Automatica 164 (2024) 111644

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

On the verification of detectability for timed discrete event systems ☆

Weijie Dong [a,b], Kuize Zhang [c], Shaoyuan Li [a,b], Xiang Yin [a,b,*]

[a] Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China
[b] Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, China
[c] Department of Computer Science, University of Surrey, Guildford GU2 7XH, UK

ARTICLE INFO

Article history:
Received 8 February 2023
Received in revised form 8 January 2024
Accepted 14 February 2024
Available online 30 March 2024

Keywords:
State estimate
Detectability
Partially-observed timed automata

ABSTRACT

In this paper, we investigate the problem of state estimation and detection in the context of timed discrete-event systems. Specifically, we study the verification of detectability, a fundamental state estimation property for dynamic systems. Existing works on this topic mainly focus on untimed DESs. In some applications, however, real-time information is critical for the purpose of system analysis. To this end, in this paper, we investigate the verification of detectability for timed DESs modeled by partially-observed timed automata. Three notions of detectability, strong detectability, weak detectability and delayed detectability, are studied in a dense-time setting, which characterizes detectability by time elapsing rather than event updating steps. We show that verifying strong detectability and delayed detectability for partially-observed timed automata is decidable by providing verifiable necessary and sufficient conditions. Furthermore, we show that weak detectability is undecidable in the timed setting by reducing the language universality problem for timed automata to the verification problem of weak detectability. Our results extend the detectability analysis of DESs from the untimed setting to a timed setting.

1. Introduction

Engineering cyber–physical systems (CPSs), such as flexible manufacturing systems, intelligent transportation systems and power systems, are generally very complex and operate in open environments. In practice, the user cannot directly obtain the full state information of the system due to observation uncertainties or nondeterminism of the system dynamics. Therefore, one turns to take state estimation process to obtain precise state information so that some subsequent tasks, which rely on state information, can be performed. To this end, it is of our interest to know whether or not the system has some desired properties, which is referred to as *detectability*, so that it has sufficient information to distinguish state under imperfect information.

In this paper, we investigate the state estimation problem in the framework of discrete-event systems (DESs) which are widely used in modeling the high-level logical dynamics in CPSs (Cassandras & Lafortune, 2021). In the context of DESs, state estimation problem was initiated by Caines, Greiner, and Wang (1988) and Ramadge (1986), where the concepts of observability was proposed and the observer structure was constructed to check if one can determine the state of the system uniquely by observations. Later in the work of Shu, Lin, and Ying (2007), detectability was systematically investigated for DESs, where four different types of detectability as well as their verification algorithms were provided. To further characterize different state estimation requirements, several variations of detectability have also been proposed in the literature, including, e.g., delayed detectability (Shu & Lin, 2012), K-detectability (Hadjicostis & Seatzu, 2016), I-detectability (Shu & Lin, 2013b), D-detectability (Balun & Masopust, 2021) and trajectory detectability in Yin, Li, and Wang (2018). The enforcement of detectability, accomplished by supervisory control, has been investigated in Shu and Lin (2013a). Recently, detectability verification has been extended to more complex DESs models, including, e.g., labeled Petri nets (Lan, Tong., & Seatzu, 2021; Masopust & Yin, 2019; Zhang & Giua, 2018, 2020), probabilistic automata (Keroglou & Hadjicostis, 2015, 2017), unambiguous weighted automata (Lai, Lahaye, & Giua, 2020) and hybrid systems (Lin, Wang, Yin, Polis, & Chen, 2022). The reader is referred to the comprehensive textbook (Hadjicostis, 2020) for more details on this topic.

The above mentioned works only consider DESs without *real-time* information. In practice, many real-world engineering

☆ This work was supported by the National Natural Science Foundation of China (62173226, 62061136004) and the Alexander von Humboldt Foundation. The material in this paper was partially presented at the 2022 American Control Conference, June 8-10, 2022, Atlanta, Georgia, USA. This paper was recommended for publication in revised form by Associate Editor Michel Reniers under the direction of Editor Christos G. Cassandras.

* Corresponding author at: Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China.
E-mail addresses: wjd_dollar@sjtu.edu.cn (W. Dong), kuize.zhang@surrey.ac.uk (K. Zhang), syli@sjtu.edu.cn (S. Li), yinxiang@sjtu.edu.cn (X. Yin).

https://doi.org/10.1016/j.automatica.2024.111644
0005-1098/

systems have time constraints when generating events and additional time information is also readily available for the purpose of state estimation. Thus, it is necessary to model and analyze executions of the system with time constraints. Such a real-time issue can be captured by timed automata proposed in the seminal work of Alur and Dill (1994). In the context of detectability analysis, if one can "measure" the time execution, e.g., by having a global clock, then additional information can be obtained to improve the capability of estimating the system. However, this critical time information is ignored in the purely untimed setting.

In this paper, we study the detectability verification problems for timed DESs modeled by partially-observed timed automata. The main contributions of this work are as follows.

- First, we introduce the notions of strong detectability, weak detectability and delayed detectability for timed automata. Different from the existing notions in the untimed setting (Shu & Lin, 2012; Shu et al., 2007), where the number of observable events is used to count the observation delays, here we directly utilize the real time information to describe detectability, which provides a more natural and realistic measure for delays.
- Second, we present effective algorithms for checking strong detectability and delayed detectability for timed automata, respectively. Our approach is to first construct a verification structure based on the original system. Then for each property, we reduce the detectability verification problem to a state search problem in the region graph of the constructed verification structure.
- Third, we show that the verification of weak detectability is undecidable in the timed setting. This result is quite different from the untimed setting (Masopust, 2018; Shu et al., 2007; Zhang, 2017), where weak detectability for finite state automata models is shown to be PSPACE-complete and hence can be verified in exponential time.

We note that state estimation for timed DESs modeled by special subclasses of partially-observed timed automata has been investigated recently by many researchers (Basile & Ferrara, 2021; Gao, Lefebvre, Seatzu, Li, & Giua, 2020; Li, Lefebvre, Hadjicostis, & Li, 2022). For example, Gao et al. (2020) discussed how to perform state estimation for timed automata using λ-observers and, in Li et al. (2022), Li et al. proposed observer construction for constant-time labeled automata (CTLA) to investigate state estimation problem of CTLA. However, both Gao et al. (2020) and Li et al. (2022) do not consider the detectability verification problem as we investigate in this work. Also, our technical approach is different from Gao et al. (2020) and Li et al. (2022), since we do not require the subset construction for observer. Furthermore, Gao et al. (2020) and Li et al. (2022) considers timed automata with only single clock reset at each event transition. Therefore, both of our system model and observation model are more general and subsume the models in Gao et al. (2020) and Li et al. (2022). In Lai, Lahaye, and Giua (2019) and Lai, Lahaye, and Komenda (2022), Lai et al. investigated the observer construction for timed systems by interpreting time into weights and using max-plus automata to model timed systems. In Basile, Cabasino, and Seatzu (2015) and Ma, Li, and Giua (2019), authors investigated state estimation for timed Petri nets. In Lin et al. (2022), detectability of hybrid systems was investigated, where hybrid systems were modeled by hybrid machines and they did not consider the influence of real-time constraints.

In the context of property verification of timed systems, effective algorithms have been proposed for checking (co)diagnosability (Cassez, 2012; Tripakis, 2002). More recently, verification for opacity has also been investigated for timed systems (Ammar, El Touati, Yeddes, & Mullins, 2021; Wang, Zhan, & An, 2018; Zhang, 2021). However, these notions are incomparable to detectability, which has been argued in the untimed setting (Shu et al., 2007). Particularly, for general timed automata that we consider in this work, it has been shown by Cassez (2009) that verification of opacity is undecidable. However, our result reveals that several versions of detectability are still decidable for general timed automata. To the best of our knowledge, detectability for timed automata has not been systematically investigated in the literature.

The outline of this paper is as follows. In Section 2, we provide basic background on timed DESs modeled by partially-observed timed automata. In Section 3, three typical detectability notions, strong detectability, weak detectability and delayed detectability, are introduced for partially-observed timed automata. Verification algorithm for strong detectability is provided in Section 4. In Section 5, we establish the undecidability result for weak detectability. Section 6 shows how to verify delayed detectability for partially-observed timed automata. Finally, we conclude the paper in Section 7. A preliminary version of some results in this paper are presented in Dong, Yin, Zhang, and Li (2022) without proofs. In this work, we provide complete proofs with more detailed examples. Furthermore, Dong et al. (2022) only studied strong detectability and weak detectability, while the present work further extends the results to the case of delayed detectability.

2. Preliminaries

2.1. System model

Let $\mathbb{R}_{\geq 0}$ be the set of non-negative real numbers and $\mathbb{N}$ be the set of natural numbers. A *clock* is a variable whose codomain is $\mathbb{R}_{\geq 0}$ and we denote by $\mathcal{X}$ a finite set of clocks. A *valuation* on $\mathcal{X}$ is a function $v : \mathcal{X} \to \mathbb{R}_{\geq 0}$ that assigns to each clock $x \in \mathcal{X}$ a real value $v(x) \in \mathbb{R}_{\geq 0}$. We denote by $V_{\mathcal{X}}$ the set of all clock valuations on $\mathcal{X}$. Given a valuation $v \in V_{\mathcal{X}}$ and a subset $\mathcal{Y} \subseteq \mathcal{X}$ of clocks, we denote by $v_{[\mathcal{Y}\leftarrow 0]}$ the valuation that sets all clocks in $\mathcal{Y}$ to zero, i.e., $v_{[\mathcal{Y}\leftarrow 0]}(x) = 0$ if $x \in \mathcal{Y}$ and $v_{[\mathcal{Y}\leftarrow 0]} = v(x)$ otherwise. We denote by $\mathbf{0}_{\mathcal{X}}$ the valuation in which all clocks are zero. For $\Delta \in \mathbb{R}_{\geq 0}$, we define the valuation function $v + \Delta$ by: for every $x \in \mathcal{X}$, $(v + \Delta)(x) = v(x) + \Delta$.

An *atomic constraint* is of form $x \sim c$, where $x \in \mathcal{X}$ is a clock, $c \in \mathbb{N}$ is a constant and $\sim \in \{\leq, <, \geq, >, =\}$. Given a valuation v, we say v satisfies the atomic constraint $x \sim c$ if $v(x) \sim c$. Then a *clock constraint* or a *guard* is a conjunction of a finite number of atomic constraints and we denote by $C(\mathcal{X})$ the set of all clock constraints (guards) over $\mathcal{X}$. For any clock constraint $g \in C(\mathcal{X})$ and valuation $v \in V_{\mathcal{X}}$, we denote that v satisfies g by $v \models g$.

In this paper, we consider timed systems modeled by partially-observed timed automata (Alur & Dill, 1994; Henzinger, Nicollin, Sifakis, & Yovine, 1994). Formally, a timed automaton (TA) is a six-tuple

$$A = (Q, q_0, \Sigma, \mathcal{X}, \mathsf{inv}, E),$$

where

- Q is a finite set of discrete states;
- $q_0 \in Q$ is the initial discrete state;
- Σ is a finite alphabet;
- $\mathcal{X}$ is a finite set of clocks;
- $\mathsf{inv} : Q \to C(\mathcal{X})$ is an invariant function that assigns to each state q a clock constraint $\mathsf{inv}(q)$ constraining the time region the system is allowed to stay in q;
- $E \subseteq Q \times C(\mathcal{X}) \times \Sigma \times 2^{\mathcal{X}} \times Q$ is the set of transitions. Specifically, each transition is of form $e = (q, g, \sigma, \mathcal{Y}, q')$, where $q \in Q$ and $q' \in Q$ are, respectively, the initial

and final discrete states of the transition, $\sigma \in \Sigma$ is the corresponding event of the transition, $g \in C(\mathcal{X})$ is the *guard* specifying the time when the transition can be enabled and $\mathcal{Y} \subseteq \mathcal{X}$ is the set of clocks that should be reset to zero after this transition.

Given a timed automaton A, a *timed state* (or simply a state) is a pair $s = (q, v)$, where $q \in Q$ is a discrete state and $v \in V_{\mathcal{X}}$ is a clock valuation such that $v \models \mathsf{inv}(q)$. We denote by $S(A) = Q \times V_{\mathcal{X}}$ the set of all states in A. In particular, the initial state of A is defined by $(q_0, \mathbf{0}_{\mathcal{X}})$. Given a state $s = (q, v)$, its discrete state component is denoted by $\mathsf{dis}(s) = q$.

