Systems & Control Letters 119 (2018) 101–107

Contents lists available at ScienceDirect

Systems & Control Letters

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

# Trajectory detectability of discrete-event systems

Xiang Yin [a,b,*], Zhaojian Li [c], Weilin Wang [d]

[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 Mechanical Engineering, Michigan State University, East Lansing, MI 48824, USA
[d] Faculty of Engineering, Monash University, Clayton, VIC 3800, Australia

ARTICLE INFO

*Article history:*
Received 19 June 2017
Received in revised form 14 July 2018
Accepted 16 July 2018

*Keywords:*
Discrete-event systems
Trajectory estimation
Detectability

ABSTRACT

We investigate the trajectory estimation problem for partially-observed discrete event systems. In some applications, only knowing the current state of the system may be insufficient, and knowing which trajectory the system takes to reach the current state could be important. This requires more precise knowledge about the system. In this paper, a language-based framework is proposed in order to tackle this problem. Two new notions of detectability, called trajectory detectability and periodic trajectory detectability, are proposed to capture different requirements in the aforementioned trajectory estimation problem. Effective verification algorithms are also provided. Our results extend the theory on detectability of discrete event systems from state estimation problem to trajectory estimation problem.


## 1. Introduction

System estimation is one of the central problems in systems and control theory. In many applications, we do not have full access to the system's internal state and need to perform state estimation. The state estimation problem becomes particularly important when one wants to make decisions based on the limited system information. In this paper, we investigate the state estimation problem for partially-observed Discrete-Event Systems (DES).

The problem of state estimation has drawn considerable attentions in the DES literature due to its importance; see, e.g., [1–6]. This problem was initiated in [1,2], where the notion of observability was defined.[1] Recently, the state estimation problem has been studied more systematically in the framework of detectability; see, e.g., [4,5,8–14]. Particularly, in [4], the authors define four types of detectability in order to capture different requirements in practice. These notions of detectability have been further generalized by [5,8]. For example, [8] defines a generalized detectability based on the state disambiguation problem [3,6]. Detectability has also been studied in the framework of stochastic DES by [9,11]. When the original system is not detectable, several approaches have been proposed in order to enforce detectability, e.g., by sensor activations [15,16] and by supervisory control [17,18]. State estimation problem has also been investigated in the context of colored graph [19].

All of the aforementioned works on detectability are *state-based*. Namely, one wants to estimate the current state of the system based on a given model. However, in some applications, knowing the current state of the system is not sufficient. For example, in the application of location-based services (LBS) [20], a DES is usually used to represent the connective of a region and each state in it corresponds to a location. Sometimes, however, simply knowing the current location may not be sufficient, for instance, if we want to know by which path this location is reached. Therefore, instead of estimating the state of the system, one may also be interested in estimating the *trajectory* of the system.

In this paper, we systematically study the trajectory estimation problem in the context of partially-observed DES. Specifically, this paper has the following contributions. First, we define the notions of trajectory detectability and periodic trajectory detectability. These two notions provide the conditions for determining *a priori* if the trajectory of a given system can be determined after a bounded delay or be determined periodically. Second, for regular languages, i.e., languages that can be marked by finite-state automata, we provide effective algorithms to verify these two conditions. In particular, the verification algorithm for trajectory detectability requires polynomial-times using a twin-machine-like construction. On the other hand, the algorithm for verifying periodic trajectory detectability requires exponential complexity.

Note that, although the study of trajectory detectability is motivated by state detectability [4], their verifications are quite different. In general, we can always refine an automaton by expanding

* Corresponding author at: Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China.
*E-mail address:* yinxiang@sjtu.edu.cn (X. Yin).

1 The notion of observability was also used in [7] as a condition under which a partial-observation supervisor can correctly make control decisions.

https://doi.org/10.1016/j.sysconle.2018.07.008
0167-6911/

its state space such that each state carries more information. However, the domain of language is infinite and one may not always be able to use finite states to precisely capture the (infinite) trajectory information. Our framework is fully language-based and it does not depend on the automaton realizing the language. Moreover, we show that infinite language information can still be effectively verified by using its underlying automaton. In particular, we only exploit the standard twin-machine construction [21,22], which is different from the detector construction proposed in [8]. We show that the phenomenon of *information merge* plays an important role in the trajectory estimation problem. This issue also does not exist in the state-based framework for detectability.

