Automatica 124 (2021) 109291

Contents lists available at ScienceDirect

Automatica

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

Technical communique

# An improved approach for verifying delayed detectability of discrete-event systems[☆]

Yang Liu[a,b], Zhaocong Liu[a], Xiang Yin[a,*], Shaoyuan Li[a]

[a] Department of Automation and Key Laboratory of System Control and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China
[b] School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

ARTICLE INFO

*Article history:*
Received 10 May 2020
Received in revised form 21 July 2020
Accepted 26 July 2020
Available online 3 October 2020

ABSTRACT

In this paper, we investigate state estimation and detection problems with information delays in the context of partially-observed discrete-event systems. Specifically, we study the verification of an important detectability property called delayed detectability which is related to the state estimation problem with information delayed. Particularly, it requires that the state of the system after $k_1$ observations can always be detected within another $k_2$ observation delay; delayed detectability is, therefore, referred to as $(k_1, k_2)$-detectability. In this paper, we provide a new verification algorithm for checking this property. The idea is to use the reversed dynamic of the system to efficiently estimate the delayed-state information. To this end, a new information structure called the two-way verifier is proposed. We show that our result improves the complexity of existing verification algorithms for the property. We also illustrate our result by simple examples.



## 1. Introduction

Discrete-event systems (DES) are dynamic systems with discrete state–spaces and event-driven dynamics. DES are widely used for modeling both logic systems that are inherently event-driven and symbolic abstractions of continuous/hybrid dynamic systems. In many problems, e.g., supervisory control and fault diagnosis, the state information of the system is usually crucial for the purpose of decision making. However, in many real world applications, we do not always have perfect knowledge of the system due to imperfect information or measurement uncertainties. Therefore, state estimation and detection are important issues in the analysis and design of partially-observed DES.

In the context of DES, the problem of state estimation dates back to the study of the property of observability; see, e.g., Lin and Wonham (1988), Ramadge (1986) and Özveren and Willsky (1990). In this problem, it is assumed that the system's behavior is only partially-known and we want to infer the system's "state" based on the imperfect information. State estimate of DES has also been investigated for different classes of system models including max-plus automata (Lai, Lahaye, & Giua, 2019), timed Petri nets (Ma, Li, & Giua, 2019) and stochastic Petri nets (Ammour, Leclercq, Sanlaville, & Lefebvre, 2017).

Recently, the state estimation of DES has been investigated in a more systematic manner in the context of detectability. The concept of detectability was first proposed by Shu and Lin in Shu, Lin, and Ying (2007), where several different notions of detectability, e.g., strong (periodic) detectability and weak (periodic) detectability, are defined. Specifically, the authors of Shu et al. (2007) considered DES modeled as finite-state automata with unobservable events. Then strong detectability captures whether or not we can always detect the current state of the system within a finite delay, while weak detectability captures whether or not the current state of the system can be detected for some path generated by the system. Verification algorithms were also provided for different notions of detectability in Shu et al. (2007). In Shu and Lin (2011), a polynomial-time algorithm was provided for the verification of strong detectability. However, it has been shown more recently by Masopust (2018) and Zhang (2017) that verifying weak detectability is PSPACE-hard. Therefore, it is unlikely that a polynomial-time algorithm exists for the verification of weak detectability.

Since the seminal work of Shu and Lin, the concept of detectability has been studied more extensively and has been extended to different settings. In Shu and Lin (2013a), the concept of delayed detectability was proposed by allowing the usage of

☆ This work was supported by the National Natural Science Foundation of China (61803259, 61833012) and Shanghai Jiao Tong University, China Scientific and Technological Innovation Funds. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Christoforos Hadjicostis under the direction of Editor André L. Tits.

* Corresponding author.
*E-mail addresses:* liuyang@nyu.edu (Y. Liu), zhaocongl@sjtu.edu.cn (Z. Liu), yinxiang@sjtu.edu.cn (X. Yin), syli@sjtu.edu.cn (S. Li).

https://doi.org/10.1016/j.automatica.2020.109291
0005-1098/

future observation for information smoothing. In Keroglou and Hadjicostis (2017), Zhao, Shu, Lin, and Zhang (2019), detectability was investigated in the stochastic setting by considering the transition probability of the system; corresponding stochastic notions of detectability capturing the probability of state detection were also provided. In Shu and Lin (2013b), Yin and Lafortune (2016), the detectability enforcement problem was studied, where the goal is to design a maximally-permissive supervisor such that the controlled system is detectable. Recently, several new types of detectability are also proposed in the literature for different detection requirements, e.g., $K$-detectability (Hadjicostis & Seatzu, 2016) and trajectory detectability (Yin, Li, & Wang, 2018). Detectability has also been extended to different system models, including nondeterministic systems (Han, Chen, & Zhao, 2017; Zhang & Zamani, 2017), Petri net systems (Masopust & Yin, 2019b; Tong & Lan, 2019; Zhang & Giua, 2018), networked systems (Alves & Basilio, 2019; Sasi & Lin, 2018) and modular systems (Masopust & Yin, 2019a; Yin & Lafortune, 2017b).