A finite (infinite) *word* over Σ is a finite (infinite) sequence $\sigma_1 \dots \sigma_n(\dots)$; we denote by Σ^* and Σ^ω, respectively, the set of finite words and the set of infinite words over Σ and define $\Sigma^+ = \Sigma^* \backslash \{\varepsilon\}$, where ε denote the empty word. A *timed word* over Σ is a word over $\mathbb{R}_{\geq 0} \times (\Sigma \cup \{\varepsilon\})$, e.g., $\rho = (\Delta_0, \sigma_0)(\Delta_1, \sigma_1)\cdots$, where (Δ_0, σ_0) denotes that event σ_0 occurs after the system has been started after Δ_0 time elapsing, and $(\Delta_i, \sigma_i), i \in \{1, 2, \dots\}$ means that event σ_i occurs Δ_i time elapsing after the previous event σ_{i-1}. Note that $(\Delta, \varepsilon) \in \mathbb{R}_{\geq 0} \times \{\varepsilon\}$ denotes that there is only Δ time elapsing but has no event occurrence. Thus, if there is an event $(\Delta', \sigma') \in \mathbb{R}_{\geq 0} \times \Sigma$ after (Δ, ε) in a timed word, we combine them by summing up the time elapsing and erasing event (Δ, ε), i.e., $(\Delta, \varepsilon)(\Delta', \sigma') = (\Delta + \Delta', \sigma')$; if there are two events $(\Delta, \varepsilon), (\Delta', \varepsilon)$ in a row, it is equivalent to replacing them by $(\Delta + \Delta', \varepsilon)$, i.e., $(\Delta, \varepsilon)(\Delta', \varepsilon) = (\Delta + \Delta', \varepsilon)$. Given a timed word ρ, we denote by $|\rho|$ the length of ρ, which is the number of events $(\Delta, \sigma) \in \mathbb{R}_{\geq 0} \times \Sigma$ appearing in ρ. We denote by $\mathsf{TW}^*(\Sigma)$ and $\mathsf{TW}^\omega(\Sigma)$, respectively, the set of all finite timed words and the set of all infinite timed words over Σ; we use $\mathsf{TW}(\Sigma) = \mathsf{TW}^*(\Sigma) \cup \mathsf{TW}^\omega(\Sigma)$ to denote all timed words. For any timed word $\rho = (\Delta_0, \sigma_0)(\Delta_1, \sigma_1)\dots$, we define $\mathsf{time}(\rho) = \sum_{i=0}^{|\rho|} \Delta_i$ as the total time elapsing in ρ and define $\mathsf{utw}(\rho) = \sigma_0\sigma_1\dots$ as its untimed word. Given a timed word $\rho \in \mathsf{TW}(\Sigma)$, we define $\mathsf{Pre}(\rho) = \{\rho' \in \mathsf{TW}^*(\Sigma) : \exists \rho'' \in \mathsf{TW}(\Sigma) \text{ s.t. } \rho'\rho'' = \rho \wedge \mathsf{time}(\rho') < \infty\}$ as the set of all finite prefixes of timed word ρ.

In timed automata, there are two types of transitions: delay state transitions and discrete state transitions. Formally, for any states $s = (q, v), s' = (q', v') \in S(A)$, time delay $\Delta \in \mathbb{R}_{\geq 0}$ and event $\sigma \in \Sigma$,

- a delay state transition $(q, v) \xrightarrow{(\Delta, \varepsilon)} (q, v + \Delta)$ is defined if $v + \Delta \models \mathsf{inv}(q)$;
- a discrete state transition $(q, v) \xrightarrow{(0, \sigma)} (q', v')$ is defined if there is a transition $(q, g, \sigma, \mathcal{Y}, q') \in E$ such that $v \models g$, $v' = v_{[\mathcal{Y} \leftarrow 0]}$ and $v' \models \mathsf{inv}(q')$.

Intuitively, a delay state transition represents the dwell time in a discrete state for Δ time elapsing, while a discrete transition corresponds to a switch between two discrete states caused by occurrence of event σ. For simplicity, we write $s \xrightarrow{(\Delta, \sigma)} s'$ if there exists a state s'' such that $s \xrightarrow{(\Delta, \varepsilon)} s''$ and $s'' \xrightarrow{(0, \sigma)} s'$.

A *run* of time automaton A starting at state $s \in S(A)$ is a sequence

$$\pi = s_0(\Delta_0, \sigma_0)s_1(\Delta_1, \sigma_1)s_2(\Delta_2, \sigma_2)s_3 \cdots, \tag{1}$$

where $s_0 = s$, for any $i \geq 0$, $s_i \in S(A)$, $(\Delta_i, \sigma_i) \in \mathbb{R}_{\geq 0} \times (\Sigma \cup \{\varepsilon\})$ and $s_i \xrightarrow{(\Delta_i, \sigma_i)} s_{i+1}$ holds. For any run π, we denote by $\rho_\pi = (\Delta_0, \sigma_0)(\Delta_1, \sigma_1)(\Delta_2, \sigma_2)\cdots$ its corresponding timed word and by $s_\pi = s_0 s_1 s_2 \cdots$ its corresponding state sequence, which is also referred to as a *path*. A run π is said to be infinite if its $|\rho_\pi|$ is infinite; otherwise π is finite. We denote by $\mathsf{Run}^\omega(A)$ and $\mathsf{Run}^*(A)$, respectively, the set of infinite runs and finite runs in A starting at $(q_0, \mathbf{0}_{\mathcal{X}})$. The set of all runs in A is $\mathsf{Run}(A) = \mathsf{Run}^\omega(A) \cup \mathsf{Run}^*(A)$.

For any finite path $s = s_0 s_1 \cdots s_n$, we denote by $\mathsf{last}(s)$ the last state in the path. The set of timed words generated by A is $\mathsf{TW}(A) = \{\rho_\pi : \pi \in \mathsf{Run}(A)\}$; $\mathsf{TW}^\omega(A)$ and $\mathsf{TW}^*(A)$ denote, respectively, the sets of infinite and finite timed words generated by A. The untimed language of A is $\mathsf{utw}(\mathsf{TW}(A)) = \{\mathsf{utw}(\rho) : \rho \in \mathsf{TW}(A)\}$. The set of runs induced by a timed word $\rho \in \mathsf{TW}(A)$ is $\mathsf{Run}(\rho) = \{\pi \in \mathsf{Run}(A) : \rho_\pi = \rho\}$. With a slight abuse of notation, we define the set of last states induced by ρ is $\mathsf{last}(\rho) = \{(q, v) \in S(A) : \exists \pi \in \mathsf{Run}(\rho) \text{ s.t. } (q, v) = \mathsf{last}(s_\pi)\}$. For the sake of simplicity, we denote by $\mathsf{last}_d(\cdot) = \mathsf{dis}(\mathsf{last}(\cdot))$ the discrete states of $\mathsf{last}(\cdot)$, where "$\cdot$" can be a finite path or a finite timed word.

Given a TA A, an infinite run $\pi \in \mathsf{Run}^\omega(A)$ is said to be *non-zeno* if $\mathsf{time}(\rho_\pi) = \infty$; otherwise, it is zeno. A zeno run describes the phenomenon that infinite events are enabled in a finite time. A state $s = (q, v) \in S(A)$ is said to be a *timelock* if all infinite runs starting from s are zeno. For the sake of simplicity, we assume that the TA is *timelock-free* (Tripakis, 2002) in the sense that there is no timelock state reachable. This assumption holds for most of the engineering systems as a timelock state will prevent time progressing, which is not realistic in real-world systems. Furthermore, we will discuss how to relax this assumption later in Remark 3.

2.2. Region automata

For later technical developments, here we briefly review the region automata (Alur & Dill, 1994), which are widely used as finite abstractions of timed automata for the purpose of verification. The reader is referred to Alur and Dill (1994) and Baier and Katoen (2008) for more details.

Given a timed automaton $A = (Q, q_0, \Sigma, \mathcal{X}, \mathsf{inv}, E)$, each *region* of A is an equivalence class of time valuations; we denote the set of regions of A by $\mathcal{R}$ and use $[v]_{\mathcal{R}}$ to denote the unique region to which valuation v belongs. The *region automaton* of A is a 4-tuple

$$RG(A) = (Q^R, q_0^R, \Sigma^R, E^R),$$

where $Q^R = Q \times \mathcal{R}$ is the set of states, $q_0^R = (q_0, [\mathbf{0}_{\mathcal{X}}]_{\mathcal{R}})$ is the initial state, $\Sigma^R = \Sigma \cup \{\tau\}$ is set of events and $E^R : Q^R \times \Sigma^R \to 2^{Q^R}$ is the non-deterministic transition function defined by: for any $(q, r), (q', r') \in Q^R$, $\sigma \in \Sigma^R$, we have $(q', r') \in E^R((q, r), \sigma)$ if (i) $\sigma \in \Sigma$ and there is a transition $(q, g, \sigma, \mathcal{Y}, q') \in E$ such that $v \models g$ and $v_{[\mathcal{Y} \leftarrow 0]} \in r'$ for any $v \in r$; or (ii) $\sigma = \tau$, $q = q'$ and r' is a time successor region of r, which is obtained by time elapsing. Details about region abstraction and how transition function E^R is defined can be found in Alur and Dill (1994) and Baier and Katoen (2008). Function E^R is extended to $Q^R \times (\Sigma^R)^*$ in the usual way. The language generated by $RG(A)$ is $\mathcal{L}(RG(A)) = \{\rho \in (\Sigma^R)^* \cup (\Sigma^R)^\omega : E^R(q_0^R, \rho)!\}$, where ! means "is defined". A run in $RG(A)$ is a finite (or infinite) sequence $\pi = q_1^R \sigma_1 q_2^R \sigma_2 \cdots q_n^R(\cdots)$, where $q_i^R \in Q^R$, $\sigma_i \in \Sigma^R$ and $q_{i+1}^R \in E^R(q_i^R, \sigma_i)$, $i = 1, 2, \dots, n(, \dots)$. In order to more clearly represent transitions in a run $\pi = q_1^R \sigma_1 q_2^R \sigma_2 \cdots q_n^R(\cdots)$ of region automaton, we write it as $\pi = q_1^R \xrightarrow{\sigma_1} q_2^R \xrightarrow{\sigma_2} \cdots q_n^R(\cdots)$. A run can also be extended by events string that is $\pi = q_1^R \xrightarrow{\rho_1} q_2^R \xrightarrow{\rho_2} \cdots q_n^R(\cdots)$ where $\rho_i \in (\Sigma^R)^*$, $q_{i+1}^R \in E^R(q_i^R, \rho_i)$ and $i = 1, 2, \dots, n(, \dots)$.

Intuitively, event τ represents the time elapsing by abstracting the precise time. Although the region automaton abstracts the time information away from the original timed automaton, their untimed languages are equivalent. Formally, let $\mathsf{utw}^R(RG(A))$ be the untimed language of $RG(A)$ by erasing all events τ of each string in $\mathcal{L}(RG(A))$. Then, we have the following relation between A and $RG(A)$ (Alur & Dill, 1994):

$$\mathsf{utw}(\mathsf{TW}(A)) = \mathsf{utw}^R(RG(A)). \tag{2}$$

Based on the relation in Eq. (2), the region automata preserves reachability of discrete state in the original system A, that is, there exists a timed word ρ leading to a discrete state $q \in Q$, i.e., $q \in \mathsf{last}_d(\rho)$, if and only if there is a word ρ^R and a state $(q, r) \in Q^R$ in $RG(A)$ such that $(q, r) \in E^R(q_0^R, \rho^R)$.

3. Detectability for partially-observed timed automata

In the state estimation problem, we assume that not all events in Σ can be observed. To this end, we assume the event set is partitioned into two disjoint sets

$$\Sigma = \Sigma_o \cup \Sigma_{uo},$$

where Σ_o is the set of observable events and Σ_{uo} is the set of unobservable events. Furthermore, in the timed setting, we assume that time information can also be measured by, e.g., having a global timer. Therefore, we define the natural projection for timed word

$$P : \mathsf{TW}^*(\Sigma) \to \mathsf{TW}^*(\Sigma_o)$$

such that, for any timed word $\rho = (\Delta_0, \sigma_0)\ldots(\Delta_n, \sigma_n) \in \mathsf{TW}^*(\Sigma)$, the projection removes events in Σ_{uo} and keeps the time elapsing until the next observable event. Formally, P is defined recursively by:

- for $(\Delta, \sigma) \in \mathbb{R}_{\geq 0} \times (\Sigma \cup \{\varepsilon\})$, we have
$$P((\Delta, \sigma)) = \begin{cases} (\Delta, \sigma) & \text{if } \sigma \in \Sigma_o \\ (\Delta, \varepsilon) & \text{otherwise} \end{cases}$$
- for any $(\Delta, \sigma_0)(\Delta_1, \sigma_1)\rho \in \mathsf{TW}^*(\Sigma)$, we have

$$P((\Delta_0, \sigma_0)(\Delta_1, \sigma_1)\rho) = \begin{cases} (\Delta_0, \sigma_0)P((\Delta_1, \sigma_1)\rho) & \text{if } \sigma_0 \in \Sigma_o \\ P((\Delta_0 + \Delta_1, \sigma_1)\rho) & \text{otherwise} \end{cases}$$

For example, if $\Sigma_o = \{a, b\}$ and $\Sigma_{uo} = \{u\}$, then for timed word $\rho = (1, a)(2, u)(3, b)$, its natural projection is $P(\rho) = (1, a)(5, b)$. Note that, for any timed word $\rho \in \mathsf{TW}^*(\Sigma)$, we have $\mathsf{time}(\rho) = \mathsf{time}(P(\rho))$. For simplicity, we also extend natural projection to $P : 2^{\mathsf{TW}^*(\Sigma)} \to 2^{\mathsf{TW}^*(\Sigma_o)}$ in the usual manner.

