by the same action and a time delay must be exactly same delay interval.

Definition 3 (Strong time simulation) *Given a binary relation* $S \subseteq \mathcal{P}_t \times \mathcal{P}_t$, *if* S *satisfies the following conditions,then* S *is called a strong timed simulation. The conditions are that if* $(W, V) \in S$ *implies that for all* $\alpha \in \Gamma \cup \{\tau\}$ *and* $t \in [0, \infty)$,

(1) if $W \overset{\alpha}{\rightarrow} W'$, then there exists $V' \in \mathcal{P}_t$ such that $V \overset{\alpha}{\rightarrow} V'$ and $(W', V') \in S$;

(2) if $W \overset{\delta(t)}{\rightarrow} W'$, then there exists $V' \in \mathcal{P}_t$, $V \overset{\delta(t)}{\rightarrow} V'$ and $(W', V') \in S$;

When S is symmetrical relation, then S is called a strong timed bisimulation. The largest strong timed bisimulation is called strong timed equivalence, denoted $\sim$.

But, in the real situations, when we choose strong time bisimulation to check the correctness of system, an exact match of time delay is very strict. In some real-time system, it is usual to meet the time delay but not satisfy the requirement. Ying presented bisimulation indexes to deal with the difference of time constraints between implementation and specification for a real-time system, which allows the approximate time delay. In order to obtain the bisimulation indexes, a natural metric on Act_t was builded in [15]. For any $t, t' \in \Theta, \mu, \mu' \in \Gamma$,

$$\rho(\delta(t), \delta(t')) = |t - t'|,$$
$$\rho(\delta(t), \mu) = \rho(\mu, \delta(t)) = \infty,$$
$$\rho(\mu, \mu') = 0, if \ \mu = \mu',$$
$$\rho(\mu, \mu') = \infty, otherwise.$$

Furthermore, based on bisimulation indexes, λ - strong bisimulation was proposed, the definition is shown as follows.

Definition 4 (λ **-strong bisimulation**) [14] *Given a binary relation* S *on processes set, if* S *satisfies the following conditions, then* S *is denoted as a* λ *-strong bisimulation in TCCS. The conditions are that for* $(W, V) \in S$, $\mu \in \Gamma \cup \{\tau\}$ *and* $t \in \Theta$

- if $W \overset{\mu}{\rightarrow} W'$, then there is $V' \in \mathcal{P}_t$ such that $V \overset{\mu}{\rightarrow} V'$ and $(W', V') \in S$
- if $V \overset{\mu}{\rightarrow} V'$, then there is $W' \in \mathcal{P}_t$, $W \overset{\mu}{\rightarrow} W'$ and $(W', V') \in S$
- if $W \overset{\delta(t)}{\rightarrow} W'$ and $\lambda' > \lambda$, then there is $V' \in \mathcal{P}_t$ and t' such that $V \overset{\delta(t')}{\rightarrow} V'$, $|t - t'| < \lambda'$ and $(W', V') \in S$
- if $V \overset{\delta(t)}{\rightarrow} V'$ and $\lambda' > \lambda$, then there is $W' \in \mathcal{P}_t$ and t' such that $W \overset{\delta(t')}{\rightarrow} W'$, $|t - t'| < \lambda'$ and $(W', V') \in S$.

If there exists λ -strong bisimulation S such that $(W, V) \in S$, then we can write $W \sim^{\lambda} V$.

In Definition 4, it is easy to notice that the usual action should be matched by itself, and only the time delay may be performed by the approximate time delay. However, in the real situations, there exists some approximate implementations which can execute the similar action and approximate time delay. Therefore, in the next section, we will try to establish the approximate strong timed bisimulation from similar action and different time delay.

3. The global approximate bisimulation

For the approximate version of strong timed bisimulation, Ying has proposed λ-strong bisimulation, where the time delay can be simulated by the approximate time delay. However, in some real-time system, such as video-played system, there exists the usual action is simulated by the similar action. Therefore, we will establish the mathematical tools to describe this kind of approximation. In this section, we generalize strong timed bisimulation to the global timed bisimulation index from approximate action and different time delay. The global timed bisimulation states the approximate degree that a binary relation becomes a strong timed bisimulation. Some algebraic properties of strong timed bisimulation are also extended to the global timed bisimulation index. Next, the definition of metric space will be reviewed.

