2022 American Control Conference (ACC)
Atlanta, USA, June 8-10, 2022

On the Verification of Detectability for Timed Systems

Weijie Dong, Xiang Yin, Kuize Zhang and Shaoyuan Li

Abstract— **In this paper, we investigate the verification of detectability, a fundamental state estimation property, for partially-observed discrete event systems (DES). Existing works on this topic mainly focus on untimed DES. In many 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 DES modeled by timed automata. Two notions of detectability, strong detectability and weak detectability, are studied in the dense-time setting, which characterize detectability by time elapsing rather than events steps. We show that verifying strong detectability for timed system is decidable by providing a verifiable necessary and sufficient condition. Furthermore, we show that weak detectability is undecidable in the timed setting by reducing the language universality problem for timed automata to this verification problem. Our results extend the detectability analysis of DES from the untimed setting to a timed setting.**

I. Introduction

State estimation is one of the most fundamental problems in complex engineering systems. In practice, the user cannot directly obtain the full state information of the system due to observation uncertainties or nondeterminism of the system dynamic. Therefore, one needs 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 the context of discrete events systems (DES), Shu and Lin [20] first systematically investigated the four different types of detectability as well as their verification algorithms. To further characterize different state estimation requirements, several variations of detectability have also been proposed in the literature, including, e.g., delayed detectability [15], [17], K-detectability [7], I-detectability [19], [23], D-detectability [4], [17] and trajectory detectability in [26]. The enforcement of detectability has been investigated in [18], [24], [25]. Recently, detectability verification has been extended to more complex DES models, including, e.g., labeled Petri nets [14], [16], [29], probabilistic automata [11] and unambiguous weighted automata [13].

However, the above mentioned works only consider DES without *real time manners*. In practice, many real-world engineering systems may have time constraints when they generate events. Thus, it is necessary to model and analyze executions of the system with time constraints. Such a real-time issue can be modeled by timed automata proposed in the seminal work of Alur and Dill [1]. In the context of detectability analysis, if one can “measure” the time execution, e.g., by having a global clock [21], 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 DES modeled by timed automata. The main contributions of this work are as follows. First, we introduce the notions of strong detectability and weak detectability for timed systems. Different from existing notions in the untimed setting [20], where the number of observable events is used to count the observation delay, here we directly utilize the real time information to describe the conditions of detectability which is a more natural and realistic measure for delays. Second, we present an effective algorithm for checking strong detectability for timed systems. Our approach is to first construct a verification structure based on the original system and then to reduce the strong detectability verification problem to a reachability problem in the region graph of the verification structure. Finally, we show that the verification of weak detectability, which is decidable in exponential time in untimed setting [20], is undecidable in the timed setting.

We note that state estimation for timed systems has been investigated recently in the literature [6], [10], [12]. For example, Gao et al. [6] discussed how to perform state estimation for timed automata using λ-observers. In [12], Lai et al. investigated the state estimation problem of timed systems by interpreting time into weights and using max-plus automata to model timed systems. However, both of the aforementioned works focus on the online state estimation problem and do not consider the inherent property of detectability. Furthermore, they consider restricted classes of timed systems, such as timed automata with one clock with reset at each event occurrence [6], [10] and timed systems whose time elapsing can be modeled as weights [12]. In the context of property verification of timed automata, effective algorithms have been proposed for checking (co)diagnosability [5], [21]. More recently, verification for opacity has also been investigated for timed systems [2], [22], [28]. However, these notions are incomparable to detectability, which has been argued in the untimed setting [20].

This work was supported by the National Natural Science Foundation of China (62061136004, 62173226, 61833012) and the Alexander von Humboldt Foundation.

Weijie Dong Xiang Yin and Shaoyuan Li are with Department of Automation and Key Laboratory of System Control and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: {wjd_dollar,yinxiang,syli}@sjtu.edu.cn.

Kuize Zhang is with Department of Computer Science, University of Surrey, Guildford GU2 7XH, U.K. E-mail: kuize.zhang@campus.tu-berlin.de.

978-1-6654-5196-3/$31.00 ©2022 AACC

II. Preliminaries

A. Timed Systems

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 taking values in $\mathbb{R}_{\geq 0}$ 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 set $\mathcal{Y} \subseteq \mathcal{X}$ 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 $\Delta \in \mathbb{R}_{\geq 0}$, $v + \Delta$ is the valuation such that $(v+\Delta)(x) = v(x) + \Delta$, for every $x \in \mathcal{X}$.