In some applications, we may not be interested in knowing the current-state of the system. Instead, we may be interested in detecting the *previous state* of the system with some information delays such as the initial-state detection problem and the delayed detection problem. For example, the concept of I-detectability was introduced by Shu and Lin (2013c). More recently, the initial-state detection problem has also been studied in the stochastic setting (Yin, 2017). In general, one may be interested in detecting the state of the system at an *arbitrary instant* possibly with information delays. This is referred to as the delayed-state estimation problem in Shu and Lin (2013a), Zhou, Shu, and Lin (2018), where the notion of delayed detectability (or $(k_1, k_2)$-detectability) was proposed. Specifically, delayed detectability requires that the state of the system after $k_1$ observations can always be detected within another $k_2$ observation delay. A verification algorithm for delayed detectability was also provided in Shu and Lin (2013a).

In this paper, we revisit the verification delayed detectability as defined in Shu and Lin (2013a). The main contributions of this paper are as follows: We propose a new structure called the *two-way verifier* (TW-verifier) for the purpose of verifying delayed detectability. The TW-verifier essentially composes the standard verifier (Jiang, Huang, Chandra, & Kumar, 2001; Yoo & Lafortune, 2002) and the proposed reversed verifier (Liu, Yin, & Li, 2019) in an asynchronous manner to capture all possible delayed information. Based on the TW-verifier, we then also provide an improved approach for the verification of delayed detectability, which reduces the complexity from $O((k_1+k_2)|\Sigma||X|^6)$ in Shu and Lin (2013a) to $O(|\Sigma||X|^4)$.[1] Our constructions of the reversed verifier and the TW-verifier are novel that combine the advantages of both the verifier structure and the reversed dynamics of the system, which yield better verification complexity compared with all existing algorithms.

## 2. Preliminaries

### 2.1. System model

Let $\Sigma$ be a finite set of events. A string $s=\sigma_1\ldots\sigma_n$ is a finite sequence of events. We denote by $|s|$ the length of the string $s$ with $|\varepsilon|=0$, where $\varepsilon$ is the empty string. We denote by $\Sigma^*$ the set of all strings over $\Sigma$ including the empty string $\varepsilon$. A language $L\subseteq\Sigma^*$ is a set of strings. We define $\Sigma_\varepsilon := \Sigma\cup\{\varepsilon\}$.

[1] Similar complexity reductions are also reported by Zhang and Giua (2019) very recently during the revision of this paper by a different approach.

We consider DES modeled as non-deterministic finite-state automaton (NFA)

$$G=(X,\Sigma,\delta,X_0) \tag{1}$$

where $X$ is a finite set of states, $\Sigma$ is a finite set of events, $\delta: X\times\Sigma\to 2^X$ is a non-deterministic (partial) transition function and $X_0\subseteq X$ is a set of initial-states. For any $x,x'\in X,\sigma\in\Sigma$, $x'\in\delta(x,\sigma)$ implies that there exists a transition from $x$ to $x'$ labeled with $\sigma$. Specifically, we define $\delta(x,\varepsilon)=x$ (here we also use $\varepsilon$ to denote the silent event for simplicity). Function $\delta$ is also extended to $\delta: X\times\Sigma^*\to 2^X$ recursively as follows: for any $s\in\Sigma^*$ and $\sigma\in\Sigma$, we have $\delta(x,s\sigma)=\cup_{x'\in\delta(x,s)}\delta(x',\sigma)$. For each state $x$, we denote by $\mathcal{L}(G,x)$ the set of strings generated by $G$ from $x$, i.e, $\mathcal{L}(G,x)=\{s\in\Sigma^*:\delta(x,s)!\}$, where "!" stands for "is defined". Therefore, $\mathcal{L}(G)=\cup_{x_0\in X_0}\mathcal{L}(G,x_0)$ is the language generated by the system.