Given a timed automaton A and suppose that a projected timed word $t \in P(\mathsf{TW}^*(A))$ is observed. Then the *current-state estimate* is defined as the set of *discrete states* the system can possibly reach after observing t, i.e.,

$$\mathsf{Reach}(t) = \left\{ q \in Q : \begin{array}{c} \exists \rho \in \mathsf{TW}^*(A) \text{ s.t.} \\ P(\rho) = t \wedge \mathsf{last}_d(\rho) = q \end{array} \right\}.$$

After observing timed word t, we may further observe timed word t'. Then, we can update the state estimate $\mathsf{Reach}(t)$ for observation t by the information from t'. The *delay state estimate* of the system at instant $\mathsf{time}(t)$ upon the occurrence of $tt' \in P(\mathsf{TW}^*(A))$ is defined as

$$\mathsf{Reach}_d(t, t') = \left\{ q \in Q : \begin{array}{c} \exists \rho_1\rho_2 \in \mathsf{TW}^*(A) \text{ s.t. } P(\rho_1) = t \wedge \\ P(\rho_2) = t' \wedge \mathsf{last}_d(\rho_1) = q \end{array} \right\}.$$

In the seminal work of Shu et al. (2007), strong detectability and weak detectability were proposed to capture different state estimation requirements. In particular, strong detectability is the stronger one requiring that the precise state of the system can always be determined after a finite number of observations, while weak detectability requires that the precise state can be determined for some observations. In some applications, we are interested in detecting the previous state with some information delays rather than the current state. To consider this delay state estimation problem, Shu and Lin proposed the notion of (k_1, k_2)-detectability (delayed detectability) in Shu and Lin (2012). Specifically, (k_1, k_2)-detectability (delayed detectability) requires that after obtaining k_1 observations, system state can always be determined with the assistance of another k_2 observation delays. However, the above original definitions by Shu and Lin are proposed for untimed finite-state automata without utilizing time information. Here, we extend these notions to the timed setting as follows.

Definition 1. Let $A = (Q, q_0, \Sigma, \mathcal{X}, \mathsf{inv}, E)$ be a timed DESs with observable events $\Sigma_o \subseteq \Sigma$. We say system A is

- *strongly detectable* if we can always determine the current and subsequent states of the system unambiguously after some finite time elapse, i.e.,
$$(\exists \Delta \in \mathbb{R}_{\geq 0})(\forall \pi \in \mathsf{Run}(A) : \mathsf{time}(\rho_\pi) = \infty)$$
$$(\forall \rho' \in \mathsf{Pre}(\rho_\pi))[\mathsf{time}(\rho') \geq \Delta \Rightarrow |\mathsf{Reach}(P(\rho'))| = 1].$$
- *weakly detectable* if we can determine the current and subsequent states of the system unambiguously for some timed words after some finite time elapse, i.e.,
$$(\exists \Delta \in \mathbb{R}_{\geq 0})(\exists \pi \in \mathsf{Run}(A) : \mathsf{time}(\rho_\pi) = \infty)$$
$$(\forall \rho' \in \mathsf{Pre}(\rho_\pi))[\mathsf{time}(\rho') \geq \Delta \Rightarrow |\mathsf{Reach}(P(\rho'))| = 1].$$

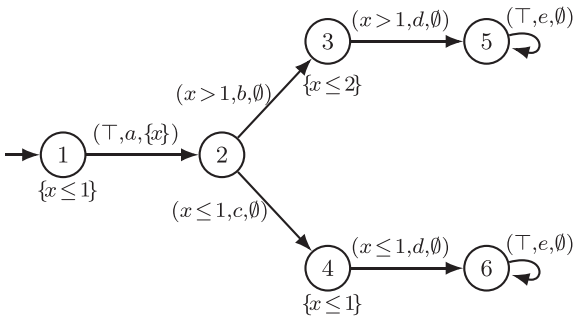
Definition 2. Let $A = (Q, q_0, \Sigma, \mathcal{X}, \mathsf{inv}, E)$ be a timed system with observable events $\Sigma_o \subseteq \Sigma$. We say system A is (k_1, k_2)-*detectable* if after k_1+k_2 time elapsing, we can determine the state of the system unambiguously k_2 time delays ago for all timed words, i.e.,

$$(\forall \rho \in \mathsf{TW}(A))(\forall \rho'\rho'' \in \mathsf{Pre}(\rho))[\mathsf{time}(\rho') \geq k_1$$
$$\wedge\, \mathsf{time}(\rho'') \geq k_2 \Rightarrow |\mathsf{Reach}_d(P(\rho'), P(\rho''))| = 1].$$

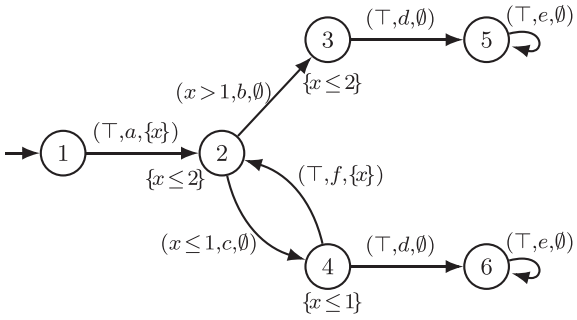
Remark 1. Here, we assume that parameters k_1 and k_2 in delayed detectability for timed systems are integers. This assumption does not reduce the generality when k_1 and k_2 are rational numbers. For example, let m be the least common denominator of k_1 and k_2. Then we can construct a new timed automaton A' by multiplying each clock constraint c in timed automaton A by m. Then we know that $\mathsf{utw}(\mathsf{TW}(A)) = \mathsf{utw}(\mathsf{TW}(A'))$. Therefore, verifying (k_1, k_2)-detectability for system A is equivalent to verifying$(k_1\ m, k_2\ m)$-detectability for system A'. However, if k_1 and k_2 are irrational numbers, then our approach does not apply directly; one may need to approximate the irrational numbers as rational numbers first.

We illustrate the definitions of detectability by the following examples.

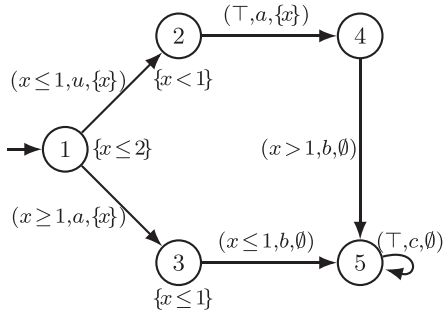
Example 1. Let us consider partially-observed timed automata A_1 shown in Fig. 1(a), where the alphabet is $\Sigma = \{a, b, c, d, e\}$ with $\Sigma_o = \{a, d, e\}$; the set of clock is $\mathcal{X} = \{x\}$; and invariant of each discrete state is the conjunction of all elements in the set next to the discrete state and is omitted if it is true. Note that in the untimed setting, this system is not (either strongly or weakly) detectable because time information is not utilized. This is because along the only possible observation $adeee\cdots$, we can never distinguish between states 5 and 6. However, with the assistance of time information, this system is strongly detectable (hence, also weakly detectable). Specifically, by the timed setting shown in Fig. 1(a), we argue that we can always determine the current state after three time units. To see this, first we note that the invariant of discrete state 1 is $x \leq 1$, which means that the system should depart from state 1 to 2 within one time unit. Since



(a) System A_1 with $\Sigma_o = \{a, d, e\}$ is strongly detectable.



(b) System A_2 with $\Sigma_o = \{a, d, e, f\}$ is weakly detectable.



(c) System A_3 with $\Sigma_o = \{a, b, c\}$ is not $(1, 2)$-detectable, but $(2, 1)$-detectable.

Fig. 1. Three timed DESs A_1, A_2 and A_3 in (a), (b) and (c) respectively. We use $\top$ to denote the true guard, which means that there is no time constraint and it allows arbitrary time elapses. The invariant of a discrete state is conjunction of all elements in the set next to the discrete state and we omit the invariant if it is true.

transition $2 \to 3$ has guard $x > 1$ but transition $2 \to 4$ can occur for any $0 \le x \le 1$, we know that the system may be either at state 2 or state 4 after observing the first event a. Therefore, we cannot distinguish states 2 and 4 within one time unit after observing the first event a without any further observation. Since transitions along $2 \to 3 \to 5$ require more than one but no more than two time units, while transitions along $2 \to 4 \to 6$ is feasible only within one time unit, there are three possibilities:

- If we observe event d within one time unit after observing a, then we know for sure that the system is at state 6;
- If we observe event d between one to two time units after observing a, then we know for sure that the system is at state 5;
- If we observe nothing within two time units after observing a, then we know for sure that the system will stay at 2 forever because the invariant of state 3 is $x \le 2$.

For each of the above three cases, after observing event a, it takes at most two time units to determine system state uniquely. Consider the largest time delay to reach state 2, i.e., one time unit, we can accurately detect the system state after three time units (or any time delays larger than two time units) in total. Thus, A_1 is strongly detectable.

Example 2. Let us consider timed system A_2 shown in Fig. 1(b) with $\Sigma_o = \{a, d, e, f\}$. For this system, there exists a run such that we cannot distinguish states 2 and 4 after any finite time of observations even by utilizing the time information. For example, for any $\Delta \in \mathbb{N}$, there is a run

$$\pi = 1(1, a)[2(0.5, c)4(0.5, f)]^{\Delta} \in \mathsf{Run}(A_2)$$

such that $\mathsf{time}(\rho_\pi) \ge \Delta$ and $\mathsf{Reach}(P(\rho_\pi)) = \{2, 4\}$. Thus, the system A_2 is not strongly detectable. However, we can find another run

$$\pi' = 1(1, a)2(1.5, b)3(0.2, d)[5(1, e)]^{\omega}$$

such that for any $\rho \in \mathsf{Pre}(\rho_{\pi'})$ satisfying $\mathsf{time}(\rho) \ge (\Delta_0 + 2.7)$ where $\Delta_0 > 0$, we can uniquely determine the state, i.e., $\mathsf{Reach}(P(\rho)) = \{5\}$. Thus, system A_2 is only weakly detectable.

Example 3. Let us consider timed system A_3 shown in Fig. 1(c) with $\Sigma_o = \{a, d, c\}$. For this system, let us consider the following two runs

$$\pi_1 = 1(1, u)2(0.4, a)4(1.1, b)5(3, c),$$
$$\pi_2 = 1(0.7, u)2(0.7, a)4(1.1, b)5(3, c),$$

and we have $\rho_1' = (1, \varepsilon) \in \mathsf{Pre}(\rho_{\pi_1})$, $\rho_1'' = (0, u)(0.4, a)(1.1, b)(3, c)$, $\rho_2' = (0.7, u)(0.3, \varepsilon) \in \mathsf{Pre}(\rho_{\pi_2})$ and $\rho_2'' = (0.4, a)(1.1, b)(3, c)$ such that $P(\rho_1') = P(\rho_2') = (1, \varepsilon)$, $P(\rho_1'') = P(\rho_2'') = (0.4, a)(1.1, b)(3, c)$, $\mathsf{time}(\rho_1') = \mathsf{time}(\rho_2') \ge 1$ and $\mathsf{time}(\rho_1'') = \mathsf{time}(\rho_2'') \ge 2$. Clearly, both states 1 and 2 can be reached after observing $(1, \varepsilon)$ and from these two states, observable timed word $(0.4, a)(1.1, b)(3, c)$ can be generated. Thus, system A_3 is not $(1, 2)$-detectable.