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 $v \models g$. For any clock $x \in \mathcal{X}$, we use c_x to denote the maximum integer c such that $x \sim c \in C(\mathcal{X})$, where $\sim \in \{\leq, <, \geq, >, =\}$.

In this paper, we consider timed discrete-event systems modeled by timed automata [1], [8]. Formally, a timed automaton (TA) is a sixtuple

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

where Q is a finite set of discrete states; $q_0 \in Q$ is the initial discrete state; Σ is a finite set of events; $\mathcal{X}$ 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 $\mathsf{inv}(q)$ specifying the length of time 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 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 $s_0 = (q_0, v_0)$, where q_0 is the initial discrete state and v_0 is the initial valuation such that $v_0(x) = 0$ for all $x \in \mathcal{X}$.

A finite (infinite) *word* over Σ is a finite (infinite) sequence $\sigma_1 \ldots \sigma_n(\ldots)$; we denote by Σ^* and Σ^ω, respectively, the set of finite words and the set of infinite words over Σ. A *timed word* over Σ is a word over $\mathbb{R}_{\geq 0} \times \Sigma$. 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. Given a timed word $\rho \in \mathsf{TW}(\Sigma)$, we define $\mathsf{Pre}(\rho) = \{\rho' \in \mathsf{TW}^*(\Sigma) : \exists \rho'' \in \mathsf{TW}(\Sigma) \text{ s.t. } \rho'\rho'' = \rho\}$ as the set of all finite prefixes of ρ. For any timed word $\rho = (\Delta_1, \sigma_1)(\Delta_2, \sigma_2)\ldots$, we define $\mathsf{time}(\rho) = \sum_{i=1}^{|\rho|} \Delta_i$ as the total time elapsing in ρ and define $\mathsf{utw}(\rho) = \sigma_1\sigma_2\ldots$ as its untimed word.

In timed automata, there are two types of transitions: delay transitions and discrete 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 transition $(q, v) \xrightarrow{\Delta} (q, v + \Delta)$ is defined if $v + \Delta' \models \mathsf{inv}(q)$ holds for any $0 \leq \Delta' \leq \Delta$;
- a discrete transition $(q, v) \xrightarrow{\sigma} (q', v')$ 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 transition represents the elapse of time duration Δ and a discrete transition corresponds to a discrete state transition with event σ. For simplicity, we write $s \xrightarrow{(\Delta,\sigma)} s'$ if there exists a state s'' such that $s \xrightarrow{\Delta} s''$ and $s'' \xrightarrow{\sigma} s'$. Given a state $s = (q, v)$, its discrete state component is denoted by $\mathsf{dis}(s) = q$.

An infinite *run* of time automaton A starting at state s is an infinite sequence

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

where $s_0 = s$ and $s_i \xrightarrow{(\Delta_i,\sigma_i)} s_{i+1}$ holds for any $i \geq 0$. A finite run of A is defined analogously. We denote by $\mathsf{Run}^\omega(A)$ and $\mathsf{Run}^*(A)$, respectively, the set of infinite runs and finite runs in A staring at s_0 with $\mathsf{Run}(A) = \mathsf{Run}^\omega(A) \cup \mathsf{Run}^*(A)$. For any run $\pi \in \mathsf{Run}(A)$, 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*. For any finite path $s_\pi = s_0 s_1 \cdots s_n$, we denote by $\mathsf{last}(s_\pi)$ 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 $\mathsf{Run}(\rho) = \{\pi \in \mathsf{Run}(A) : \rho_\pi = \rho\}$ and 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 sake of simplicity, we denote by $\mathsf{last}_d(\cdot) = \mathsf{dis}(\mathsf{last}(\cdot))$ the discrete states of the last part.

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. In this paper, we assume that the TA is *timelock-free* (called *well-timed* in [21]) in the sense that there is no timelock state reachable. This assumption holds for plenty of engineering systems as a timelock state will prevent time progressing, which is not realistic in real-world systems.