In many applications, the occurrence of event cannot be perfectly observed. To capture the imperfect observation, we assume that the event set is partitioned as

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

where $\Sigma_o$ is the set of observable events and $\Sigma_{uo}$ is the set of unobservable events. The natural projection $P:\Sigma^*\to\Sigma_o^*$ is defined recursively by: $\forall s\in\Sigma^*,\sigma\in\Sigma$

$$P(\varepsilon)=\varepsilon,\quad P(s\sigma)=\begin{cases}P(s)\sigma & \text{if } \sigma\in\Sigma_o\\ P(s) & \text{if } \sigma\in\Sigma_{uo}\end{cases}$$

The natural projection is also extended to $P: 2^{\Sigma^*}\to 2^{\Sigma_o^*}$ by: $\forall L\subseteq\Sigma^*: P(L)=\{P(s)\in\Sigma_o^*: s\in L\}$.

Finally, we make the following standard assumptions in the analysis of partially-observed DES:

- A1 System $G$ is deadlock-free, i.e., $\forall x\in X,\exists\sigma\in\Sigma:\delta(x,\sigma)!$;
- A2 System $G$ does not contain unobservable cycle, i.e., $\forall x\in X,\forall s\in\Sigma_{uo}^*$ such that $|s|\geq 1: x\notin\delta(x,s)$.

### 2.2. Delayed detectability

In Shu and Lin (2013a), delayed detectability has been proposed in order to answer the following question: after observing more than $k_1$ observable events, can we always determine the state of the system within at most $k_2$ steps of delays? To formally define delayed detectability, for any $\alpha\beta\in P(\mathcal{L}(G))$, we define

$$\hat{X}(\alpha\mid\alpha\beta)=\left\{x\in X: \begin{array}{c}\exists x_0\in X_0,\ wv\in\mathcal{L}(G,x_0)\text{ s.t.}\\ P(w)=\alpha,\ P(wv)=\alpha\beta,\ x\in\delta(x_0,w)\end{array}\right\} \tag{2}$$

as the *delayed state estimate* of the system at instant $\alpha$ upon the occurrence of $\alpha\beta$. Then, $(k_1,k_2)$-detectability can be defined as follows:

**Definition 2.1** (*$(k_1,k_2)$-detectability*)**.** A DES $G$ is said to be *$(k_1,k_2)$-detectable* if we can determine the state of the system within $k_2$ steps of delays after observing $k_1$ steps of observable events, that is, for any $\alpha\beta\in P(\mathcal{L}(G))$,

$$[|\alpha|\geq k_1\wedge|\beta|\geq k_2]\Rightarrow|\hat{X}(\alpha\mid\alpha\beta)|=1. \tag{3}$$

We illustrate $(k_1,k_2)$-detectability by the following example.

**Example 1.** Let us consider the system $G$ shown in Fig. 1, where we have $X_0=\{0,1\}$ and $\Sigma_o=\{a,b\}$. Note that both states 3 and 4 can be reached after observing $a$, and for any $n\geq 1$, observable string $b^n$ is also able to occur from both states. So we have $\{3,4\}\subseteq\hat{X}(a\mid ab^2)$. Therefore, by Definition 2.1, we know that system $G$ is not $(1,2)$-detectable.

## 3. Verification of $(k_1, k_2)$-detectability

In this section, we investigate the verification of $(k_1, k_2)$-detectability. In Shu and Lin (2013a), the authors have provided an algorithm for the verification of $(k_1, k_2)$-detectability. The complexity of the algorithm proposed in Shu and Lin (2013a) is $O((k_1 + k_2)|\Sigma||X|^6)$. In this section, we propose an improved algorithm for the verification of $(k_1, k_2)$-detectability using a more compact information structure called the *two-way verifier*.

In Yin and Lafortune (2017a), the authors have proposed the notion of *Two-Way Observer* (TW-observer) for the verification of infinite-step and $K$-step opacity. However, the size of the TW-observer is exponential in the size of the system. Motivated by the TW-observer but by incorporating the feature of delayed detectability, we define the structure of *Two-Way Verifier* (TW-verifier), which asynchronously composes the standard verifier and the *reversed* verifier proposed in Liu et al. (2019), as follows:

**Definition 3.1.** The *TW-verifier* of $G$ is an NFA

$$V_{TW}(G) = (Q_{TW}, E_{TW}, f_{TW}, Q_{TW,0}) \tag{4}$$

where

- $Q_{TW} \subseteq X \times X \times X \times X$ is a set of states;
- $E_{TW} \subseteq (\Sigma_\varepsilon \times \Sigma_\varepsilon \times \{\varepsilon\} \times \{\varepsilon\}) \cup (\{\varepsilon\} \times \{\varepsilon\} \times \Sigma_\varepsilon \times \Sigma_\varepsilon) \setminus \{(\varepsilon, \varepsilon, \varepsilon, \varepsilon)\}$ is a set of events;
- $Q_{TW,0} = X_0 \times X_0 \times X \times X$ is a set of initial-states;
- $f_{TW} : Q_{TW} \times E_{TW} \to 2^{Q_{TW}}$ is the non-deterministic transition function defined as follows: For any $q = (x_1, x_2, x_3, x_4) \in Q_{TW}, \sigma \in \Sigma$, the following transitions are defined:

(a) If $\sigma \in \Sigma_o$, then

$$f_{TW}((x_1, x_2, x_3, x_4), (\sigma, \sigma, \varepsilon, \varepsilon)) = \{(x'_1, x'_2, x_3, x_4) \in Q_{TW} : x'_1 \in \delta(x_1, \sigma) \wedge x'_2 \in \delta(x_2, \sigma)\} \tag{5}$$

$$f_{TW}((x_1, x_2, x_3, x_4), (\varepsilon, \varepsilon, \sigma, \sigma)) = \{(x_1, x_2, x'_3, x'_4) \in Q_{TW} : x_3 \in \delta(x'_3, \sigma) \wedge x_4 \in \delta(x'_4, \sigma)\} \tag{6}$$

(b) If $\sigma \in \Sigma_{uo}$, then

$$f_{TW}((x_1, x_2, x_3, x_4), (\sigma, \varepsilon, \varepsilon, \varepsilon)) = \{(x'_1, x_2, x_3, x_4) \in Q_{TW} : x'_1 \in \delta(x_1, \sigma)\} \tag{7}$$

$$f_{TW}((x_1, x_2, x_3, x_4), (\varepsilon, \sigma, \varepsilon, \varepsilon)) = \{(x_1, x'_2, x_3, x_4) \in Q_{TW} : x'_2 \in \delta(x_2, \sigma)\} \tag{8}$$

$$f_{TW}((x_1, x_2, x_3, x_4), (\varepsilon, \varepsilon, \sigma, \varepsilon)) = \{(x_1, x_2, x'_3, x_4) \in Q_{TW} : x_3 \in \delta(x'_3, \sigma)\} \tag{9}$$

$$f_{TW}((x_1, x_2, x_3, x_4), (\varepsilon, \varepsilon, \varepsilon, \sigma)) = \{(x_1, x_2, x_3, x'_4) \in Q_{TW} : x_4 \in \delta(x'_4, \sigma)\} \tag{10}$$

**Remark 1.**
The TW-verifier is motivated by the standard verifier (or twin-machine) construction that has been widely used in the verification of partially-observed DES. In the standard verifier, the structure tracks a pair of strings having the same observation. However, the TW-verifier tracks two pairs of strings, i.e., we consider the first two components as a pair and the last two components as another pair. The common property of the two pairs is that strings in each part have the same projection. However, the first pair of strings follows the *original* dynamics of the system while the second pair of strings follows the *reversed* dynamics of the original system. This is why we call it "two-way" since the information tracking directions in the first and the second pair are opposite.

Before we propose the algorithm for the verification of the $(k_1, k_2)$-detectability, we first show some properties of the TW-verifier. We observe that in the proposed TW-verifier, the first component is essentially the standard verifier (see, e.g., Jiang et al., 2001; Yoo & Lafortune, 2002), while the second component is the reversed verifier proposed in our recent work (Liu et al., 2019), and these two parts are not synchronized. Therefore, similar to the propositions of the reversed verifier in Liu et al. (2019), we have the following properties of $V_{TW}(G)$ immediately.

**Proposition 1.** *For any sequence*

$$(x_1^0, x_2^0, x_3^0, x_4^0) \xrightarrow[TW]{(\sigma_1^1, \sigma_2^1, \sigma_3^1, \sigma_4^1)} (x_1^1, x_2^1, x_3^1, x_4^1) \xrightarrow[TW]{(\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2)} \cdots \xrightarrow[TW]{(\sigma_1^n, \sigma_2^n, \sigma_3^n, \sigma_4^n)} (x_1^n, x_2^n, x_3^n, x_4^n) \tag{11}$$

*in $V_{TW}(G)$, we have*

*(i) $\forall i = 1, 2 : \sigma_i^1 \sigma_i^2 \ldots \sigma_i^n \in \mathcal{L}(G, x_i^0)$; and*
*(ii) $\forall i = 3, 4 : \sigma_1^n \sigma_1^{n-1} \ldots \sigma_1^1 \in \mathcal{L}(G, x_i^n)$; and*
*(iii) $P(\sigma_1^1 \sigma_1^2 \ldots \sigma_1^n) = P(\sigma_2^1 \sigma_2^2 \ldots \sigma_2^n)$; and*
*(iv) $P(\sigma_3^n \sigma_3^{n-1} \ldots \sigma_3^1) = P(\sigma_4^n \sigma_4^{n-1} \ldots \sigma_4^1)$.*