Definition 5 (Metric space)[16] *Suppose* $D \neq \varnothing, \eta : D \times D \to [0,\infty]$ *be a function. If* η *satisfies the following conditions, then* (D,η) *is known as a metric space:*

(1) $\eta(x,y)=0$ if and only if $x=y$;
(2) $\eta(x,y)=\eta(y,x)$; and
(3) $\eta(x,z)\le\eta(x,y)+\eta(y,z)$ for any $x,y,z\in D$.

Suppose η be a metric on Δ. As expected, we define a function η' from $\Gamma\cup\{\tau\}\times\Gamma\cup\{\tau\}$ to $[0,\infty]$: for any $a,b\in\Delta$,

(1) $\eta'(\tau,\tau)=0$.
(2) $\eta'(a,b)=\eta'(\bar{a},\bar{b})=\eta(a,b)$.
(3) $\eta'(\tau,b)=\eta'(a,\tau)=\eta'(\tau,\bar{b})=\eta'(\bar{a},\tau)=\eta(a,\bar{b})=\eta(\bar{a},b)=\infty.$

According to the definition of metric, it is easy to know η' is a metric on $\Gamma\cup\{\tau\}$. For simplicity, we always denote η for η'. Then the numerical generalization of Definition 3 will be defined.

Definition 6 (Global timed bisimulation index) *Given a timed labeled transition system* $(\mathcal{P}_t, Act_t, \{\stackrel{\alpha}{\to} : \alpha\in Act_t\})$. η *is a metric on the set* $\Gamma\cup\{\tau\}$. $R\subseteq\mathcal{P}_t\times\mathcal{P}_t$ *is a binary relation on* $\mathcal{P}_t$. *For all* $\alpha\in Act_t$, *if* $\alpha\in\Gamma\cup\{\tau\}$, *we compute:*

$$b_R(W,V';\alpha)=\inf\{\eta(\alpha,\mu):\mu\in\Gamma\cup\{\tau\} \text{ and there is } W'\in\mathcal{P}_t \text{ such that } W\stackrel{\mu}{\to}W' \text{ and } (W',V')\in R\};$$

if $\alpha=\delta(t)\in\delta_\Theta$, then we define:

$$b_R(W,V';\alpha)=\inf\{|t-t'|:t'\in\Theta \text{ and there is } W'\in\mathcal{P}_t \text{ such that } W\stackrel{\delta(t')}{\to}W' \text{ and } (W',V')\in R\};$$

$$b_R(W,V)=\sup\{b_R(W,V';\alpha):V'\in\mathcal{P}_t \text{ and } V\stackrel{\alpha}{\to}V'\};$$

$$b_R=\sup\{\max[b_R(W,V),b_{R^{-1}}(V,W)]:(W,V)\in R\};$$

Then, b_R is denoted as the global timed bisimulation index of R.

Proposition 1 *Let* $(\mathcal{P}_t, Act_t, \{\stackrel{\alpha}{\to} : \alpha\in Act_t\})$ *be a timed labeled transition system.*

(1) If R is strong timed bisimulation in TCCS, then $b_R=0.$

(2) $b_{R^{-1}} = b_R$

(3) $b_{R_0 \circ R_2} \le b_{R_1} + b_{R_2}$.

(4) $b_{\bigcup_{i\in I} R_i} \le \sup_{i\in I} b_{R_i}$.

Proof According to Definition 6, it is easy to get the proofs of (1) and (2).