B. Region Automata

For later technical developments, here we briefly review the region automata [1], which are widely used as finite abstractions of timed automata for the purpose of verification. The reader can refer to [1], [3] 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 $\mathcal{R}$. The region automaton of A is $RG(A) = (Q^R, q^R_0, \Sigma^R, E^R)$, where $Q^R = Q \times \mathcal{R}$ is the set of states, $q^R_0 = (q_0, \mathbf{0}_{\mathcal{X}})$ 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 the time successor region of r, which is obtained by time elapsing. Details about region abstraction and how the above transition function E^R is defined can be found in [1], [3]. Function E^R is extended to $Q^R \times (\Sigma^R)^*$ in the usual way. The language generated by $RG(A)$ is $\mathcal{L}(RG(A)) = \{\rho \in (\Sigma^R)^* \cup (\Sigma^R)^\omega : E^R(q^R_0, \rho)!\}$, where ! means "is defined". A run in $RG(A)$ is a finite or infinite sequence $\pi = q^R_1 \xrightarrow{\sigma_1} q^R_2 \xrightarrow{\sigma_2} \cdots q^R_n \cdots$, where $q^R_i \in Q^R, \sigma_i \in \Sigma^R$ and $q^R_{i+1} \in E^R(q^R_i, \sigma_i), i = 1, 2, \ldots$.

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 $RG(A)$ [1]:

$$\mathsf{utw}(\mathsf{TW}(A)) = \mathsf{utw}^R(RG(A)). \tag{1}$$

Based on the relation in Equation (1), the region automata preserves reachability of discrete state in the original system A, that is, there exists a timed word ρ reaching discrete state $q \in Q$, i.e., $q \in \mathsf{last}_d(\rho)$, if and only if there is a word ρ^R in $RG(A)$ such that $(q, r) \in E^R(q^R_0, \rho^R)$ where $(q, r) \in Q^R$.

III. Detectability for Timed Systems

Given a finite timed word $\rho = (\Delta_0, \sigma_0) \ldots (\Delta_n, \sigma_n) \in \mathsf{TW}^*(\Sigma)$, 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 $(\Delta, \sigma) \in \mathbb{R}_{\geq 0} \times \Sigma$, $P((\Delta, \sigma)) = \begin{cases} (\Delta, \sigma) & \text{if } \sigma \in \Sigma_o \\ (\Delta, \varepsilon) & \text{otherwise} \end{cases}$
- 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 $P : 2^{\mathsf{TW}^*(\Sigma)} \to 2^{\mathsf{TW}^*(\Sigma)}$ 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.,

$$\mathsf{Reach}(t) = \left\{ q \in Q : \begin{array}{c} \exists \pi \in \mathsf{Run}^*(A) \text{ s.t.} \\ P(\rho_\pi) = t \wedge (q, v) = \mathsf{last}(s_\pi) \end{array} \right\}$$

In the seminal work of Shu and Lin [20], strong detectability and weak detectability have been proposed to capture different state estimation requirements. In particular, strong detectability is the strongest 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. However, the original definitions by Shu and Lin are proposed for untimed finite-state automata without utilizing time information. Here, we extend these notions to a timed setting as follows.

Definition 1: Let $A = (Q, q_0, \Sigma, \mathcal{X}, \mathsf{inv}, E)$ be a timed DES 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 delay, i.e.,

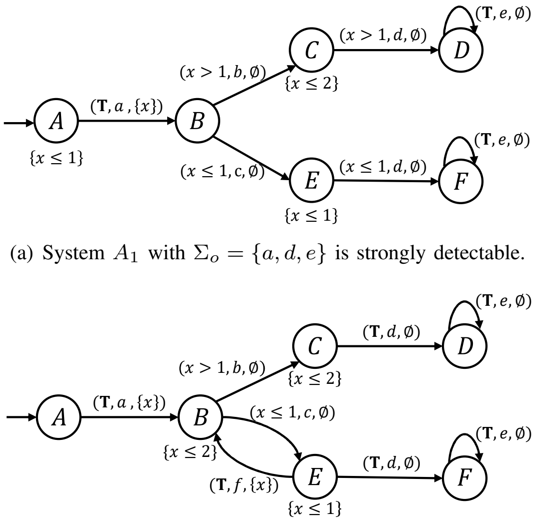
$$\begin{aligned} &(\exists \Delta \in \mathbb{R}_{\geq 0})(\forall \pi \in \mathsf{Run}^*(A)) \\ &\mathsf{time}(\rho_\pi) \geq \Delta \Rightarrow |\mathsf{Reach}(P(\rho_\pi))| = 1. \end{aligned}$$

- *weakly detectable* if we can determine the current and subsequent states of the system unambiguously for some runs in A, i.e.,

$$\begin{aligned} &(\exists \Delta \in \mathbb{R}_{\geq 0})(\exists \pi \in \mathsf{Run}^\omega(A))(\forall \rho \in \mathsf{Pre}(\rho_\pi)) \\ &\mathsf{time}(\rho) \geq \Delta \Rightarrow |\mathsf{Reach}(P(\rho))| = 1. \end{aligned}$$

We illustrate the definitions of detectability by the following examples.