**Proposition 2.** *For any two initial states $x_{0,1}, x_{0,2} \in X_0$, two states $x_3, x_4 \in X$, and for strings $s_1 \in \mathcal{L}(G, x_{0,1}), s_2 \in \mathcal{L}(G, x_{0,2}), s_3 \in \mathcal{L}(G, x_3), s_4 \in \mathcal{L}(G, x_4)$ such that $P(s_1) = P(s_2)$ and $P(s_3) = P(s_4)$, there must exist a sequence*

$$\begin{aligned}(x_1^0, x_2^0, x_3^0, x_4^0) &\xrightarrow[TW]{(\sigma_1^1, \sigma_2^1, \varepsilon, \varepsilon)} (x_1^1, x_2^1, x_3^0, x_4^0) \xrightarrow[TW]{(\sigma_1^2, \sigma_2^2, \varepsilon, \varepsilon)} \cdots \\ &\xrightarrow[TW]{(\sigma_1^n, \sigma_2^n, \varepsilon, \varepsilon)} (x_1^n, x_2^n, x_3^0, x_4^0) \\ &\xrightarrow[TW]{(\varepsilon, \varepsilon, \sigma_3^1, \sigma_4^1)} (x_1^n, x_2^n, x_3^1, x_4^1) \xrightarrow[TW]{(\varepsilon, \varepsilon, \sigma_3^2, \sigma_4^2)} \cdots \\ &\xrightarrow[TW]{(\varepsilon, \varepsilon, \sigma_3^m, \sigma_4^m)} (x_1^n, x_2^n, x_3^m, x_4^m)\end{aligned} \tag{12}$$

*in $V_{TW}(G)$ such that*

*(i) $(x_1^0, x_2^0, x_3^m, x_4^m) = (x_{0,1}, x_{0,2}, x_3, x_4)$; and*
*(ii) For any $i = 1, 2$, $s_i = \sigma_i^1 \sigma_i^2 \ldots \sigma_i^n$; and*
*(iii) For any $i = 3, 4$, $s_i = \sigma_i^m \sigma_i^{m-1} \ldots \sigma_i^1$.*

Next, we show that, for any state $(x_1, x_2, x_3, x_4)$ reached in $V_{TW}(G)$, if $\{x_1, x_2\} \cap \{x_3, x_4\} \neq \emptyset$, the first component of strings and the second component of strings corresponding to the above state can be "connected" at a state in $\{x_1, x_2\} \cap \{x_3, x_4\}$.

**Proposition 3.** *Let $s = (s_1, s_2, s_3, s_4) \in \mathcal{L}(V_{TW}(G))$ be a string in the TW-verifier and $(x_1, x_2, x_3, x_4)$ be a state reached by $s$. Suppose $P(s_1) = P(s_2) = \alpha$ and $P(s_3^R) = P(s_4^R) = \beta$. Then we have:*

*(i) if $\{x_1, x_2\} \cap \{x_3, x_4\} \neq \emptyset$, then $\alpha\beta \in P(\mathcal{L}(G))$; and*
*(ii) for any $x \in \{x_1, x_2\} \cap \{x_3, x_4\}$, there exist an initial state $x_0 \in X_0$ and a string $wv \in \mathcal{L}(G)$ such that $P(w) = \alpha, P(v) = \beta, x \in \delta(x_0, w)$ and $\delta(x, v)!$.*

**Proof.** By the construction of $V_{TW}(G)$, we know that

$$\begin{aligned}&\exists x_{0,1} \in X_0 : x_1 \in \delta(x_{0,1}, s_1)\\ &\exists x_{0,2} \in X_0 : x_2 \in \delta(x_{0,2}, s_2)\\ &\exists x'_3 \in X : x'_3 \in \delta(x_3, s_3)\\ &\exists x'_4 \in X : x'_4 \in \delta(x_4, s_4)\end{aligned}$$

Without loss of generality, we suppose that $x_1 = x_3$ so that $x_1 \in \{x_1, x_2\} \cap \{x_3, x_4\}$. Then we have $x_1 \in \delta(x_{0,1}, s_1)$ and $\delta(x_1, s_3)!$. Therefore, we can know that $\delta(x_{0,1}, s_1 s_3)!$ and $P(s_1 s_3) = \alpha\beta$. This proves both (i) and (ii). □