(3) If $b_{R_1} = \infty$ or $b_{R_2} = \infty$, then it is clear. Next, suppose that $b_{R_1} < \infty$ and $b_{R_2} < \infty$. Then we can obtain sequences $\{\lambda_{1n}\}$ and $\{\lambda_{2n}\}$, such that $b_{R_1} < \lambda_{1n}$ and $b_{R_2} < \lambda_{2n} (n = 1,2,\cdots)$, $\lim_{n\to\infty} \lambda_{1n} = b_{R_1}$, $\lim_{n\to\infty} \lambda_{2n} = b_{R_2}$. For any $P,W \in \mathcal{P}_t$, if $(P,W) \in R_1 \circ R_2$, then there has $Q \in \mathcal{P}_t$ with $(P,Q) \in R_1$ and $(Q,W) \in R_2$. For any $P' \in \mathcal{P}_t$ and $\alpha \in Act_t$, $P \xrightarrow{\alpha} P'$, if $\alpha \in \Gamma \cup \{\tau\}$, then $b_{R_1} \le \lambda_{1n}$ leads to there have $Q' \in \mathcal{P}_t$ and $\mu \in \Gamma \cup \{\tau\}$ such that $Q \xrightarrow{\mu} Q'$ and $\eta(\alpha,\mu) \le \lambda_{1n}$ and $(P',Q') \in R_1$. Furthermore, by $b_{R_2} \le \lambda_{2n}$, we can obtain $W' \in \mathcal{P}_t$ and $\nu \in \Gamma \cup \{\tau\}$ such that $W \xrightarrow{\nu} W'$, $(Q',W') \in R_2$ and $\eta(\mu,\nu) \le \lambda_{2n}$. Therefore, $\eta(\alpha,\nu) \le \eta(\alpha,\mu) + \eta(\mu,\nu) \le \lambda_{1n} + \lambda_{2n}$, and $(P',W') \in R_1 \circ R_2$; If $\alpha = \delta(t) \in \delta_\Theta$, then $b_{R_1} \le \lambda_{1n}$ leads to we have $Q' \in \mathcal{P}_t$ and $\delta(t') \in \delta_\Theta$ such that $Q \xrightarrow{\delta(t')} Q'$ and $|t - t'| \le \lambda_{1n}$ and $(P',Q') \in R_1$. Furthermore, by $b_{R_2} \le \lambda_{2n}$, we can get $W' \in \mathcal{P}_t$ and $\delta(t'') \in \delta_\Theta$ such that $W \xrightarrow{\delta(t'')} W'$ such that $|t - t''| \le \lambda_{2n}$ and $(Q',W') \in R_2$. Thus, $(P',W') \in R_1 \circ R_2$, $|t - t''| \le |t - t'| + |t' - t''| \le \lambda_{1n} + \lambda_{2n}$, and $b_{R_1 \circ R_2} \le \lim_{n\to\infty} (\lambda_{1n} + \lambda_{2n}) = \lim_{n\to\infty} \lambda_{1n} + \lim_{n\to\infty} \lambda_{1n} = b_{R_1} + b_{R_2}$.

(4) similar to (3).

From this definition, we can see that the difference from the paper in Ying's paper is that the common action may be matched by the similar actions, and the time delay may be performed by the approximate time delay.

Definition 7 *Given a timed labeled transition system* $\sigma = (\mathcal{P}_t, Act_t, \{\xrightarrow{\alpha} : \alpha \in Act_t\})$. $R \subseteq \mathcal{P}_t \times \mathcal{P}_t$ *is a binary relation,* $\lambda \in [0,\infty)$. *If* $b_R \le \lambda$, *then* R *is denoted as a global timed* λ *-bisimulation.*

Clearly, any a binary relation R on $\mathcal{P}_t \times \mathcal{P}_t$ is a global timed ∞-bisimulation. If $\lambda_1 \le \lambda_2$ and R is global timed λ_1-bisimulation, then we can obtain R is a global timed λ_2-bisimulation. Furthermore, when R is global timed λ_i-bisimulation for every $i \in I$, then we have R is a global timed $\inf_{i\in I} \lambda_i$-bisimulation.

Corollary 1 *Given a timed labeled transition system* $\sigma = (\mathcal{P}_t, Act_t, \{\xrightarrow{\alpha} : \alpha \in Act_t\})$, $R \subseteq \mathcal{P}_t \times \mathcal{P}_t$ *is a binary relation, then we can get the following properties:*

(1) whenever R is a strong timed bisimulation, then R is a global timed 0-bisimulation.

(2) whenever R is a λ-strong bisimulation, then R is a global timed λ-bisimulation.

(3) whenever R is a global timed λ-bisimulation, if and only if so is R^{-1}.

(4) whenever R is a λ-bisimulation in Definition4, then R is a global timed λ - bisimulation.

(5) whenever R^i is a global timed λ_i- bisimulation $(i = 1,2)$, then $R^1 \circ R^2$ is a global timed $\lambda_1 + \lambda_2$- bisimulation.

(6) whenever R^i is a glob al timed λ-bisimulation $(i \in I)$, so is $\bigcup_{i\in I} R_e^i$.

From the definition of global timed λ-bisimulation, it is natural to build the global timed λ-bisimilarity .

Definition 8 *Given a timed labeled transition system* $\sigma = (\mathcal{P}_t, Act_t, \{\xrightarrow{\alpha} : \alpha \in Act_t\})$, $R \subseteq \mathcal{P}_t \times \mathcal{P}_t$ *is a binary relation. For any* $\lambda \in [0,\infty)$, *the global timed* λ*-bisimilar is defined as follows:*

$$\sim^g_\lambda = \bigcup \{R \mid R \text{ is a global timed } \lambda\text{-bisimulation}\}$$

This definition tells us if there exists a global timed λ-bisimulation R satisfying $(P,Q) \in R,$ then P and Q are called global timed λ-bisimilar, written by $P \sim^g_\lambda Q$.