However, system A_3 is $(2, 1)$-detectable, since for any timed word $\rho \in \mathsf{TW}(A)$ and every prefix $\rho'\rho'' \in \mathsf{Pre}(\rho)$ such that $\mathsf{time}(\rho') \ge 2$ and $\mathsf{time}(\rho'') \ge 1$, we have $|\mathsf{Reach}_d(P(\rho'), P(\rho''))| = 1$. For example, let us consider run

$$\pi_3 = 1(1, u)2(0.9, a)4(1.1, b)5(3, c),$$

and prefixes $\rho_{\pi_3}' = (1, u)(0.9, a)(0.1, \varepsilon)$ and $\rho_{\pi_3}'' = (1, b)(3, c)$. States that can be reached after observing timed word $P(\rho_{\pi_3}') = (1.9, a)(0.1, \varepsilon)$ are 3 and 4. From state 3, event b is able to occur within one time unit, while from state 4, event b can only occur after one time unit. We have $\mathsf{Reach}_d(P(\rho_{\pi_3}'), P(\rho_{\pi_3}'')) = \{4\}$. Thus, although we observe the same event label, we can still use time information to distinguish state 3 and 4.

4. Verification of strong detectability

In this section, we investigate the verification of strong detectability. First, we construct a verification system that captures all pairs of runs with the same observation (both projected events and time elapsing). Then a necessary and sufficient condition for strong detectability is derived based on the region graph of the verification system, which yields the decidability of strong detectability.

4.1. Construction of the verification system

According to Definition 1, a system is not strongly detectable if for any arbitrarily long time elapsing, there exists a pair of two runs such that they have the same observation but result in different discrete states. Motivated by this observation, we construct a *verification system* that captures all pairs of runs with the same (timed) observation and can distinguish if a timed word has finite or infinite time elapsing. Given a timed automaton $A = (Q, q_0, \Sigma, \mathcal{X}, \text{inv}, E)$, the verification system of A is a new timed automaton

$$V(A) = (Q_V, q_{V0}, \Sigma_V, \mathcal{X}_V, \text{inv}_V, E_V),$$

where

- $Q_V = Q \times Q$ is the set of discrete states;
- $q_{V0} = (q_0, q_0)$ is the initial discrete state;
- $\Sigma_V = \Sigma \cup \{\lambda\}$ is a finite set of events, where $\lambda \notin \Sigma$ is a new event;
- $\mathcal{X}_V = \mathcal{X} \cup \hat{\mathcal{X}} \cup \{x_v\}$ is a finite set of clocks, where $\hat{\mathcal{X}} = \{\hat{x} : x \in \mathcal{X}\}$ is a copy of the original clock set $\mathcal{X}$ and x_v is a new clock;
- $\text{inv}_V : Q_V \to C(\mathcal{X}_V)$ is the invariant function defined by: for any $(q_1, q_2) \in Q_V$, $\text{inv}_V(q_1, q_2) = \text{inv}(q_1) \wedge \widehat{\text{inv}}(q_2) \wedge x_v \leq 1$, where $\widehat{\text{inv}}(q)$ simply replaces each $x \in \mathcal{X}$ in $\text{inv}(q)$ by $\hat{x} \in \hat{\mathcal{X}}$;
- $E_V \subseteq Q_V \times \Sigma_V \times C(\mathcal{X}_V) \times 2^{\mathcal{X}_V} \times Q_V$ is the transition relation defined by: for any $(q_1, q_2) \in Q_V$,

  - if $\sigma \in \Sigma_o$, then

$$\begin{aligned}&(q_1, \sigma, g_1, \mathcal{Y}_1, q_1'), (q_2, \sigma, g_2, \mathcal{Y}_2, q_2') \in E\\ \Rightarrow&((q_1, q_2), \sigma, g_1 \wedge \hat{g}_2, \mathcal{Y}_1 \cup \hat{\mathcal{Y}}_2, (q_1', q_2')) \in E_V \end{aligned} \tag{3}$$

  - if $\sigma \in \Sigma_{uo}$, then

$$\begin{aligned}&(q_1, \sigma, g_1, \mathcal{Y}_1, q_1') \in E \Rightarrow\\ &((q_1, q_2), \sigma, g_1, \mathcal{Y}_1, (q_1', q_2)) \in E_V,\\ &(q_2, \sigma, g_2, \mathcal{Y}_2, q_2') \in E \Rightarrow\\ &((q_1, q_2), \sigma, \hat{g}_2, \hat{\mathcal{Y}}_2, (q_1, q_2')) \in E_V \end{aligned} \tag{4}$$

  - if $\sigma = \lambda$, then

$$((q_1, q_2), \sigma, x_v = 1, \{x_v\}, (q_1, q_2)) \in E_V, \tag{5}$$

  where $\hat{g}_2$ and $\hat{\mathcal{Y}}_2$ are the copies of g_2 and $\mathcal{Y}_2$, respectively, to new clock set $\hat{X}$.

Intuitively, in the verification system $V(A)$, each discrete state is a pair of discrete states of the original system A. Since each discrete state in $V(A)$ corresponds to two discrete states in A, the clock set is the union of the original two clock sets, where we use a copy $\hat{\mathcal{X}}$ to distinguish them from $\mathcal{X}$. The invariant for each state is the conjunction of invariants of its two components in the state. In addition, we add a new event λ and a new clock x_v to measure the time elapsing explicitly. Specifically, we assign an additional invariant $x_v \leq 1$ to each discrete state, which means that clock x_v cannot be greater than 1 for each discrete state. The transition rule for event λ is specified by Eq. (5), which says that event λ can only occur whenever clock x_v equals to 1. Therefore, event λ will occur whenever one time unit elapses, which is used as an auxiliary time measurement in later verification algorithms. Also, for any state $(q_1, q_2) \in Q_V$, if $\sigma \in \Sigma_o$, then σ must be enabled *simultaneously* at q_1 and q_2 to ensure observational equivalence. On the other hand, if $\sigma \in \Sigma_{uo}$, then event σ can be enabled either at q_1 or q_2 while the other component remains unchanged.

Therefore, the construction of $V(A)$ guarantees that it (only) tracks all pairs of runs in V having the same observation. Formally, we have following properties (Tripakis, 2002):

- For any finite run π in $V(A)$,

$$\begin{aligned}\pi =&[(q_0, q_0'), v_0](\Delta_0, \sigma_0)[(q_1, q_1'), v_1](\Delta_1, \sigma_1)\\ &\cdots[(q_n, q_n'), v_n]\end{aligned}$$

  there exist two runs $\pi_1, \pi_2 \in \text{Run}(A)$ such that $\text{last}_d(s_{\pi_1}) = q_n$, $\text{last}_d(s_{\pi_2}) = q_n'$ and $P(\rho_\pi) = P(\rho_{\pi_1}) = P(\rho_{\pi_2})$;
- For any pair of finite runs π_1, π_2 in A with the same observation, there exists a finite run π in $V(A)$ having the same observation and the discrete part of the last state of π is $(\text{dis}(\text{last}(s_{\pi_1})), \text{dis}(\text{last}(s_{\pi_2})))$, i.e.,

$$\begin{aligned}&(\forall \pi_1, \pi_2 \in \text{Run}(A) : P(\rho_{\pi_1}) = P(\rho_{\pi_2}))\\ &(\exists \pi \in \text{Run}(V(A)))\big[(P(\rho_\pi) = P(\rho_{\pi_1}) = P(\rho_{\pi_2})\\ &\wedge \text{last}_d(s_\pi) = (\text{last}_d(s_{\pi_1}), \text{last}_d(s_{\pi_2})))\big].\end{aligned}$$

4.2. Verifying strong detectability

Recall that strong detectability requires that we can determine the current and subsequent state uniquely after finite time for all runs. To this end, we call a discrete state $(q_1, q_2) \in Q_V$ an *ambiguous state* if $q_1 \neq q_2$ and we denote by $AM = \{(q_1, q_2) \in Q_V : q_1 \neq q_2\}$ the set of all ambiguous states. By the properties of the verification systems, an ambiguous state is reached if there are two observationally equivalent runs in V reaching different discrete states. Furthermore, when an ambiguous state is reached, we cannot distinguish which state the system is actually in by the observation. Therefore, to test strong detectability, it suffices to test whether or not an ambiguous state in $V(A)$ can be reached by runs with arbitrarily large time elapsing. However, this cannot be tested directly based on $V(A)$ because it has infinite reachable states in general. Our approach is to consider the region automaton of $V(A)$ denoted by $V^R(A) = (Q^R, q_0^R, \Sigma^R, E^R)$. Similarly, we define the set of ambiguous states in $V^R(A)$ as $AM^R = \{(q, r) \in Q^R : q \in AM\}$.

The following theorem shows that the region automaton $V^R(A)$ abstracts sufficient information of $V(A)$ for verifying strong detectability.

Theorem 1. *System A is not strongly detectable with respect to Σ_o, if and only if, in the region automaton $V^R(A)$ of $V(A)$, there exists a run*

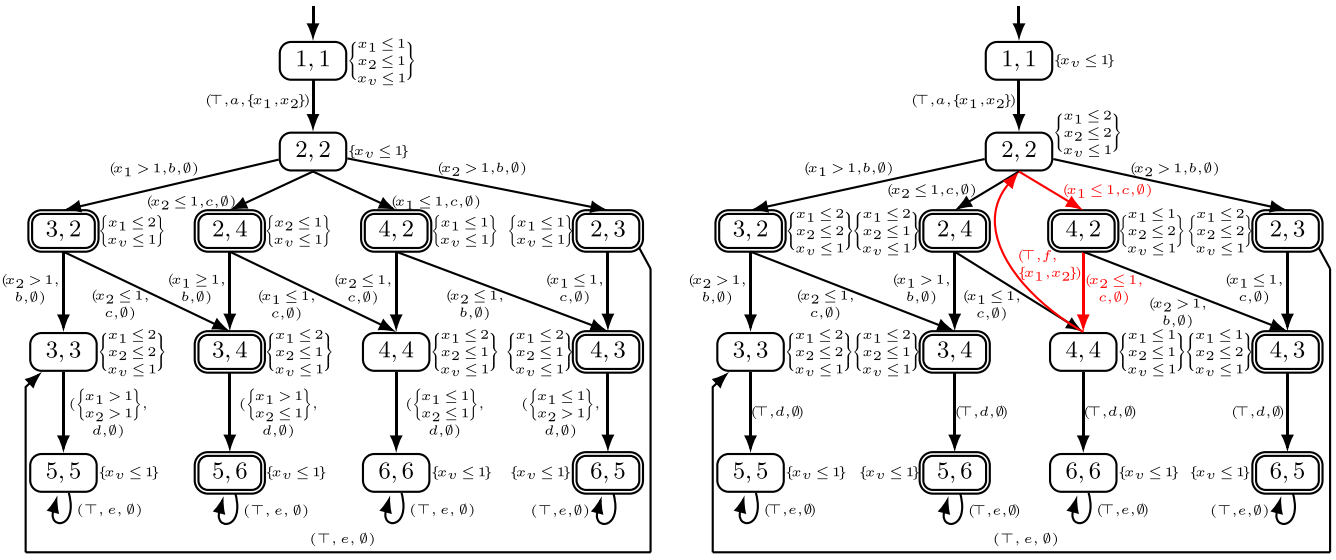
$$\pi = q_0^R \xrightarrow{\sigma_1} q_1^R \xrightarrow{\sigma_2} \cdots \xrightarrow{\sigma_i} q_i^R \cdots \xrightarrow{\sigma_j} q_j^R \cdots \xrightarrow{\sigma_n} q_n^R$$

where $i < j$ such that

(i) $q_n^R \in AM^R$; and
(ii) $q_i^R = q_j^R$; and
(iii) $\exists i < k \leq j : \sigma_k = \lambda$.

Proof. ($\Leftarrow$) Assume there exists a run π' in $V^R(A)$ satisfying conditions (i)–(iii). For any $\Delta \in \mathbb{R}_{\geq 0}$, we use $\lceil \Delta \rceil$ to denote the smallest integer greater than Δ. Thus, we have $\lceil \Delta \rceil \geq \Delta$. Based on condition (ii), we can obtain a word $\rho^R = \sigma_1\sigma_2 \cdots \sigma_{i-1}(\sigma_i \cdots \sigma_j)^{\lceil \Delta \rceil} \sigma_{j+1} \cdots \sigma_n$ by repeating transitions from q_i^R to q_j^R in π' for $\lceil \Delta \rceil$ times. By condition (iii), the number of event λ in ρ^R is at least $\lceil \Delta \rceil$. Because the last state of π' is included in AM^R, we have $E^R(q_0^R, \rho^R) \cap AM^R \neq \emptyset$. By Eq. (2), there exists a run π in $V(A)$ such that ρ_π has at least $\lceil \Delta \rceil$ number of event λ and the last state of π is ambiguous, i.e., $\text{time}(\rho_\pi) \geq \lceil \Delta \rceil$ and $\text{last}_d(s_\pi) \in AM$. By the first property of the verification system, there are two runs $\pi_1, \pi_2 \in \text{Run}(A)$ such that $\text{last}_d(s_{\pi_1}) \neq \text{last}_d(s_{\pi_2})$ and $P(\rho_{\pi_1}) = P(\rho_{\pi_2})$. We cannot determine the state by observation $P(\rho_{\pi_1})$, i.e., $|\text{Reach}(P(\rho_{\pi_1}))| \neq 1$. Therefore, A is not strongly detectable.