Example 1: Let us consider timed system A_1 shown in Figure 1(a) with $\Sigma_o = \{a, d, e\}$. Note that in the untimed setting, this system is not (either strongly or weakly) detectable without utilizing the time information. This is because along the only possible observation $adeee\cdots$, we can never distinguish between states D and F. However, in the timed setting, this system is strongly detectable (hence, also weakly detectable). Specifically, 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 A is $x \leq 1$, which means that the system should depart from state A to B within one time unit and we know for sure that the system is at state B immediately after observing the first event a. Since the transitions along $B \to C \to D$ require more than



(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.

Fig. 1: Two timed DESs A_1 and A_2 in (a) and (b) respectively. For each guard, $\mathbf{T}$ is the abbreviation of true. 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.

one but no more than two time units, while transitions along $B \to E \to F$ 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 F;
- 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 D;
- If we observe nothing within two time units after observing a, then we know for sure that the system will stay at B forever because the invariant of state C is $x \leq 2$.

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

Example 2: Let us consider timed system A_2 shown in Figure 1(b) with $\Sigma_o = \{a, d, e, f\}$. In this case, there exists a run such that we can not distinguish state B and E after a finite time of observations even by utilizing the time information. For example, if we choose arbitrary $\Delta \in \mathbb{N}$, there is a run

$$\pi = A(1, a)[B(0.5, c)E(0.5, f)]^{\Delta} \in \mathsf{Run}(A)$$

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

$$\pi' = A(1, a)B(1.5, b)C(0.2, d)[D(1, e)]^{\omega}$$

such that for any $\rho \in \mathsf{Pre}(\rho_{\pi'})$ satisfying $\mathsf{time}(\rho) \geq (\Delta_0 + 1.7)$, we can uniquely determine the state, i.e., $\mathsf{Reach}(P(\rho_{\pi'})) = \{D\}$. Thus, system A_2 is weakly detectable.

IV. 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.

A. 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 words has finite or infinite time elapsing. Given timed DES $A = (Q, q_0, \Sigma, \mathcal{X}, \mathsf{inv}, E)$, the verification system of A is a new timed automaton

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

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;
- $\mathsf{inv}_V : Q_V \to C(\mathcal{X}_V)$ is the invariant function defined by: for any $(q_1, q_2) \in Q_V$, $\mathsf{inv}_V(q_1, q_2) = \mathsf{inv}(q_1) \land \widehat{\mathsf{inv}}(q_2) \land x_v \leq 1$, where $\widehat{\mathsf{inv}}(q)$ simply replaces each $x \in \mathcal{X}$ in $\mathsf{inv}(q)$ 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
  $$\begin{aligned} &(q_1, \sigma, g_1, \mathcal{Y}_1, q_1'), (q_2, \sigma, g_2, \mathcal{Y}_2, q_2') \in E \\ \Rightarrow &((q_1, q_2), \sigma, g_1 \land \hat{g}_2, \mathcal{Y}_1 \cup \hat{\mathcal{Y}}_2, (q_1', q_2')) \in E_V \end{aligned} \quad (2)$$
  - if $\sigma \in \Sigma_{uo}$, then
  $$\begin{aligned} &(q_1, \sigma, g_1, \mathcal{Y}_1, q_1') \in E \Rightarrow ((q_1, q_2), \sigma, g_1, \mathcal{Y}_1, (q_1', q_2)) \in E_V \\ &(q_2, \sigma, g_2, \mathcal{Y}_2, q_2') \in E \Rightarrow ((q_1, q_2), \sigma, \hat{g}_2, \hat{\mathcal{Y}}_2, (q_1, q_2')) \in E_V \end{aligned} \quad (3)$$
  - if $\sigma = \lambda$, then
  $$((q_1, q_2), \sigma, x_v = 1, \{x_v\}, (q_1, q_2)) \in E_V, \quad (4)$$