In the following, we will focus on some algebraic properties of global timed λ-bisimulation. These properties can help the developer to formalize the course of design software.

Proposition 2 *Let* $(\mathcal{P}_t, Act_t, \{\xrightarrow{\alpha} : \alpha \in Act_t\})$ *be a timed labeled transition system. Then*

(1) $\sim \subseteq \sim^g_0$. If $\lambda_1 \le \lambda_2$, then $\sim^g_{\lambda_1} \subseteq \sim^g_{\lambda_2}$.

(2) For any $\lambda \in [0,\infty)$, $\sim^g_\lambda$ is a global timed λ-bisimulation and it is reflexive and symmetric; $\sim^g_{\lambda_1} \circ \sim^g_{\lambda_2} = \sim^g_{\lambda_1+\lambda_2}$.

Proposition 3 *Let* $(\mathcal{P}_t, Act_t, \{\xrightarrow{\alpha} : \alpha \in Act_t\})$ *be a timed labeled transition system.* $P_1 \sim^g_\lambda P_2$ *if and only if* $b_{\sim^g_\lambda}(P_1,P_2) \le \lambda$ *and* $b_{\sim^g_\lambda}(P_2,P_1) \le \lambda$.

Proof If $P_1 \sim^g_\lambda P_2$, then we have that $b_{\sim^g_\lambda}(P_1,P_2) \le \lambda$ and $b_{\sim^g_\lambda}(P_2,P_1) \le \lambda$

Conversely, we define a binary relation on the set of time processes $\mathcal{P}_t$. It is that:

$$R = \{(P_1,P_2) \mid b_{\sim^g_\lambda}(P_1,P_2) \le \lambda \text{ and } b_{\sim^g_\lambda}(P_2,P_1) \le \lambda\},$$

Next, we need to prove that $b_R \le \lambda$.

In fact, if $(P_1,P_2) \in R$, then from $b_{\sim^g_\lambda}(P_1,P_2) \le \lambda$ and $b_{\sim^g_\lambda}(P_2,P_1) \le \lambda$, we know that for any $P_1 \in \mathcal{P}_t$, $\alpha \in Act_t$ and $P_1 \xrightarrow{\alpha} P_1'$, if $\alpha \in \Gamma \cup \{\tau\}$, then for any $n \le 1$, we have $P_2' \in \mathcal{P}_t$ and $\mu \in \Gamma \cup \{\tau\}$ such that $P_2 \xrightarrow{\mu} P_2'$ and $\eta(\alpha,\mu) < \lambda + \frac{1}{n}$, and $(P_1', P_2') \in \sim^g_\lambda$. By noting that $P_1' \sim^g_\lambda P_2'$ implies $(P_1', P_2') \in R$. Thus, we can get that $b_R(P_1,P_2) \le \lambda$. On the other hand, if $\alpha \in \delta_\Theta$, then for any $n \le 1$, we have that $P_2' \in \mathcal{P}_t$ and $\delta(t') \in \Theta$ such that $P_2 \xrightarrow{\delta(t')} P_2'$ and $|t - t'| < \lambda + \frac{1}{n}$ and $(P_1', P_2') \in \sim^g_\lambda$. Therefore, $b_R(P_1,P_2) \le \lambda$. Similarly, $b_{R^{-1}}(P_2,P_1) \le \lambda$. By the definition of global λ-bisimulation, R is a gobal timed λ-bisimulation.

4. The substitutivity laws of global timed λ-bisimulation

In the real word situations, for the developer of complicated software, the modular and hierarchic method are often used to design the software. That means the software is developed by combining some modulars. Therefore, when we use formalize method to verify the correctness of software, it is necessary to guarantee the correctness of combination between modulars. And the combination between modulars is based on some constructions of software. These constructions can be described as some operators in TCCS. Therefore, the substitutivity laws under various operators should be proved in order to ensure the correctness of software. Since we focus on establishing the approximate correctness of real-time system, and global timed λ -bisimulation can be used to quantify the correctness, the substitutivity laws of global timed λ -bisimulation should be discussed in this section. First, we will introduce two definitions that are necessary for our conclusions.