($\Rightarrow$) Assume A is not strongly detectable. That is, for any $\Delta \in \mathbb{R}_{\geq 0}$, there are two runs $\pi_1, \pi_2 \in \text{Run}(A)$ such that $P(\rho_{\pi_1}) =$



(a) Verification system $V(A_1)$ of timed system A_1 (b) Verification system $V(A_2)$ of timed system A_2

Fig. 2. In the above figures, double circles denote ambiguous states. We omit transition $(x_v = 1, \lambda, \{x_v\})$ at each discrete state. The invariant of a discrete state is conjunction of all elements in the set next to the discrete state, e.g., $\{x_1 \leq 2, x_2 \leq 1, x_v \leq 1\}$ represents $x_1 \leq 2 \wedge x_2 \leq 1 \wedge x_v \leq 1$, and we omit the invariant if it is true. For each guard, the abbreviation of true is denoted by $\top$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$P(\rho_{\pi_2})$, $\mathsf{last}_d(s_{\pi_1}) \neq \mathsf{last}_d(s_{\pi_2})$ and $\mathsf{time}(\rho_{\pi_1}) = \mathsf{time}(\rho_{\pi_1}) \geq \Delta$. According to the second property of verification system, there exists a run π in $V(A)$ such that $\mathsf{last}_d(s_\pi) \in AM$ and $P(\rho_\pi) = P(\rho_{\pi_1}) = P(\rho_{\pi_2})$. As a result of $\mathsf{time}(\rho_\pi) \geq \Delta$, ρ_π has at least $\lceil \Delta \rceil$ number of event λ. Then, we obtain the region automaton $V^R(A) = (Q^R, q_0^R, \Sigma^R, E^R)$. By Eq. (2), there is a run π' in $V^R(A)$ whose last state is in AM^R and π' has at least $\lceil \Delta \rceil$ number of event λ. Let $\Delta > |Q^R|$. Because $V^R(A)$ is finite, run π' must firstly go through a run $q_i^R \xrightarrow{\sigma_{i+1}} \cdots \xrightarrow{\sigma_j} q_j^R$ in $V^R(A)$ which forms a loop, i.e., $q_i^R = q_j^R$, and has at least one λ in it, i.e., $\exists i < k \leq j : \sigma_k = \lambda$. Namely, there exists a run $q_0^R \xrightarrow{\sigma_1} q_1^R \xrightarrow{\sigma_2} \cdots \xrightarrow{\sigma_i} q_i^R \cdots \xrightarrow{\sigma_j} q_j^R \cdots \xrightarrow{\sigma_n} q_n^R$ in $V^R(A)$ satisfying conditions (i)–(iii). □

Intuitively, run $\pi = q_0^R \xrightarrow{\sigma_1} q_1^R \xrightarrow{\sigma_2} \cdots \xrightarrow{\sigma_i} q_i^R \cdots \xrightarrow{\sigma_j} q_j^R \cdots \xrightarrow{\sigma_n} q_n^R$ in Theorem 1 is equivalent to the existence of a reachable cycle π' from the initial state, i.e., the part $\pi' = q_i^R \cdots \xrightarrow{\sigma_j} q_j^R$, in which there exists at least one event λ and we can reach an ambiguous state from the cycle, i.e., from state q_i^R to state q_n^R. Then cycle π' and event λ in it can be extended to any time elapsing, and the reachability from q_j^R to an ambiguous state q_n^R prevents us from determining states unambiguously. Therefore, that system A is not strongly detectable means that we can find a run satisfying conditions (i)–(iii) in Theorem 1.

Remark 2. Let us discuss the complexity of checking strong detectability for a timed system $A = (Q, q_0, \Sigma, \mathcal{X}, \mathsf{inv}, E)$. For any clock $x \in \mathcal{X}$, we use c_x to denote the maximum integer c such that $x \sim c \in C(\mathcal{X})$ is a subformula appearing in A, where $\sim \in \{\leq, <, \geq, >, =\}$. The verification system $V(A)$ has at most $|Q|^2$ states and $|Q|^2(|Q|^2 - 1)$ transitions, where $|Q|$ is the number of states in A. Since its clocks consist of two copies of the original clocks and a new clock, the number of clocks in $V(A)$ is $|\mathcal{X}_V| = 2|\mathcal{X}| + 1$, where $|\mathcal{X}|$ is the clock number of A. By Alur and Dill (1994), the number of regions is bounded by $|\mathcal{X}_V|! \cdot 2^{|\mathcal{X}_V|} \cdot \prod_{x \in \mathcal{X}_V}(2c_x + 2)$. Thus, there are at most $(|Q|^2 \cdot (|Q|^2 - 1) \cdot |\mathcal{X}_V|! \cdot 2^{|\mathcal{X}_V|} \cdot \prod_{x \in \mathcal{X}_V}(2c_x + 2))$ states in $V^R(A)$ and we can construct the region automaton $V^R(A)$ within time $O(|Q|^4 \cdot |\mathcal{X}_V|! \cdot 2^{|\mathcal{X}_V|} \cdot \prod_{x \in \mathcal{X}_V}(2c_x + 2))$. According to Theorem 1, verifying strong detectability requires to find all cycles that contain event λ in $V^R(A)$ and then, to check reachability from these cycles to ambiguous states. Both of above steps can be done in time polynomial in the number of states in $V^R(A)$. Therefore, the whole complexity mainly relies on the size of the region automaton $V^R(A)$.

Remark 3. Recall that, in Section 2.1, we have assumed that the TA is timelock-free. In fact, with some slight modification, Theorem 1 can still apply when this assumption does not hold. Specifically, suppose that TA A contains a timelock state. Then it is not strongly detectable, if and only if, in the region automaton $V^R(A)$ of $V(A)$, there exists a run

$$\pi = q_0^R \xrightarrow{\sigma_1} q_1^R \xrightarrow{\sigma_2} \cdots \xrightarrow{\sigma_i} q_i^R \cdots \xrightarrow{\sigma_j} q_j^R \cdots \xrightarrow{\sigma_n} q_n^R \xrightarrow{\sigma_{n+1}} \cdots \xrightarrow{\sigma_m} q_m^R \xrightarrow{\sigma_{m+1}} \cdots \xrightarrow{\sigma_{m'}} q_{m'}^R$$

such that, in addition to the satisfactions of conditions (i)–(iii) in Theorem 1, we further require that $q_m^R = q_{m'}^R$ and there exists $m < k' \leq m'$ such that $\sigma_{k'} = \lambda$. Intuitively, the new requirements say that the region automaton $V^R(A)$ can reach a cycle $\pi'' = q_m^R \xrightarrow{\sigma_{m+1}} \cdots \xrightarrow{\sigma_{m'}} q_{m'}^R$ in which there is at least one event λ from state q_n^R. Since cycle π'' can be extended to any time elapsing by repeating it, it excludes runs that terminate at timelock states.

Example 4. Let us consider timed system A_1 shown in Fig. 1(a) with observable event set $\Sigma_o = \{a, d, e\}$. We obtain the verification system $V(A_1)$ by the aforementioned steps, which is depicted in Fig. 2(a) and for simplicity, we omit the transition $(x_v = 1, \lambda, \{x_v\})$ at each discrete state. One can compute the region automaton $V^R(A_1)$ for $V(A_1)$, in which there does not exist a run satisfying the conditions in Theorem 1. Thus, A_1 is strongly detectable.

Example 5. However, system A_2 shown in Fig. 1(b), where $\Sigma_o = \{a, d, e, f\}$, is not strongly detectable. To see this, first, we obtain the verification system $V(A_2)$ shown in Fig. 2(b), and based on which we construct the region automaton $V^R(A_2)$ of $V(A_2)$. Part

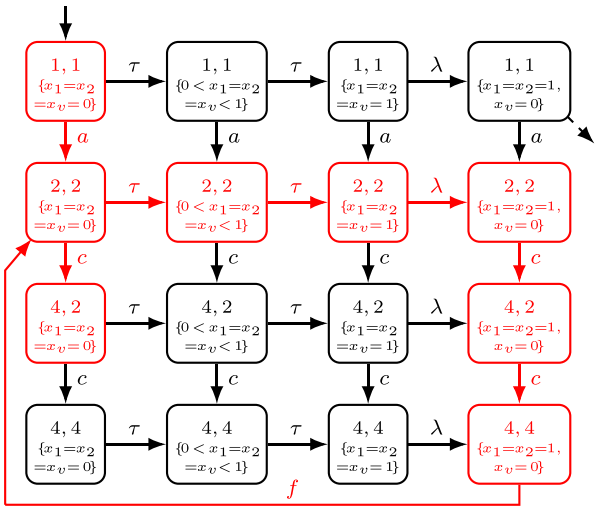


Fig. 3. Part of the region automaton $V^R(A_2)$. The cycle highlighted in red contains event λ and an ambiguous state $((4,2),\{x_1 = x_2 = 1, x_v = 0\})$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of the region automaton $V^R(A_2)$ is shown in Fig. 3. Here, we can find the following run

$$\begin{aligned}\pi =&((1,1),\{x_1 = x_2 = x_v = 0\}) \xrightarrow{a}\\ &((2,2),\{x_1 = x_2 = x_v = 0\}) \xrightarrow{\tau}\\ &((2,2),\{0 < x_1 = x_2 = x_v < 1\}) \xrightarrow{\tau}\\ &((2,2),\{x_1 = x_2 = x_v = 1\}) \xrightarrow{\lambda}\\ &((2,2),\{x_1 = x_2 = 1, x_v = 0\}) \xrightarrow{c}\\ &((4,2),\{x_1 = x_2 = 1, x_v = 0\}) \xrightarrow{c}\\ &((4,4),\{x_1 = x_2 = 1, x_v = 0\}) \xrightarrow{f}\\ &((2,2),\{x_1 = x_2 = x_v = 0\}) \xrightarrow{c}\\ &((4,2),\{x_1 = x_2 = x_v = 0\}),\end{aligned}$$

such that $q_7^R = ((4,2),\{x_1 = x_2 = x_v = 0\}) \in AM^R$. Furthermore, run π contains a cycle part as highlighted in Fig. 3, where $q_1^R = q_6^R = ((2,2),\{x_1 = x_2 = x_v = 0\})$ and $\sigma_4 = \lambda$. Thus, run π satisfies all conditions in Theorem 1, which means that A_2 is not strongly detectable. In fact, the cycle found in the above run π corresponds to a cycle in the verification structure $V(A_2)$ as highlighted by red color in Fig. 2(b). Based on the cycle, we can actually extract a timed word $\rho = (0,a)[(1,c)(0,f)]^n$ with time elapsing $\mathsf{time}(\rho) = n$ where n is an arbitrary integer in $\mathbb{N}$. Because event c is unobservable, we can never distinguish between system states 2 and 4.

Remark 4. The basic idea of the verification system $V(A)$ is motivated by the construction for the verification of diagnosability in untimed DESs (Jiang, Huang, Chandra, & Kumar, 2001; Yoo & Lafortune, 2002) and timed DESs (Cassandras & Lafortune, 2021; Tripakis, 2002), where it is termed as the twin-plant or the verifier. Our construction of the verification system itself is also similar to the *parallel composition* of timed automata with guards in Cassandras and Lafortune (2021) and Tripakis (2002). In particular, the new auxiliary event λ and self-loops can also be added to each state after constructing the parallel composition system. However, the necessary and sufficient condition derived is quite different. In particular, in diagnosability analysis, one needs to test whether or not the time is divergent after some faulty events. This condition can be formulated as the Büchi emptiness condition based on the verification system directly. However, in the context of strong detectability, there is no faulty indicator from which one needs to count the time. Instead, here we need to test if an ambiguous state can be reached following an arbitrarily long prefix. This condition cannot be captured directly by standard Büchi condition in the verification system, which motivates the use of region automata to test the conditions.

Remark 5. In fact, the verification condition for strong detectability in Theorem 1 can be captured by a computation tree logic (CTL) model checking problem (Emerson & Clarke, 1982) over the region automaton of the verification system. Specifically, starting from an arbitrary state, both the existence of a path to an ambiguous state and the existence of a cycle containing λ can be captured by CTL formulae. Then one just needs to check the reachability of such state with the above two properties, which again can be captured by CTL formula. Therefore, our condition in Theorem 1 can be implemented by existing tools as follows. First build the region automaton of the verification system. Then, perform a CTL model checking for the above described formula over the region automaton.