Before we introduce the main result for the verification of $(k_1, k_2)$-detectability, we first define some new notations. Let $V_{TW}(G)$ be the TW-verifier. We denote by $\text{Reach}_{k_1,k_2}(Q_{TW,0})$ the set of states that can be reached by a string whose first part is longer than $k_1$ and second part is longer than $k_2$, that is:

$$\text{Reach}_{k_1,k_2}(Q_{TW,0}) = \left\{ q \in Q_{TW} : \begin{array}{c} \exists s = (s_1, s_2, s_3, s_4) \in \mathcal{L}(V_{TW}(G)), \\ \exists q_0 \in Q_{TW,0} \text{ s.t. } q \in f_{TW}(q_0, s) \wedge \\ |P(s_1)| \geq k_1 \wedge |P(s_3)| \geq k_2 \end{array} \right\}. \tag{13}$$

The following theorem provides an approach for the verification of $(k_1, k_2)$-detectability.

**Theorem 1.** *System $G$ is $(k_1, k_2)$-detectable if and only if for any state $(x_1, x_2, x_3, x_4)$ in $\text{Reach}_{k_1,k_2}(Q_{TW,0})$ such that $(x_1, x_2) = (x_3, x_4)$, we have $x_1 = x_2$ and $x_3 = x_4$.*

**Proof.** ($\Rightarrow$) By contraposition. Suppose that there exists a state $(x_1, x_2, x_1, x_2) \in \text{Reach}_{k_1,k_2}(Q_{TW,0})$ such that $(x_1, x_2) = (x_3, x_4)$ but $x_1 \neq x_2$. By the construction of $\text{Reach}_{k_1,k_2}(Q_{TW,0})$ and Proposition 3, we know that there exists a string $(s_1, s_2, s_3, s_4) \in \mathcal{L}(V_{TW}(G))$ such that $P(s_1) = P(s_2) = \alpha$, $P(s_3^R) = P(s_4^R) = \beta$, $|\alpha| \geq k_1$ and $|\beta| \geq k_2$, and two initial states $x_{0,1}, x_{0,2} \in X_0$ such that $x_1 \in \delta(x_{0,1}, s_1)$, $x_2 \in \delta(x_{0,2}, s_2)$, $\delta(x_1, s_3)!$ and $\delta(x_2, s_4)!$. By the definition of delayed state estimate, we know that $\{x_1, x_2\} \subseteq \hat{X}(\alpha \mid \alpha\beta)$. Therefore, $x_1 \neq x_2$ implies that $|\hat{X}(\alpha \mid \alpha\beta)| \geq 2$, which means that system $G$ is not $(k_1, k_2)$-detectable.

($\Leftarrow$) By contraposition. Suppose that system $G$ is not $(k_1, k_2)$-detectable, which means that there exists $\alpha\beta \in P(\mathcal{L}(G))$ such that $|\alpha| \geq k_1$, $|\beta| \geq k_2$ and $|\hat{X}(\alpha \mid \alpha\beta)| > 1$. Let $x_1, x_2 \in X$ be two distinct states in $\hat{X}(\alpha \mid \alpha\beta)$. This implies that there exist two initial states $x_{0,1}, x_{0,2} \in X_0$ and two strings $s_1, s_2$ such that $P(s_1) = P(s_2) = \alpha$, $x_1 \in \delta(x_{0,1}, s_1)$ and $x_2 \in \delta(x_{0,2}, s_2)$. Also, there exist another two strings $s_3 \in \mathcal{L}(G, x_1)$ and $s_4 \in \mathcal{L}(G, x_2)$ such that $P(s_3) = P(s_4) = \beta$. By Proposition 2, we know that there exists a sequence

$$\begin{aligned} (x_1^0, x_2^0, x_3^0, x_4^0) &\xrightarrow[TW]{(\sigma_1^1, \sigma_2^1, \varepsilon, \varepsilon)} (x_1^1, x_2^1, x_3^0, x_4^0) \xrightarrow[TW]{(\sigma_1^2, \sigma_2^2, \varepsilon, \varepsilon)} \cdots \\ &\xrightarrow[TW]{(\sigma_1^n, \sigma_2^n, \varepsilon, \varepsilon)} (x_1^n, x_2^n, x_3^0, x_4^0) \\ &\xrightarrow[TW]{(\varepsilon, \varepsilon, \sigma_3^1, \sigma_4^1)} (x_1^n, x_2^n, x_3^1, x_4^1) \xrightarrow[TW]{(\varepsilon, \varepsilon, \sigma_3^2, \sigma_4^2)} \cdots \\ &\xrightarrow[TW]{(\varepsilon, \varepsilon, \sigma_3^m, \sigma_4^m)} (x_1^n, x_2^n, x_3^m, x_4^m) \end{aligned} \tag{14}$$