Definition 9 (λ -round)[15] *Suppose* (M,η) *be a metric space.* Y *is a subset of* M, $\lambda \geq 0.$ *For any element* x, y *of the set* M *, if* $\eta(x,y) \leq \lambda \wedge x \in Y \Rightarrow y \in Y,$ *then* Y *is denoted as* λ *-round.*

Definition 10 (Isomorphism mapping) [12] *Suppose* (M,η) *be a metric space.* $f: M \rightarrow M$ *is a mapping. For any* $x, y \in M$, *if* $\eta(f(x), f(y)) = \eta(x,y),$ *then* f *is called isomorphism mapping.*

Now, for every combinators, the substitutivity laws of global timed λ -bisimulation will be researched.

Proposition 4 *Let* $\sigma = (\mathcal{P}_t, Act_t, \{\xrightarrow{\alpha}: \alpha \in Act_t\})$ *be a timed labeled transition system. If* $P_1 \sim_\lambda^g P_2$ *, then*

(1) $\delta(0).P_1 \sim_\lambda^g \delta(0).P_2;$

(2) $\alpha.P_1 \sim_\lambda^g \alpha.P_2;$

(3) $\delta(u).P_1 \sim_\lambda^g \delta(u).P_2;$

(4) $\delta(t+u).P_1 \sim_\lambda^g \delta(t+u).P_2;$

(5) $l.P_1 \sim_\lambda^g l.P_2;$

(6) $P_1 + W \sim_\lambda^g P_2 + W$, when the sum operator performs the observable actions.

(7) If L is λ -round and $P_1 \sim_\lambda^g P_2$, then $P_1 \backslash L \sim_\lambda^g P_2 \backslash L$;

(8) If f is isomorphism mapping on Act_t, then $P_1[f] \sim_\lambda^g P_2[f]$.

Proof (1)We need to show $b_{\sim_\lambda^g}(\delta(0).P_1, \delta(0).P_2) \leq \lambda$ and $b_{\sim_\lambda^g}(\delta(0).P_2, \delta(0).P_1) \leq \lambda$. By the transition rule, we can obtain that for any $P_1' \in \mathcal{P}_t$, and $\delta(0).P_1 \xrightarrow{\mu} P_1'$, $\mu \in \Gamma \cup \{\tau\}$, then $P_1 \xrightarrow{\mu} P_1'$. Furthermore, $P_1 \sim_\lambda^g P_2$ leads to there exists $P_2' \in \mathcal{P}_t$ and $\nu \in \Gamma \cup \{\tau\}$ such that $P_2 \xrightarrow{\nu} P_2'$ and $\eta(\mu,\nu) \leq \lambda$, $P_1' \sim_\lambda^g P_2'$. The transition rule tells us $\delta(0).P_2 \xrightarrow{\nu} P_2'$. Thus, we can get $P_2' \in \mathcal{P}_t$ and $\nu \in \Gamma \cup \{\tau\}$ such that $\delta(0).P_2 \xrightarrow{\nu} P_2'$ and $\eta(\mu,\nu) \leq \lambda$ and $P_1' \sim_\lambda^g P_2'$, by the definition of $b_{\sim_\lambda^g}$, we have that $b_{\sim_\lambda^g}(\delta(0).P_1, \delta(0).P_2) \leq \lambda$. Similarly, $b_{\sim_\lambda^g}(\delta(0).P_2, \delta(0).P_1) \leq \lambda$ holds. So, $\delta(0).P_1 \sim_\lambda^g \delta(0).P_2$.

(2) It is necessary to prove that $b_{\sim_\lambda^g}(\alpha.P_1, \alpha.P_2) \leq \lambda$ and $b_{\sim_\lambda^g}(\alpha.P_2, \alpha.P_1) \leq \lambda.$ In fact, according to transitions rule, we known that $\alpha.P_1 \xrightarrow{\alpha} P_1$ and $\alpha.P_2 \xrightarrow{\alpha} P_2$. According to the definition of metric η , we can get that $\eta(\alpha,\alpha) = 0 \leq \lambda$. Furthermore, $P_1 \sim_\lambda^g P_2$ can guarantee that $b_{\sim_\lambda^g}(\alpha.P_1, \alpha.P_2) \leq \lambda$ holds. Similarly, $b_{\sim_\lambda^g}(\alpha.P_2, \alpha.P_1) \leq \lambda$ holds. Thus, $\alpha.P_1 \sim_\lambda^g \alpha.P_2$.