5. Undecidability of weak detectability

In this section, we investigate the verification of weak detectability for timed systems. Unfortunately, we prove that weak detectability is undecidable by reducing the universality problem, which is known to be undecidable for timed automata, to the verification of weak detectability.

Given a timed automaton A, the universality problem asks whether or not all strings in $\mathsf{TW}(\Sigma)$ can be generated by A, i.e., decide whether or not we have

$$\mathsf{TW}(A) = \mathsf{TW}(\Sigma).$$

In Alur and Dill (1994), Alur and Dill showed that the universality problem is undecidable for timed automata. We will use this result to show the undecidability of weak detectability for TA.

Given a TA $A = (Q, q_0, \Sigma, \mathcal{X}, \mathsf{inv}, E)$, we construct a new TA

$$G = (Q_G, q_{ini}, \Sigma_G, \mathcal{X}, \mathsf{inv}_G, E_G),$$

where

- $Q_G = Q \dot\cup \{q_{ini}, q_B\}$, where q_{int} and q_B are two new discrete states;
- $q_{ini} \in Q_G$ is the initial discrete state;
- $\Sigma_G = \Sigma \dot\cup \{\sigma_o\}$, where $\sigma_o \notin \Sigma$ is a new event;
- $\mathcal{X}$ is the clock set the same as A;
- inv_G is the same as inv for states in Q and for states q_{ini} and q_B, invariants are defined by: $\mathsf{inv}_G(q_{ini})$ is $x \le 1$, where $x \in \mathcal{X}$ is an arbitrary clock in system A, and $\mathsf{inv}_G(q_B)$ is true;
- E_G is the set of transitions defined by

$$\begin{aligned}&E_G = E \cup \{(q_B, \sigma, true, \mathcal{X}, q_B) : \sigma \in \Sigma\} \cup\\ &\{(q_{ini}, \sigma_o, x = 1, \mathcal{X}, q_0), (q_{ini}, \sigma_o, x = 1, \mathcal{X}, q_B)\} \qquad (6)\end{aligned}$$

The construction of G is conceptually illustrated in Fig. 4. Intuitively, G starts from a new initial state q_{ini} and non-deterministically goes to either the initial state of the original system A, i.e., q_0, or a new state q_B, via the same event σ_o. Here, we use a new auxiliary event $\sigma_o \notin \Sigma$ as the initial prefix that matches the initial conditions of strings in A and strings in $\mathsf{TW}(\Sigma)$. Specifically, we assign invariant $x \le 1$ to the initial state q_{ini}, where clock x is an arbitrary clock in $\mathcal{X}$. Thus, G can only stay at state q_{ini} for at most 1 time unit. As described in Eq. (6), there are only two transitions from q_{ini}, i.e., $(q_{ini}, \sigma_o, x = 1, \mathcal{X}, q_0)$ and

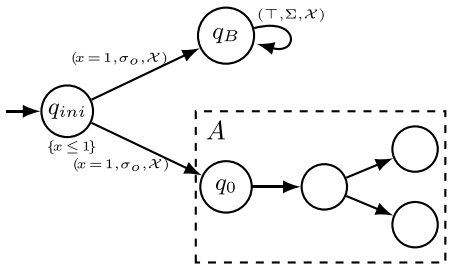


Fig. 4. Illustration of the construction of G. Edge $(\top,\Sigma,\mathcal{X})$ attached to q_B represents the set of all edges in $\{(\top,\sigma,\mathcal{X}):\sigma\in\Sigma\}$.

$(q_{ini},\sigma_o,x=1,\mathcal{X},q_B)$, which have guard $x=1$ and reset all clocks in $\mathcal{X}$. As a result, only $(1,\sigma_o)$ can occur from initial state q_{ini} and the valuation of all clocks are the same after the occurrence of σ_o. Then, from state q_0, G will follow exactly the same dynamic of A. On the other hand, from state q_B, all events in Σ can occur freely, which actually corresponds to a TA satisfying universality requirement.

Now, we make the following observations from the construction of the new system G. First, we note that, starting from state q_B, all timed words $\rho\in\mathsf{TW}(\Sigma)$ can be generated. Therefore, for any timed word $(1,\sigma_o)\rho$, it may end up with discrete state q_B. Similarly, if A is universal, then timed word $(1,\sigma_o)\rho$ may also end up with a state in Q. By assuming that all events in G are observable, then upon the occurrence of $(1,\sigma_o)\rho$, we cannot distinguish between state q_B from some state in Q. On the other hand, if A is not universal, then there exists a timed word $(1,\sigma_o)\rho$ such that ρ is not feasible from q_0 but is feasible from q_B. Since we assume all events are observable, we can determine for sure that the system is at q_B upon the occurrence of $(1,\sigma_o)\rho$. Furthermore, we know the state forever since q_B only has self-loops, which means that G is weakly detectable. The above observations lead to the following main theorem.

Theorem 2. *Weak detectability is undecidable for timed automata.*

Proof. It suffices to show that the universality problem for TA, which is undecidable, can be reduced to an instance of the weak detectability verification problem for TA. Specifically, given a timed automaton $A=(Q,q_0,\Sigma,\mathcal{X},\mathsf{inv},E)$ for which we want to decide whether or not $\mathsf{TW}(A)=\mathsf{TW}(\Sigma)$, we construct TA G by steps aforementioned. Next, we show that $\mathsf{TW}(A)=\mathsf{TW}(\Sigma)$ iff G is not weakly detectable with respect to Σ_G.

Suppose that G is weak detectable. Then we know that there exists a time elapsing $\Delta\in\mathbb{R}_{\geq 0}$ and an infinite run $\pi\in\mathsf{Run}^\omega(G)$ such that

$$(\forall\rho\in\mathsf{Pre}(\rho_\pi):\mathsf{time}(\rho)\geq\Delta)[|\mathsf{Reach}(P(\rho))|=1].$$

Since we assume all events in G are observable, we have $\mathsf{Reach}(P(\rho))=\mathsf{last}_d(\rho)$. Therefore, there exists $\rho=(1,\sigma_o)\rho'\in\mathsf{TW}(G)$ such that $|\mathsf{last}_d((1,\sigma_o)\rho')|=1$ for some continuation ρ'. Since $q_B\in\mathsf{last}_d((1,\sigma_o)\rho')$ and $q_B\notin Q$, we know that $\mathsf{last}_d((1,\sigma_o)\rho')\cap Q=\emptyset$. This means that $\rho'\notin\mathsf{TW}(A)$, i.e., A is not universal.

Similarly, if A is not universal, we can find $\rho\notin\mathsf{TW}(A)$. This means $\mathsf{last}_d((1,\sigma_o)\rho)=\{q_B\}$. Furthermore, once the system reaches q_B, it stays forever. Therefore, for any event $\sigma\in\Sigma$, we have $\mathsf{last}_d((1,\sigma_o)\rho(1,\sigma)^n)=\{q_B\}$. Therefore, there exists a time elapsing $\Delta=1$ and an infinite run $\pi=q_{ini}(1,\sigma_o)[q_B(1,\sigma)]^\omega\in\mathsf{Run}^\omega(G)$ such that

$$(\forall\rho\in\mathsf{Pre}(\rho_\pi):\mathsf{time}(\rho)\geq\Delta)[|\mathsf{Reach}(P(\rho))|=|\mathsf{last}_d(\rho)|=1].$$

This completes the proof. □

6. Verifying delayed detectability

Note that strong detectability and weak detectability are both concerned with the current-state estimate of the system. Delayed detectability, however, is concerned with the delay state estimate. In this section, we show that delayed detectability can also be effectively verified based on the verification system proposed in Section 4.1.

Compared with strong detectability, delayed detectability requires that even if we cannot distinguish two current states after k_1 time elapsing, this disambiguation should be resolved after another k_2 elapsing from the current instant. This implies that non-existence of two observational equivalent strings with k_2 elapsing from these two indistinguishable states. This observation leads to the following necessary and sufficient condition characterizing delayed detectability.

Theorem 3. *A timed system A is not (k_1,k_2)-detectable with respect to Σ_o, if and only if, in the region automaton $V^R(A)$ of the verification system $V(A)$, there exists a run*

$$\pi=q_0^R\xrightarrow{\rho_0^R}\cdots\xrightarrow{\rho_{k_1-1}^R}q_{k_1}^R\xrightarrow{\rho_{k_1}^R}\cdots\xrightarrow{\rho_{k_1+k_2-1}^R}q_{k_1+k_2}^R,$$

where $\forall 0\leq i\leq k_1+k_2: q_i^R\in Q_R\wedge\rho_i^R\in(\Sigma^R)^*$ such that

(i) $\forall 0\leq i\leq k_1+k_2-1:\lambda\in\rho_i^R$; and
(ii) $q_{k_1}^R\in AM^R$.

Proof. ($\Rightarrow$) Assume that timed system A is not (k_1,k_2)-detectable. According to Definition 2, we know that the following equation holds

$$(\exists\rho\in\mathsf{TW}(A))(\exists\rho'\rho''\in\mathsf{Pre}(\rho))[\mathsf{time}(\rho')\geq k_1\wedge\mathsf{time}(\rho'')\geq k_2\wedge|\mathsf{Reach}(P(\rho'),P(\rho''))|\neq 1]. \tag{7}$$

Thus, there is a run π_1 such that word ρ_{π_1} has a prefix $\rho'\rho''$ satisfying $\mathsf{time}(\rho')\geq k_1$, $\mathsf{time}(\rho'')\geq k_2$ and $|\mathsf{Reach}(P(\rho'),P(\rho''))|\neq 1$. Corresponding to $\rho'\rho''$, we can extract the run $\pi_1'\pi_1''$ from π_1 such that $\rho_{\pi_1'}=\rho'$ and $\rho_{\pi_1''}=\rho''$. By $|\mathsf{Reach}(P(\rho'),P(\rho''))|\neq 1$, there exists another run $\pi_2'\pi_2''$ such that $P(\rho_{\pi_2'})=P(\rho')$, $P(\rho_{\pi_2''})=P(\rho'')$ and $\mathsf{last}_d(\rho_{\pi_2'})\neq\mathsf{last}_d(\rho_{\pi_1'})$. Thus, by the second property of the verification system, there is a run $\pi_V'\pi_V''\in\mathsf{Run}(V(A))$ such that $P(\rho_{\pi_V'})=P(\rho'),P(\rho_{\pi_V''})=P(\rho'')$ and $\mathsf{last}_d(s_{\pi_V'})=(\mathsf{last}_d(s_{\pi_1'}),\mathsf{last}_d(s_{\pi_2'}))$. Because $\mathsf{time}(\rho_{\pi_V'})=\mathsf{time}(\rho')\geq k_1$ and $\mathsf{time}(\rho_{\pi_V''})=\mathsf{time}(\rho'')\geq k_2$, event λ occurs for at least k_1 times in $\rho_{\pi_V'}$ and for at least k_2 times in $\rho_{\pi_V''}$. In the region automaton $V^R(A)$, by Eq. (2), there exists a run $\pi=q_0^R\xrightarrow{\rho_0^R}\cdots\xrightarrow{\rho_{k_1-1}^R}q_{k_1}^R\xrightarrow{\rho_{k_1}^R}\cdots\xrightarrow{\rho_{k_1+k_2-1}^R}q_{k_1+k_2}^R$ such that each ρ_i^R, where $i\in\{0,\ldots,k_1+k_2-1\}$, contains at least one event λ and the k_1th state $q_{k_1}^R=((q_1,q_2),r)$ satisfies $q_1=\mathsf{last}_d(s_{\pi_1'})$ and $q_2=\mathsf{last}_d(s_{\pi_2'})$. That is, there exists a run satisfying conditions (i) and (ii).