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

Intuitively, in the verification system $V(A)$, each discrete state is a pair of discrete states of 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 from $\mathcal{X}$. The invariant for each state is the conjunction of invariants of its two components in the state. In addition, we add $x_v \leq 1$ for new clock x_v; this together with the transition in Equation (4) guarantee that event λ can occur every time unit. 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 the 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 [21]:

- 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 $\pi_1, \pi_2 \in \mathsf{Run(A)}$ such that $\mathsf{last}_d(s_{\pi_1}) = q_n, \mathsf{last}_d(s_{\pi_2}) = q_n'$ 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 $(\mathsf{dis}(\mathsf{last}(s_{\pi_1})), \mathsf{dis}(\mathsf{last}(s_{\pi_2})))$, i.e.,

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

B. 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 and, if an ambiguous state is reached, then we cannot distinguish which state the system A is actually in by its 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 an 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) $\exists i < k \leq j : \sigma_k = \lambda$.

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 state q_j^R. The cycle π' and event λ in it supply capacity of any time elapsing. And the reachability from q_j^R to an ambiguous state prevent us determining states unambiguously.

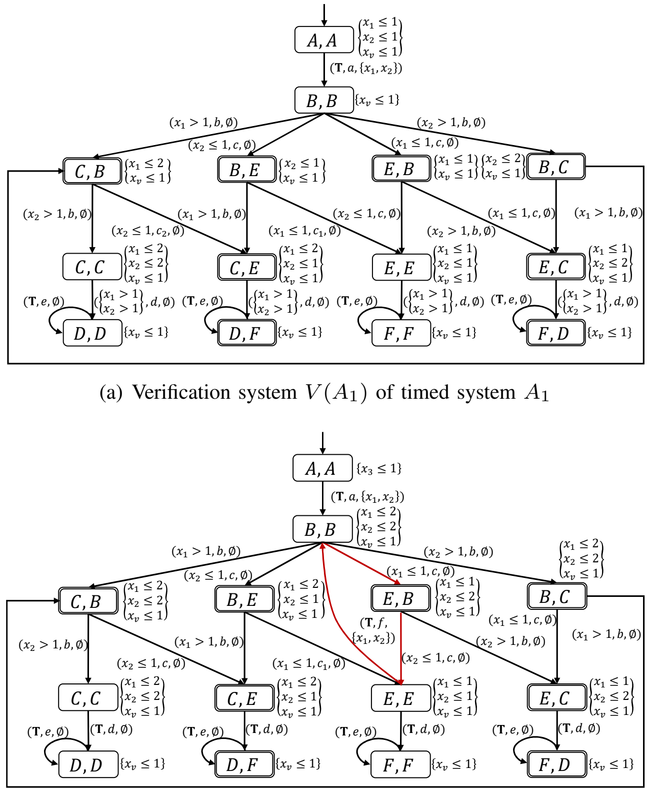
Remark 1: Let us discuss the complexity of checking strong detectability for a timed system A. The verification system $V(A)$ has at most $|Q|^2$ states and $|Q|^2(|Q|^2 - 1)$ transitions, where $|Q|$ is the number of states in A. Because the clocks consist of two copies of the original and a new clock, 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 [1], 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, we can construct $V^R(A)$ 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 solved in time polynomial in the number of states in $V^R(A)$. Therefore, the whole complexity mainly relies on the size of region automaton $V^R(A)$.

Example 3: Let us consider timed DES A_1 shown in Figure 1(a) with observable event set $\Sigma_o = \{a, d, e, f\}$. We obtain the verification system $V(A_1)$ by aforementioned steps, which is depicted in Figure 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 any run satisfying conditions in Theorem 1. Thus, A_1 is strongly detectable.

Example 4: However, system A_2 shown in Figure 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 Figure 2(b), and based on which we construct the region automaton $V^R(A_2)$ of $V(A_2)$. Part of the region automaton $V^R(A_2)$ is shown in Figure 3. Here, we can find a a run,