(3) We need to prove that $b_{\sim_\lambda^g}(\delta(u).P_1, \delta(u).P_2) \leq \lambda$ and $b_{\sim_\lambda^g}(\delta(u).P_2, \delta(u).P_1) \leq \lambda$. In fact, if $\delta(u).P_1 \xrightarrow{\delta(t+u)} P_1'$, then the transition rule tells us $P_1 \xrightarrow{\delta(t)} P_1'$. And $P_1 \sim_\lambda^g P_2$ leads to there exist $P_2' \in \mathcal{P}_t$ and $\delta(t') \in \delta_\Theta$ such that $P_2 \xrightarrow{\delta(t')} P_2'$, $|t - t'| \leq \lambda$ and $P_1' \sim_\lambda^g P_2'$. Furthermore, according to the transition rules, we have that $\delta(u).P_2 \xrightarrow{\delta(t'+u)} P_2'$,

$|t+u-(t'+u)|=|t-t'|\leq\lambda$ and $P_1'\sim_\lambda^g P_2'$. Therefore, according to the definition of $b_{\sim_\lambda^g}$, we have $b_{\sim_\lambda^g}(\delta(u).P_1,\delta(u).P_2)\leq\lambda$ holds. Similarly, $b_{\sim_\lambda^g}(\delta(u).P_2,\delta(u).P_1)\leq\lambda$ holds. Thus, we have $\delta(u).P_1\sim_\lambda^g\delta(u).P_2$.

(4)We should prove that $b_{\sim_\lambda^g}(\delta(t+u).P_1,\delta(t+u).P_2)\leq\lambda$ and $b_{\sim_\lambda^g}(\delta(t+u).P_2,\delta(t+u).P_1)\leq\lambda$. In fact, according to the transition rule, we have that $\delta(t+u).P_1\xrightarrow{\delta(t)}\delta(u).P_1$, and $\delta(t+u).P_2\xrightarrow{\delta(t)}\delta(u).P_2$. Since $P_1\sim_\lambda^g P_2$, by (3), we know that $\delta(u).P_1\sim_\lambda^g\delta(u).P_2$. And $|t-t|=0\leq\lambda$. Thus, $b_{\sim_\lambda^g}(\delta(t+u).P_1,\delta(t+u).P_2)\leq\lambda$. Similarly, we have $b_{\sim_\lambda^g}(\delta(t+u).P_2,\delta(t+u).P_1)\leq\lambda$. So, $\delta(t+u).P_1\sim_\lambda^g\delta(t+u).P_2$.

(5) It is necessary to prove $b_{\sim_\lambda^g}(l.P_1,l.P_2)\leq\lambda$ and $b_{\sim_\lambda^g}(l.P_2,l.P_1)\leq\lambda$. In fact, if $l.P_1\xrightarrow{l}P_1$, by the transition rules, we have $l.P_2\xrightarrow{l}P_2$. Since $P_1\sim_\lambda^g P_2$, and $\eta(l,l)=0\leq\lambda$. Therefore, $b_{\sim_\lambda^g}(l.P_1,l.P_2)\leq\lambda$. On the other hand, if $l.P_1\xrightarrow{\delta(t)}l.P_1$, and $l.P_{12}\xrightarrow{\delta(t')}l.P_2$, then no matter how long they delayed, our aim is to compare the degree to which they execute the observable actions, therefore, we can get $b_{\sim_\lambda^g}(l.P_1,l.P_2)\leq\lambda$. Similarly, we also get $b_{\sim_\lambda^g}(l.P_2,l.P_1)\leq\lambda$ .Thus, $l.P_1\sim_\lambda^g l.P_2$

(6) We need to show that $b_{\sim_\lambda^g}(P_1+W,P_2+W)\leq\lambda$ and $b_{\sim_\lambda^g}(P_2+W,P_1+W)\leq\lambda$ when the sum process performs the observable actions. In fact, if $P_1+W\xrightarrow{\mu}P_1'$, then by the transition rules, there are two cases: $P_1\xrightarrow{\mu}P_1'$ or $W\xrightarrow{\mu}P_1'$. Case 1: $P_1\xrightarrow{\mu}P_1'$. $P_1\sim_\lambda^g P_2$ leads to there exist $P_2'\in\mathcal{P}_t$ and $\nu\in\Gamma\cup\{\tau\}$ such that $P_2\xrightarrow{\nu}P_2'$, $\eta(\mu,\nu)\leq\lambda$ and $P_1'\sim_\lambda^g P_2'$. Furthermore, by the transition rule, we obtain that $P_2+W\xrightarrow{\nu}P_2'$, $\eta(\mu,\nu)\leq\lambda$ and $P_1'\sim_\lambda^g P_2'$. Thus, $b_{\sim_\lambda^g}(P_1+W,P_2+W)\leq\lambda$. If $W\xrightarrow{\mu}P_1'$, then $P_2+W\xrightarrow{\mu}P_1'$. Since $(P_1',P_1')\in\sim_0^g\subseteq\sim_\lambda^g$, and $\eta(\mu,\mu)=0\leq\lambda$, $b_{\sim_\lambda^g}(\delta(t+u).P_2,\delta(t+u).P_1)\leq\lambda$ holds. Similarly, $b_{\sim_\lambda^g}(P_2+W,P_1+W)\leq\lambda$ holds. Thus, $P_1+W\sim_\lambda^g P_2+W$, when the sum operator performs the observable actions.