($\Leftarrow$) Assume that there is a run π satisfying conditions (i) and (ii) in region automaton $V^R(A)$. Based on condition (i), we know that the number of event λ in word $\rho_R'=\rho_0^R\ldots\rho_{k_1}^R$ is at least k_1 and in word $\rho_R''=\rho_{k_1}^R\ldots\rho_{k_1+k_2-1}^R$ is at least k_2. By Eq. (2), there exists a run $\pi_V'\pi_V''$ in $V(A)$ such that $\mathsf{utw}(\rho_{\pi_V'})=\mathsf{utw}^R(\rho_R')$, $\mathsf{utw}(\rho_{\pi_V''})=\mathsf{utw}^R(\rho_R'')$. By the construction of region automata, the discrete component of last state of $s_{\pi_V'}$ is ambiguous, $\rho_{\pi_V'}$ has at least k_1 number of λ events and $\rho_{\pi_V''}$ has at least k_2 λ events, i.e. $\mathsf{last}_d(s_{\pi_V'})\in AM$, $\mathsf{time}(\rho_{\pi_V'})\geq k_1$ and $\mathsf{time}(\rho_{\pi_V''})\geq k_2$. By the first property of the verification system, there exist two

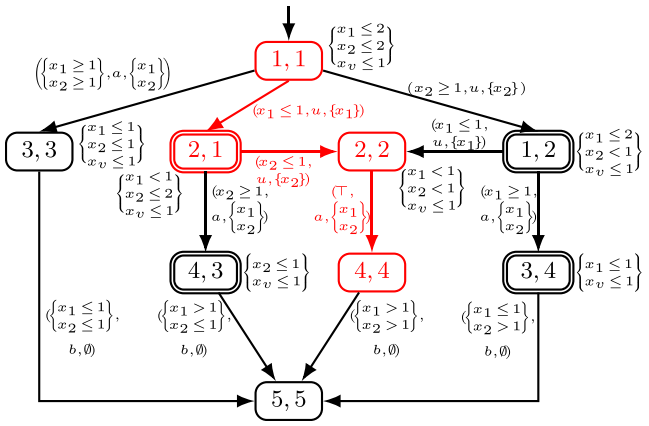


Fig. 5. Verification system $V(A_3)$ of timed system A_3. We omit transition $(x_v = 1, \lambda, \{x_v\})$ at each discrete state. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

runs $\pi_1'\pi_1'', \pi_2'\pi_2'' \in \mathsf{Run}(A)$ such that (i) for two runs π_1' and π_2', $P(\rho_{\pi_1'}) = P(\rho_{\pi_2'}) = P(\rho_{\pi_V'})$, $\mathsf{time}(\rho_{\pi_1'}) = \mathsf{time}(\rho_{\pi_2'}) \geq k_1$ and $\mathsf{last}_d(s_{\pi_1'}) \neq \mathsf{last}_d(s_{\pi_2'})$; (ii) for two runs π_1'' and π_2'', we have $P(\rho_{\pi_1''}) = P(\rho_{\pi_2''}) = P(\rho_{\pi_V''})$ and $\mathsf{time}(\rho_{\pi_1''}) = \mathsf{time}(\rho_{\pi_2''}) \geq k_2$. Thus, for the timed word $\rho_{\pi_1'}\rho_{\pi_1''}$, when k_1 time units elapse, we cannot determine the discrete state even considering k_2 time units delayed information, because $|\mathsf{Reach}(P(\rho_{\pi_1'}), P(\rho_{\pi_1''}))| \neq 1$. Thus, the system is not (k_1, k_2)-detectable. □

Intuitively, run $\pi = q_0^R \xrightarrow{\rho_0^R} \cdots \xrightarrow{\rho_{k_1-1}^R} q_{k_1}^R \xrightarrow{\rho_{k_1}^R} \cdots \xrightarrow{\rho_{k_1+k_2-1}^R} q_{k_1+k_2}^R$ in Theorem 3 reveals the existence of an ambiguous state $q_{k_1}^R$ which can be reached after at least k_1 number of λ event and from which the run can still execute event λ for at least k_2 times. Ambiguous state $q_{k_1}^R$ means that we cannot determine the current state after k_1 time elapsing. Furthermore, the part $q_{k_1}^R \xrightarrow{\rho_{k_1}^R} \cdots \xrightarrow{\rho_{k_1+k_2-1}^R} q_{k_1+k_2}^R$ means that from two different discrete components in q_{k_1}, there still exist two observation-equivalent executions having more than k_2 time elapsing, that is, we cannot distinguish the ambiguous discrete components in $q_{k_1}^R$ even with another k_2 time units' information. Therefore. the existence of a run satisfying conditions (i)–(ii) in Theorem 3 means that the system is not (k_1, k_2)-detectable.

Remark 6. Let us discuss the complexity of checking (k_1, k_2)-detectability for a timed system $A = (Q, q_0, \Sigma, \mathcal{X}, \mathsf{inv}, E)$. Similar in Remark 2, for any clock $x \in \mathcal{X}$, we use c_x to denote the maximum integer c such that $x \sim c \in C(\mathcal{X})$ is a subformula appearing in A. As mentioned in Remark 2, the complexity of $V^R(A)$ is $O(|Q|^4 \cdot |\mathcal{X}_V|! \cdot 2^{|\mathcal{X}_V|} \cdot \prod_{x \in \mathcal{X}_V}(2c_x + 2))$. To check (k_1, k_2)-detectability for system A by Theorem 3, we construct $V^R(A) = (Q^R, q_0^R, \Sigma^R, E^R)$ and compute the state set list $Q_0, \ldots, Q_{k_1}, \ldots, Q_{k_1+k_2}$ where $Q_0 = \{q_0^R\}$, $Q_{k_1} \subseteq AM^R$ and for $i \in \{0, \ldots, k_1 + k_2 - 1\}$, $Q_{i+1} = \{q \in Q^R : (\exists q' \in Q_i)(\exists \rho \in (\Sigma^R)^+ \backslash (\Sigma^R \backslash \{\lambda\})^+)[q \in E^R(q', \rho)]\}$. If set $Q_{k_1+k_2}$ is empty, then system A is (k_1, k_2)-detectable; otherwise it is not (k_1, k_2)-detectable. The calculation cost of Q_{i+1} from Q_i is at most the size of $V^R(A)$ and we need to repeat this procedure for $k_1 + k_2$ times. Thus, verifying (k_1, k_2)-detectability can also be solved in polynomial-time in the number of states in $V^R(A)$.

Example 6. Consider system A_3 shown in Fig. 1(c), where $\Sigma_o = \{a, d, c\}$. We claim that system A is not $(1, 2)$-detectable. To see this, we first obtain the verification system $V(A_3)$ shown in Fig. 5.

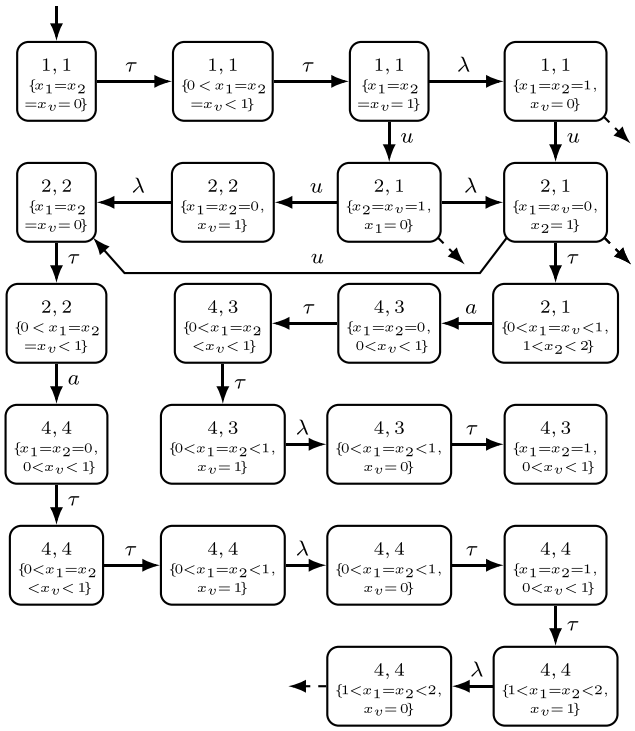


Fig. 6. Part of the region automaton $V^R(A_3)$.

Then, we construct the region automaton $V^R(A_3)$ of $V(A_3)$. Part of the region automaton $V^R(A_3)$ is shown in Fig. 6, where we can find a run

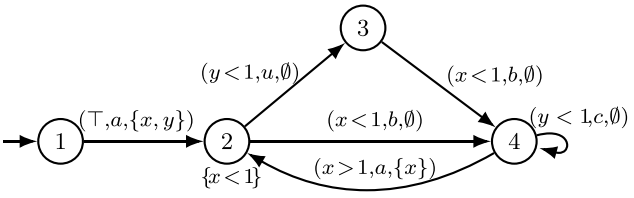
$$\begin{aligned}\pi_1 =&((1,1),\{x_1 = x_2 = x_v = 0\}) \xrightarrow{\tau\tau\lambda u}\\ &((2,1),\{x_1 = x_v = 0, x_2 = 1\}) \xrightarrow{u\tau a\tau\tau\lambda}\\ &((4,4),\{0 < x_1 = x_2 < 1, x_v = 0\}) \xrightarrow{\tau\tau\lambda}\\ &((4,4),\{1 < x_1 = x_2 < 2, x_v = 0\}),\end{aligned}$$

such that $q_1^R = ((2,1),\{x_1 = x_v = 0, x_2 = 1\}) \in AM^R$. Furthermore, for ρ_0, ρ_1 and ρ_2, each of them contains an event λ. That is, run π_1 satisfies all conditions in Theorem 3 and we can conclude that A_3 is not $(1, 2)$-detectable. In fact, run π_1 corresponds to the execution in the verification structure $V(A_2)$ as highlighted by red color in Fig. 5. We can extract a timed word $\rho = (1, u)(0.5, a)(1.5, \varepsilon)$ such that $\mathsf{Reach}((1, u), (0.5, a)(1.5, \varepsilon)) = \{1, 2\}$, $\mathsf{time}((1, u)) = 1$ and $\mathsf{time}((0.5, a)(1.5, \varepsilon)) = 2$. That is, we cannot determine the current state after one time elapsing event with the assistance of two time units delayed information. However, since there does not exist a run $\pi = q_0^R \xrightarrow{\rho_0^R} q_1^R \xrightarrow{\rho_1^R} q_2^R \xrightarrow{\rho_2^R} q_3^R$ such that $q_2^R \in AM^R$ and each ρ_i^R, $i \in \{0, 1, 2\}$ contains at least event λ, system A_3 is $(2, 1)$-detectable. For example, consider the run

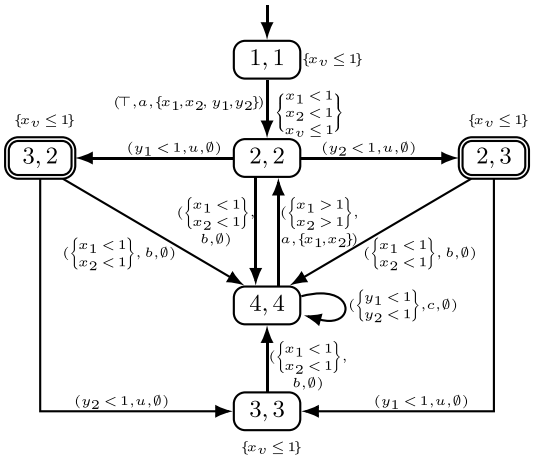
$$\begin{aligned}\pi_2 =&((1,1),\{x_1 = x_2 = x_v = 0\}) \xrightarrow{\tau\tau\lambda u}\\ &((2,1),\{x_1 = x_v = 0, x_2 = 1\}) \xrightarrow{\tau a\tau\tau\lambda}\\ &((4,3),\{0 < x_1 = x_2 < 1, x_v = 0\}).\end{aligned}$$

Although state $q_2^R = ((4,3),\{0 < x_1 = x_2 < 1, x_v = 0\})$ is ambiguous, there is no path from q_2^R that contains event λ.

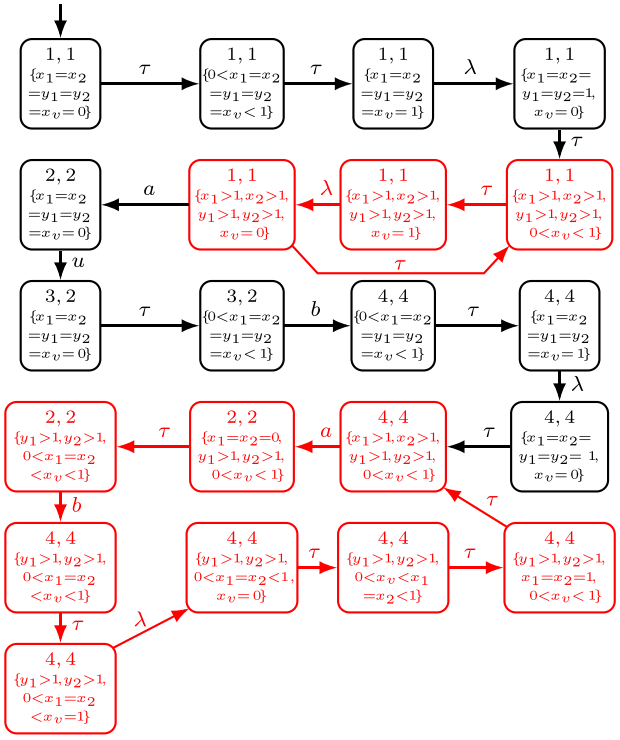
Finally, we illustrate our approach by a more complex timed automaton with two clocks and loop structures.