$$\begin{aligned}\pi =&((A, A), x_1 = x_2 = x_v = 0) \xrightarrow{a}\\ &((B, B), x_1 = x_2 = x_v = 0) \xrightarrow{\tau}\\ &((B, B), 0 \leq x_1 = x_2 = x_v \leq 1) \xrightarrow{\tau}\\ &((B, B), x_1 = x_2 = x_v = 1) \xrightarrow{\lambda}\\ &((B, B), x_1 = x_2 = 1 \wedge x_v = 1) \xrightarrow{c}\\ &((E, B), x_1 = x_2 = 1 \wedge x_v = 0) \xrightarrow{f}\\ &((B, B), x_1 = x_2 = x_v = 0) \xrightarrow{c}\\ &((E, B), x_1 = x_2 = x_v = 0),\end{aligned}$$

such that $q_7^R = ((E,B), x_1 = x_2 = x_v = 0) \in AM^R$. Furthermore, run π contains a cycle part as highlighted in Figure 3, where $q_1^R = q_6^R = ((B,B), x_1 = x_2 = x_v = 0)$ 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 Figure 2(b). Based on the cycle, we can actually extract a timed word $\rho = (0,a)[(1,c)(0,f)]^\omega$ such that $\mathsf{time}(\rho) = \infty$. Because event c is unobservable, we can never distinguish between system states B and E.



(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 every guard, the abbreviation of true is denoted by $\mathbf{T}$.

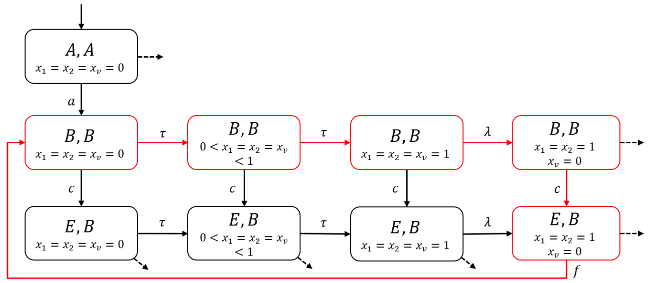


Fig. 3: Part of the region automaton $V^R(A_2)$. The cycle highlighted in red contains event λ and an ambiguous state $((E,B), x_1 = x_2 = x_v = 0)$.

Remark 2: The basic idea of the verification system $V(A)$ is motivated by the construction for the verification of diagnosability in untimed DES [9], [27] and timed DES [21], where it is termed as the twin-plant or the verifier. Our construction of the verification system itself is quite similar to that of [21]. However, the necessary and sufficient condition derived is quite different. In particular, in diagnosaibility 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 arbitrary long prefix. This condition cannot be captured directly by standard model checking conditions, which motivates the use of region graph to test the condition.

V. 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 DES 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 [1], 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

- $Q_G = Q \cup \{q_{ini}, q_B\}$, where q_{int} and q_B are two new discrete states;
- $q_{ini} \in Q_G$ is the initial discrete state;
- $\Sigma_G = \Sigma \cup \{\sigma_o\}$, where σ_o 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 $x \leq 1$ and $\mathsf{inv}_G(q_B)$ is true;
- E_G is the set of transitions defined by

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

The construction of G is depicted in Figure 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.

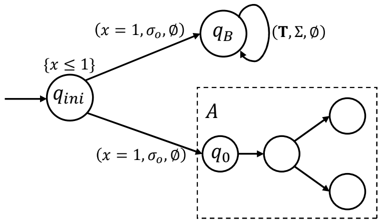

$(x=1,\sigma_o,\emptyset)$
q_B
$(\mathbf{T},\Sigma,\emptyset)$
$\{x \leq 1\}$
q_{ini}
A
$(x=1,\sigma_o,\emptyset)$
q_0

Fig. 4: Illustration of the construction of G. The edge $(\mathbf{T}, \Sigma, \emptyset)$ attached to q_B represents the set of edges $\{(\mathbf{T}, \sigma, \emptyset) : \sigma \in \Sigma\}$.

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 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 state discrete 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 is 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 detectble. The above observations lead to the following main theorem.

Theorem 2: Weak detectability is undecidable for timed automata.

VI. Conclusion

In this paper, we investigated the verification of detectability for timed discrete-event systems in the dense-time framework. We extended both strong detectability and weak detectability to a 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 TA to the weak detectability verification problem. In the future, we would like to further investigate more types of detectability in the time setting, including, e.g., periodic detectability and delayed detectability.

References

[1] R. Alur and D. Dill. A theory of timed automata. *Theoretical computer science*, 126(2):183–235, 1994.

[2] I. Ammar, Y. El Touati, M. Yeddes, and J. Mullins. Bounded opacity for timed systems. *Journal of Information Security and Applications*, 61:102926, 2021.