(7) We construct a binary relation on the set of timed processes $\mathcal{P}_t$. Let $R=\{(P_1\setminus L,P_2\setminus L):P_1\sim_\lambda^g P_2\}$. Next, we need to show $b_R\leq\lambda$.

If $P_1\setminus L\xrightarrow{\alpha}P_1'$, then $\alpha,\alpha'\notin L\cup\overline{L}$, $P_1\xrightarrow{\alpha}P_1''$ and $P_1'=P_1''\setminus L$. Since $P_1\sim_\lambda^g P_2$, there exist $P_2''\in\mathcal{P}_t$ and $\mu\in\Gamma\cup\{\tau\}$ such that $P_2\xrightarrow{\mu}P_2''$ with $\rho_1(\mu,\alpha)\leq\lambda$, and $P_1''\sim_\lambda^g P_2''$. Since L is λ-round, $\mu,\overline{\mu}\neq L\cup\overline{L}$. By the transition rule, $P_2\setminus L\xrightarrow{\mu}P_2''\setminus L=P_2'$ with $\eta(\mu,\alpha)\leq\lambda$ and $(P_1',P_2')\in R$. Therefore, $b_R\leq\lambda$.

(8) We define a binary relation on the set $\mathcal{P}_t$. Let $R=\{(P_1[f],P_2[f]):P_1\sim_\lambda^g P_2\}$. Next, we only need to show $b_R\leq\lambda$.

In fact, if $P_1[f] \xrightarrow{\alpha} P_1'$, then there exists $\beta \in Act_t$, such that $P_1 \xrightarrow{\beta} P_1''$, $P_1''[f] = P_1'$ and $\alpha = f(\beta)$. If $\alpha \in \Gamma \cup \{\tau\}$, since $P_1 \sim_\lambda^g P_2$, there exist $P_2'' \in \mathcal{P}_t$ and $\gamma \in \Gamma \cup \{\tau\}$ such that $P_2 \xrightarrow{\gamma} P_2''$ with $\eta(\gamma, \beta) \le \lambda$, and $P_1'' \sim_\lambda^g P_2''$. By the transition rule, we have that $P_2[f] \xrightarrow{f(\gamma)} P_2''[f]$, and $\mu(\alpha, f(\gamma)) \le \eta(f(\beta), f(\gamma)) \le \eta(\beta, \gamma) \le \lambda,$ and $(P_1''[f], P_2''[f]) \in R$. On the other hand, if $\alpha = \delta(t) \in \delta_\Theta$, then there exists $\beta = \delta(t) \in Act_t$, such that $P_1 \xrightarrow{\delta(t)} P_1''$, $P_1''[f] = P_1'$ and $\delta(t) = f(\delta(t))$. $P_1 \sim_\lambda^g P_2$ makes there exist $P_2'' \in \mathcal{P}_t$ such that $P_2 \xrightarrow{\delta(t')} P_2''$ and $|t - t'| \le \lambda$ and $P_1'' \sim_\lambda^g P_2''$. By the transition rule, $P_2[f] \xrightarrow{f(\delta(t'))} P_2''[f]$, and $|f(\delta(t)) - f(\delta(t'))| = |\delta(t) - \delta(t')| = |t - t'| \le \lambda$ and $(P_1''[f], P_2''[f]) \in R$. Therefore, $b_R \le \lambda$.

Noticing that, in the substitutivity laws of global timed λ-bisimulation, the sum operator only holds congruence when the processes execute the observable action. It means that when the sum process performs the time delay, it is necessary that every element in sum should have the same time delay. According to the definition of global timed bisimulation, we do not find a suitable way to prove the sum process satisfies the substitutivity property in this case. At the same time, we do not obtain some suitable methods to prove the congruence of parallel operator. In the next work, we will try to modify our model and make the sum and parallel operator hold the substitutivity law.