(a) System A_4 with $\Sigma_o = \{a, b, c\}$.



(b) Verification system $V(A_4)$ of timed system A_4 and we omit transition $(x_v = 1, \lambda, \{x_v\})$ at each discrete state.



(c) Part of the region automaton $V^R(A_4)$.

Fig. 7. An example for timed automaton with 2 clocks.

Example 7. Let us consider timed automaton A_4 shown in Fig. 7(a), which has two clocks, i.e., $\mathcal{X} = \{x, y\}$, and observable events $\Sigma_o = \{a, b, c\}$. We verify both strong detectability and delayed detectability for A_4. First, we construct its verification system $V(A_4)$ as depicted in Fig. 7(b). Then, based on the verification system $V(A_4)$, we compute the region automaton $V^R(A_4)$, which is partially shown in Fig. 7(c). Particularly, we can find the following run

$$\begin{aligned}\pi_1 =&((1,1),\{x_1=x_2=y_1=y_2=x_v=0\})\xrightarrow{\tau\tau\lambda\tau}\\ &((1,1),\{x_1>1,x_2>1,y_1>1,y_2>1,0<x_v<1\})\xrightarrow{\tau\lambda\tau}\\ &((1,1),\{x_1>1,x_2>1,y_1>1,y_2>1,0<x_v<1\})\\ &\xrightarrow{\tau\lambda au}((3,2),\{x_1=x_2=y_1=y_2=x_v=0\}),\end{aligned}$$

such that $q^R_{12} = ((3,2),\{x_1 = x_2 = y_1 = y_2 = x_v = 0\}) \in AM^R$ and there exists a cycle as highlighted in Fig. 7(c), where $q^R_5 = q^R_{10} = ((1,1),\{x_1 > 1, x_2 > 1, y_1 > 1, y_2 > 1, 0 < x_v < 1\})$ and $\sigma_6 = \lambda$. Thus, run π_1 satisfies the condition in Theorem 1, which means that A_4 is not strong detectable. On the other hand, for any $k_1, k_2 \in \mathbb{N}$ we can find another run

$$\begin{aligned}\pi_2 =&((1,1),\{x_1=x_2=y_1=y_2=x_v=0\})\\ &\xrightarrow{\tau\tau\lambda\tau(\tau\lambda\tau)^{k_1}\tau\lambda au}\\ &((3,2),\{x_1=x_2=y_1=y_2=x_v=0\})\\ &\xrightarrow{\tau b\tau\lambda\tau(a\tau b\tau\lambda\tau\tau\tau)^{k_2}}\\ &((4,4),\{x_1>1,x_2>1,y_1>1,y_2>1,0<x_v<1\})\end{aligned}$$

such that state $((3,2),\{x_1 = x_2 = y_1 = y_2 = x_v = 0\}) \in AM^R$. Therefore, run π_2 also satisfies the condition in Theorem 3, which means that system A_4 is not (k_1, k_2)-detectable for any $k_1, k_2 \in \mathbb{N}$.

7. Conclusion

In this paper, we investigated the verification of detectability for timed discrete-event systems in the dense-time framework. We extended three types of classical detectability: strong detectability, weak detectability and delayed detectability to the timed setting. Specifically, to verify strong detectability, we constructed the verification system based on the original system, and then provided a necessary and sufficient condition for strong detectability based on region automaton of the verification system. Furthermore, we showed that weak detectability is undecidable in the timed setting by reducing the universality problem for timed automata to the weak detectability verification problem. Based on the verification system constructed for strong detectability, we also provided a necessary and sufficient condition to verify delayed detectability. In the future, we would like to investigate enforcement of detectability in the timed setting. Also, we are interested in investigating detectability verification for discrete-time linear or nonlinear dynamical systems with continuous state spaces.

Acknowledgments

The authors would like to thank the anonymous reviewers for their useful comments that improved this paper. In particular, they sincerely thank an anonymous reviewer who pointed out that the verification conditions in Theorem 1 can be described by CTL formulae.

References

Alur, R., & Dill, D. (1994). A theory of timed automata. *Theoretical Computer Science, 126*(2), 183–235.

Ammar, I., El Touati, Y., Yeddes, M., & Mullins, J. (2021). Bounded opacity for timed systems. *Journal of Information Security and Applications, 61*, Article 102926.

Baier, C., & Katoen, J. (2008). *Principles of model checking*. MIT Press.

Balun, J., & Masopust, T. (2021). On verification of D-detectability for discrete event systems. *Automatica*, *133*, Article 109884.
Basile, F., Cabasino, M., & Seatzu, C. (2015). State estimation and fault diagnosis of labeled time Petri net systems with unobservable transitions. *IEEE Transactions on Automatic Control*, *4*(60), 997–1009.
Basile, F., & Ferrara, L. (2021). Finite-time accuracy of timed discrete event systems. In *60th IEEE conference on decision and control* (pp. 1744–1749).
Caines, P., Greiner, R., & Wang, S. (1988). Dynamical logic observers for finite automata. In *27th IEEE conference on decision and control* (pp. 226–233).
Cassandras, C., & Lafortune, S. (2021). *Introduction to discrete event systems*. Springer Nature.
Cassez, F. (2009). The dark side of timed opacity. In *International conference on information security and assurance* (pp. 21–30). Springer.
Cassez, F. (2012). The complexity of codiagnosability for discrete event and timed systems. *IEEE Transactions on Automatic Control*, *57*(7), 1752–1764.
Dong, W., Yin, X., Zhang, K., & Li, S. (2022). On the verification of detectability for timed systems. In *American control conference* (pp. 3752–3758).
Emerson, E., & Clarke, E. (1982). Using branching time temporal logic to synthesize synchronization skeletons. *Science of Computer Programming*, *2*(3), 241–266.
Gao, C., Lefebvre, D., Seatzu, C., Li, Z., & Giua, A. (2020). A region-based approach for state estimation of timed automata under no event observation. In *25th IEEE international conference on emerging technologies and factory automation* (pp. 799–804).
Hadjicostis, C. (2020). Introduction to estimation and inference in discrete event systems. In *Estimation and inference in discrete event systems* (pp. 1–14). Springer.
Hadjicostis, C., & Seatzu, C. (2016). K-detectability in discrete event systems. In *55th IEEE conference on decision and control* (pp. 420–425).
Henzinger, T., Nicollin, X., Sifakis, J., & Yovine, S. (1994). Symbolic model checking for real-time systems. *Information and Computation*, *111*(2), 193–244.
Jiang, S., Huang, Z., Chandra, V., & Kumar, R. (2001). A polynomial algorithm for testing diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, *46*(8), 1318–1321.
Keroglou, C., & Hadjicostis, C. (2015). Detectability in stochastic discrete event systems. *Systems & Control Letters*, *84*, 21–26.
Keroglou, C., & Hadjicostis, C. (2017). Verification of detectability in probabilistic finite automata. *Automatica*, *86*, 192–198.
Lai, A., Lahaye, S., & Giua, A. (2019). State estimation of max-plus automata with unobservable events. *Automatica*, *105*, 36–42.
Lai, A., Lahaye, S., & Giua, A. (2020). Verification of detectability for unambiguous weighted automata. *IEEE Transactions on Automatic Control*, *66*(3), 1437–1444.
Lai, A., Lahaye, S., & Komenda, J. (2022). Observer construction for polynomially ambiguous max-plus automata. *IEEE Transactions on Automatic Control*, *67*(3), 1582–1588.
Lan, H., Tong., Y., & Seatzu, C. (2021). Analysis of strong and strong periodic detectability of bounded labeled Petri nets. *Nonlinear Analysis. Hybrid Systems*, *42*, Article 101087.
Li, J., Lefebvre, D., Hadjicostis, C., & Li, Z. (2022). Observers for a class of timed automata based on elapsed time graphs. *IEEE Transactions on Automatic Control*, *67*(2), 767–779.
Lin, F., Wang, L., Yin, G., Polis, M., & Chen, W. (2022). On detectability of a class of hybrid systems. *IEEE Transactions on Automatic Control*, 1–12.
Ma, Z., Li, Z., & Giua, A. (2019). Marking estimation in a class of time labeled Petri nets. *IEEE Transactions on Automatic Control*, *65*(2), 493–506.
Masopust, T. (2018). Complexity of deciding detectability in discrete event systems. *Automatica*, *93*, 257–261.
Masopust, T., & Yin, X. (2019). Deciding detectability for labeled Petri nets. *Automatica*, *104*, 238–241.
Ramadge, P. (1986). Observability of discrete event systems. In *25th IEEE conf. decision and control* (pp. 1108–1112).
Shu, S., & Lin, F. (2012). Delayed detectability of discrete event systems. *IEEE Transactions on Automatic Control*, *58*(4), 862–875.
Shu, S., & Lin, F. (2013a). Enforcing detectability in controlled discrete event systems. *IEEE Transactions on Automatic Control*, *58*(8), 2125–2130.
Shu, S., & Lin, F. (2013b). I-Detectability of discrete-event systems. *IEEE Transactions on Automation Science and Engineering*, *10*(1), 187–196.
Shu, S., Lin, F., & Ying, H. (2007). Detectability of discrete event systems. *IEEE Transactions on Automatic Control*, *52*(12), 2356–2359.
Tripakis, S. (2002). Fault diagnosis for timed automata. In *International sym. formal techniques in real-time and fault-tolerant systems* (pp. 205–221). Springer.
Wang, L., Zhan, N., & An, J. (2018). The opacity of real-time automata. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *37*(11), 2845–2856.
Yin, X., Li, Z., & Wang, W. (2018). Trajectory detectability of discrete-event systems. *Systems & Control Letters*, *119*, 101–107.
Yoo, T., & Lafortune, S. (2002). Polynomial-time verification of diagnosability of partially observed discrete-event systems. *IEEE Transactions on Automatic Control*, *47*(9), 1491–1495.
Zhang, K. (2017). The problem of determining the weak (periodic) detectability of discrete event systems is PSPACE-complete. *Automatica*, *81*, 217–220.
Zhang, K. (2021). State-based opacity of real-time automata. In *27th IFIP WG 1.5 international workshop on cellular automata and discrete complex systems*.
Zhang, K., & Giua, A. (2018). Weak (approximate) detectability of labeled Petri net systems with inhibitor arcs. *IFAC-PapersOnLine*, *51*(7), 167–171.
Zhang, K., & Giua, A. (2020). On detectability of labeled Petri nets and finite automata. *Discrete Event Dynamic Systems*, *30*, 465–497.

Weijie Dong was born in Xinjiang, China, in 1996. He received the B.S. degree from Huazhong University of Science and Technology in 2019. He is pursuing the Ph.D. degree at Shanghai Jiao Tong University in the Department of Automation. His current research interests include fault diagnosis, state estimation of discrete event systems.

Kuize Zhang received the B.S. and Ph.D. degrees in Mathematics and Control Science and Engineering from Harbin Engineering University, China, in 2009 and 2014, respectively. He is currently a lecturer at Department of Computer Science, University of Surrey, UK. He was an Alexander von Humboldt Fellow at Technical University of Berlin, Germany, from July 2020 to April 2022. He held postdoc positions at KTH Royal Institute of Technology, Sweden (2017-2020) and Technical University of Munich, Germany (2016-2017). His current research interests include Boolean control networks, finite automata, Petri nets, weighted automata over monoids, timed automata, etc., in theoretical computer science and control theory, with applications to systems biology, etc. He is a senior member of IEEE (2017-). He has co-authored one monograph in Springer Nature and authored one monograph in Foundations and Trends® in Systems and Control.

Shaoyuan Li was born in Hebei, China, in 1965. He received the B.S. and M.S. degrees in automation from the Hebei University of Technology, Tianjin, China, in 1987 and 1992, respectively, and the Ph.D. degree from Nankai University, Tianjin, in 1997. Since 1997, he has been with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China, where he is currently a Professor. His current research interests include model predictive control, dynamic system optimization, and cyber–physical systems. He is the vice-president of the Chinese Association of Automation.

Xiang Yin was born in Anhui, China, in 1991. He received the B.Eng degree from Zhejiang University in 2012, the M.S. degree from the University of Michigan, Ann Arbor, in 2013, and the Ph.D degree from the University of Michigan, Ann Arbor, in 2017, all in electrical engineering. Since 2017, he has been with the Department of Automation, Shanghai Jiao Tong University, where he is an Associate Professor. His research interests include formal methods, discrete-event systems and cyber–physical systems.

Dr. Yin is serving as the chair of the IEEE CSS Technical Committee on Discrete Event Systems, an Associate Editor for the Journal of Discrete Event Dynamic Systems: Theory & Applications, and a member of the IEEE CSS Conference Editorial Board. Dr. Yin received the IEEE Conference on Decision and Control (CDC) Best Student Paper Award Finalist in 2016.