in $V_{TW}(G)$ such that $(x_1^0, x_2^0) = (x_{0,1}, x_{0,2})$ and $(x_1^n, x_2^n, x_3^m, x_4^m) = (x_1, x_2, x_1, x_2)$. Besides, we have that for any $i = 1, 2$, $s_i = \sigma_i^1 \sigma_i^2 \ldots \sigma_i^n$ and for any $i = 3, 4$, $s_i = \sigma_i^m \sigma_i^{m-1} \ldots \sigma_i^1$. Since $P(s_1) = P(s_2) = |\alpha| \geq k_1$ and $P(s_3) = P(s_4) = |\beta| \geq k_2$, we know that $(x_1, x_2, x_1, x_2) \in \text{Reach}_{k_1,k_2}(Q_{TW,0})$ and $x_1 \neq x_2$. □

Next, we illustrate how to verify delayed detectability using Theorem 1.

**Example 2.** Let us consider system $G$ shown in Fig. 1. We have discussed in Example 1 that system $G$ is not (1,1)-detectable; here we use Theorem 1 to verify this result. First, we construct the TW-verifier $V_{TW}(G)$. For the sake of simplicity, we only consider part of $V_{TW}(G)$ in Fig. 2, which is enough to disprove delayed detectability. In Fig. 2, the state $(0, 1, 3, 4)$ is an initial state in $V_{TW}(G)$ which can reach state $(3, 4, 3, 4)$ via event $(a, a, \varepsilon, \varepsilon)$. Then we see that state $(3, 4, 3, 4)$ can reach itself via event $(\varepsilon, \varepsilon, b, b)$. Therefore, we have $(3, 4, 3, 4) \in \text{Reach}_{1,1}(V_{TW}(G))$. Since $(3, 4) = (3, 4)$ but $3 \neq 4$, by Theorem 1, we can conclude that $G$ is not (1,1)-detectable.

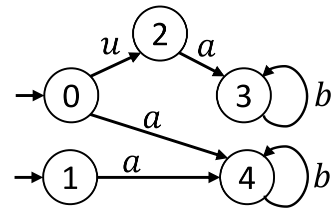


**Fig. 1.** System $G$ with $\Sigma_o = \{a, b\}$.

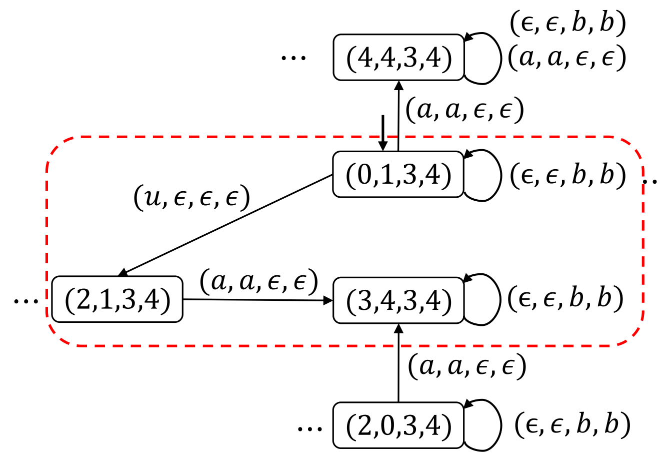


**Fig. 2.** Part of the TW-verifier $V_{TW}(G)$ for system $G$ shown in Fig. 1.

**Remark 2.** We now discuss the complexity of verifying delayed detectability using Theorem 1. First, we need to construct the TW-verifier, which contains at most $|X|^4$ states and $4|\Sigma||X|^4$ transitions. Then we need to compute state set $\text{Reach}_{k_1,k_2}(Q_{TW,0})$ in $V_{TW}(G)$. Note that for any string $(s_1, s_2, s_3, s_4) \in \mathcal{L}(V_{TW}(G))$, the first part $(s_1, s_2)$ and the second part $(s_3, s_4)$ are independent. Therefore, we can first perform a $k_1$-level Breadth First Search (BFS) in the first part (first two state/event components) of $V_{TW}(G)$ in order to determine states that can be reached via more than $k_1$ observable events in $s_1$ or $s_2$. Similarly, we perform a $k_2$-level BFS in the second part (last two state/event components) of $V_{TW}(G)$ in order to determine states that can be reached via more than $k_2$ observable events $s_3$ and $s_4$. The worst-case complexity of BFS is $O(V + E)$ where $V$ is the number of vertices and $E$ is the number of edges, which is linear no matter how large $k_1$ and $k_2$ are. Therefore, the overall complexity of verifying delayed detectability using Theorem 1 is $O(|\Sigma||X|^4)$, which improves the algorithm in Shu and Lin (2013a) whose complexity is $O((k_1 + k_2)|\Sigma||X|^6)$.