5. Conclusions

In order to formalize the approximate correctness of real-time system, in this paper, we established the global timed bisimulation index by the metric on the usual action and the metric on the time field. The global timed λ-bisimultaion is established to get the approximate metric between implementation and specification of a real-time system. Furthermore, the substitutivity law of global time λ-bisimuaton is proved to guarantee the modular design and hierarchy development. In the future, one hand, we will try to modify our model and make it has more perfect properties. Another hand, we will design a suitable algorithm to realize the verification of the metric automatically.

Acknowledgements

The work is supported by the National Natural Science Foundation of China (61021004),the Natural Science Foundation of Anhui Province (1508085MA14, 1708085MF159), the Natural Science Foundation of the Anhui Higher Education Institutions (KJ2017A375,KJ2017A391,KJ2016B018) and Qing Lan Project.

References

1. C. Baier, J. Katoen, D. Latella, "Metric Semantics for True Concurrent Real Time," *International Colloquium on Automata, Languages, and Programming*, 254(1–2): 568-579, 1998
2. C. Baier, N. Bertrand, T. Brihaye, T. Brihaye, "Probabilistic and topological semantics for timed automata," In the *Proceeding of the 27th International Conference on Foundations of Software Technology and Theoretical Computer Science*, 4855 :179-191, 2007.
3. C. A. R. Hoare, "Communication Sequential Processes," Prentice Hall, New York, 1985.
4. C. Peter, J. Mesegure, "Formalization and Correctness of the PALS Architectural Pattern for Distributed Real-time System," *Theoretical Computer Science*, 451:1-37, 2012.
5. G. D. Plotkin, "A Structural Approach to Operational Semantics," *Technical Report DAIMI FN-19*, Computer Science Department, Aarhus University,1981.
6. J. C. M. Baeten, J. Bergstra, "Real-Time Process Algebra," *Formal Aspects of Computing*, 3:142-188, 1991.
7. J. Sproston, "Model Checking for Probabilistic Timed Systems," *Lectrue Notes in Compuer Science*, 2925:189-229, 2004.
8. K. G. Larsen, Y. Wang, "Time-Abstract Bisimulation: Implicit Specifications and Decidability," *Information and Comoputation*, 134: 75-101, 1997.
9. L Guoqiang, L Li, "Asynchronous Multi-process Timed Automata," *Softwar Quality Control*, 2017, DOI: 10.1007/s11219-017-9380-8.
10. L Guoqiang , W Yueqing, Y Shoyi, "Updatable Timed Automata with One Updatable Clock," *Science in China Series F Information Sciences*, DOI: 10.1007/s11432-016-9027-y, 2017.
11. M. Bianchia, C. Mereghettib, B. Palano, "Quantum finite automata: Advances on Bertonis ideas," *Theoretical Computer Science,* 664: 39-53, 2017.

12. M. Hennesy, T. Regan, "A Process Algebra for Timed Systems," *Information and Computation*,117: 221-239, 1995.
13. M. Krichen, S. Tripakis, "Conformance Testing for Real-Time Systems," *Formal Methods in System Design*, 2009, 34: 238-304.
14. M. S. Ying, "Topology in Process Calculus: Approximate Correctness and Infinite Evolution of Concurrency Programs," Springer-Verlag, 2001.
15. M. S. Ying, "Bisimulation Indexes and Their Applications," *Theoretical Computer Science*, 275(1-2): 1-68, 2002.
16. R. Engelking, "General Topology," Polish Science, 1977.
17. M. Hennesy, T. Regan, "A Process Algebra for Timed Systems," *Information and Computation*, 117: 221-239, 1995.
18. R. Milner. "Communication and Concurrency," Prentice Hall, New York,1989.
19. V. Gupta, T. A. Henzinger, R. Jagadeesan, "Robust Timed Automata," *Lectrue Notes in Compuer Science*, 1201: 331-345, 1997.
20. X. Hu, R. S. Sambandam, "Multi-Valued Performance Metrics for Real-Time Embedded Systems," Design Automation for Embedded Systems, 5: 5-28, 2000
21. Y. F. Ma, M. Zhang, Y. X. Chen, L.Chen, "Two-thirds Simulation Indexes and Modal Logic Characterization," *Frontiers of Computer Science in China*, 5(4) , 454-471, 2011.
22. Y. Wang, "Real-Time Behaviour of Asynchronous Agents," in *Proceeding of CONCUR'90, Lecture Notes in Computer Science*, 458: 502-520, 1990.
23. Y. Wang, "CCS + Time = an Interleaving Model for Real Time Systems," *Lecture Note in Computer Science*, 1991, 510: 217-228.
24. Y. Wang, "A Calculus of Real Time Systems," Ph.D thesis, Chalmers University,1991.