[3] C. Baier and J. Katoen. *Principles of model checking*. MIT press, 2008.

[4] J. Balun and T. Masopust. On verification of d-detectability for discrete event systems. *Automatica*, 133:109884, 2021.

[5] F. Cassez. The Complexity of Codiagnosability for Discrete Event and Timed Systems. *IEEE Transactions on Automatic Control*, 57(7):1752–1764, 2012.

[6] C. Gao, D. Lefebvre, C. Seatzu, Z. Li, and A. Giua. 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 (ETFA)*, volume 1, pages 799–804. IEEE, 2020.

[7] C. Hadjicostis and C. Seatzu. K-detectability in discrete event systems. In *55th IEEE Conference on Decision and Control (CDC)*, pages 420–425, 2016.

[8] T. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. *Information and computation*, 111(2):193–244, 1994.

[9] S. Jiang, Z. Huang, V. Chandra, and R. Kumar. A polynomial algorithm for testing diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, 46(8):1318–1321, 2001.

[10] L. Jun, D. Lefebvre, C. Hadjicostis, and Z. Li. Observers for a class of timed automata based on elapsed time graphs. *IEEE Transactions on Automatic Control*, 2021.

[11] C. Keroglou and C. Hadjicostis. Verification of detectability in probabilistic finite automata. *Automatica*, 86:192–198, 2017.

[12] A. Lai, S. Lahaye, and A. Giua. State estimation of max-plus automata with unobservable events. *Automatica*, 105:36–42, 2019.

[13] A. Lai, S. Lahaye, and A. Giua. Verification of detectability for unambiguous weighted automata. *IEEE Transactions on Automatic Control*, 66(3):1437–1444, 2020.

[14] H. Lan, Y. Tong., and C. Seatzu. Analysis of strong and strong periodic detectability of bounded labeled petri nets. *Nonlinear Analysis: Hybrid Systems*, 42:101087, 2021.

[15] Y. Liu, Z. Liu, X. Yin, and S. Li. An improved approach for verifying delayed detectability of discrete-event systems. *Automatica*, 124:109291, 2021.

[16] T. Masopust and X. Yin. Deciding detectability for labeled Petri nets. *Automatica*, 104:238–241, 2019.

[17] S. Shu and F. Lin. Delayed detectability of discrete event systems. *IEEE Transactions on Automatic Control*, 58(4):862–875, 2012.

[18] S. Shu and F. Lin. Enforcing Detectability in Controlled Discrete Event Systems. *IEEE Transactions on Automatic Control*, 58(8):2125–2130, 2013.

[19] S. Shu and F. Lin. I-Detectability of Discrete-Event Systems. *IEEE Transactions on Automation Science and Engineering*, 10(1):187–196, 2013.

[20] S. Shu, F. Lin, and H. Ying. Detectability of Discrete Event Systems. *IEEE Transactions on Automatic Control*, 52(12):2356–2359, 2007.

[21] S. Tripakis. Fault diagnosis for timed automata. In *International symposium on formal techniques in real-time and fault-tolerant systems*, pages 205–221. Springer, 2002.

[22] L. Wang, N. Zhan, and J. An. The opacity of real-time automata. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(11):2845–2856, 2018.

[23] X. Yin. Initial-state detectability of stochastic discrete-event systems with probabilistic sensor failures. *Automatica*, 80:127–134, 2017.

[24] X. Yin and S. Lafortune. A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems. *IEEE Transactions on Automatic Control*, 61(8):2140–2154, 2016.

[25] X. Yin and S Lafortune. A general approach for optimizing dynamic sensor activation for discrete event systems. *Automatica*, 105:376–383, 2019.

[26] X. Yin, Z. Li, and W. Wang. Trajectory detectability of discrete-event systems. *Systems & Control Letters*, 119:101–107, 2018.

[27] T. Yoo and S. Lafortune. Polynomial-time verification of diagnosability of partially observed discrete-event systems. *IEEE Transactions on Automatic Control*, 47(9):1491–1495, 2002.

[28] K. Zhang. State-based opacity of real-time automata. In *27th IFIP WG 1.5 International Workshop on Cellular Automata and Discrete Complex Systems*, 2021.

[29] K. Zhang and A. Giua. On detectability of labeled petri nets and finite automata. *Discrete Event Dynamic Systems*, 30:465–497, 2020.