## 4. Conclusion

In this paper, we revisited the verification of delayed detectability in the context of partially-observed DES. To this end, a new information structure called the TW-verifier was proposed that handles the delayed information effectively. We then provided an improved approach for the verification of delayed detectability by using the TW-verifier. The complexity of the proposed algorithm is $O(|\Sigma||X|^4)$ compared with complexity $O((k_1 + k_2)|\Sigma||X|^6)$ of the previous algorithm.

## References


Alves, M. V. S., & Basilio, J. C. (2019). State estimation and detectability of networked discrete event systems with multi-channel communication networks. In *American control conference*.

Ammour, R., Leclercq, E., Sanlaville, E., & Lefebvre, D. (2017). State estimation of discrete event systems for RUL prediction issue. *International Journal of Productions Research*, *55*(23), 7040–7057.

Hadjicostis, C. N., & Seatzu, C. (2016). $K$-Detectability in discrete event systems. In *55th IEEE conf. decision and control* (pp. 420–425).

Han, X., Chen, Z., & Zhao, J. (2017). Matrix approach to detectability of discrete event systems under partial observation. In *IEEE conf. automation science and engineering* (pp. 187–192).

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. N. (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.

Lin, F., & Wonham, W. M. (1988). On observability of discrete-event systems. *Information Sciences*, *44*(3), 173–198.

Liu, Z., Yin, X., & Li, S. (2019). Improved approaches for verifying I-detectability of discrete-event systems. In *12th Asian control conference* (pp. 248–253.

Ma, Z., Li, Z., & Giua, A. (2019). Marking estimation in a class of time labelled petri nets. *IEEE Transactions on Automatic Control*.

Masopust, T. (2018). Complexity of deciding detectability in discrete event systems. *Automatica*, *93*, 257–261.

Masopust, T., & Yin, X. (2019a). Complexity of detectability, opacity and A-diagnosability for modular discrete event systems. *Automatica*, *101*, 290–295.

Masopust, T., & Yin, X. (2019b). Deciding detectability for labeled Petri nets. *Automatica*, *104*, 238–241.

Özveren, C. M., & Willsky, A. S. (1990). Observability of discrete event dynamic systems. *IEEE Transactions on Automatic Control*, *35*(7), 797–806.

Ramadge, P. (1986). Observability of discrete event systems. In *25th IEEE conf. decision and control, no. 25* (pp. 1108–1112).

Sasi, Y., & Lin, F. (2018). Detectability of networked discrete event systems. *Discrete Event Dynamic Systems*, *28*(3), 449–470.

Shu, S., & Lin, F. (2011). Generalized detectability for discrete event systems. *Systems & Control Letters*, *60*(5), 310–317.

Shu, S., & Lin, F. (2013a). Delayed detectability of discrete event systems. *IEEE Transactions on Automatic Control*, *58*(4), 862–875.

Shu, S., & Lin, F. (2013b). Enforcing detectability in controlled discrete event systems. *IEEE Transactions on Automatic Control*, *58*(8), 2125–2130.

Shu, S., & Lin, F. (2013c). 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.

Tong, Y., & Lan, H. (2019). verification of detectability in labeled Petri nets. *American control conference*.

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

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

Yin, X., & Lafortune, S. (2017a). A new approach for the verification of infinite-step and K-step opacity using two-way observers. *Automatica*, *80*, 162–171.

Yin, X., & Lafortune, S. (2017b). Verification complexity of a class of observational properties for modular discrete events systems. *Automatica*, *83*, 199–205.

Yin, X., Li, Z., & Wang, W. (2018). Trajectory detectability of discrete-event systems. *Systems & Control Letters*, *119*, 101–107.

Yoo, T.-S., & 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., & Giua, A. (2018). Weak (approximate) detectability of labeled Petri net systems with inhibitor arcs. In *14th international workshop on discrete event systems* (pp. 167–171).

Zhang, K., & Giua, A. (2019). Revisiting delayed strong detectability of discrete-event systems. arXiv:1910.13768.

Zhang, K., & Zamani, M. (2017). Detectability of nondeterministic finite transition systems. *IFAC-PapersOnLine*, *50*(1), 9272–9277.

Zhao, P., Shu, S., Lin, F., & Zhang, B. (2019). Detectability measure for state estimation of discrete event systems. *IEEE Transactions on Automatic Control*, *64*(1), 433–439.

Zhou, L., Shu, S., & Lin, F. (2018). N-$(k_1, k_2)$-detectability of discrete event systems under nondeterministic observations. In *American control conference* (pp. 294–299).