We also would like to remark that our paper is not the first one investigating the trajectory estimation problem in DES. In [23], the authors investigate a similar problem and the notion of *invertibility* is proposed. Our work is different from [23] due to the following reasons. First, we systematically investigate both trajectory detectability and periodic trajectory detectability; both of these two notions are different from invertibility. Specially, invertibility only requires to recover the last $n$ events but detectability requires to recover the precise trajectory executed by the system. Second, we provide a language-based framework for studying this problem, while the result in [23] is state dependent. Finally, invertibility is defined only for prefix-closed languages, while trajectory detectability is defined for non-prefix-closed languages.

## 2. Preliminaries

Let $\Sigma$ be a finite set of events. A string $s = \sigma_1 \ldots \sigma_n$ is finite sequence of events and we denote by $|s|$ the length of $s$. We use $\epsilon$ to denote the empty string with $|\epsilon| = 0$. We denote by $\Sigma^*$ the set of all strings including $\epsilon$. A language $L \subseteq \Sigma^*$ is a set of strings. The prefix-closure of language $L$ is defined as $\overline{L} := \{w \in \Sigma^* : \exists v \in \Sigma^* \text{ s.t. } wv \in L\}$. We say that $L$ is prefix-closed if $\overline{L} = L$. We denote by $L/s$ the post-language of $L$ after string $s$, i.e., $L/s := \{t \in \Sigma^* : st \in L\}$. We denote by $Card[L]$ the cardinality of $L$, which is the number of strings in $L$. For two strings $s, t \in \Sigma^*$, we write $s \leq t$ if $s \in \overline{\{t\}}$.

A deterministic finite-state automaton (DFA) is a 5-tuple $G = (Q, \Sigma, \delta, q_0, Q_m)$, where $Q$ is the finite set of states, $\Sigma$ is the finite set of events, $q_0$ is the initial state, $Q_m$ is the set of marked states and $\delta : Q \times \Sigma \to Q$ is the partial transition function, where $\delta(q, \sigma) = q'$ means that there exists a transition labeled with event $\sigma$ from state $q$ to state $q'$. The transition function is also extended to $Q \times \Sigma^*$ in the usual manner; see, e.g., [24]. We denote by $\mathcal{L}(G)$ the language *generated* by $G$, i.e., $\mathcal{L}(G) = \{s \in \Sigma^* : \delta(q_0, s)!\}$, where ! means "is defined". We denoted by $\mathcal{L}_m(G)$ the language *marked* by $G$, i.e., $\mathcal{L}(G) = \{s \in \Sigma^* : \delta(q_0, s) \in Q_m\}$. We say that a language $L \subseteq \Sigma^*$ is *regular* if there exists a DFA $G$ such that $\mathcal{L}_m(G) = L$.

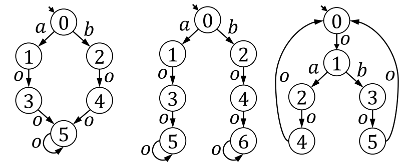
In many cases, the event generated by the system cannot be observed perfectly. Therefore, we assume that the event set $\Sigma$ is partitioned into two disjoint sets $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$, where $\Sigma_o$ is the set of observable events and $\Sigma_{uo}$ denotes the set of unobservable events. Then $P : \Sigma^* \to \Sigma_o^*$ denotes the natural projection that erases event in $\Sigma_{uo}$ from a string; this can be defined by

$$P(\epsilon) = \epsilon \text{ and } P(s\sigma) = \begin{cases} P(s)\sigma & \text{if } \sigma \in \Sigma_o \\ P(s) & \text{if } \sigma \in \Sigma_{uo} \end{cases} \tag{1}$$

The natural projection is also extended to $2^{\Sigma^*}$ by $P(L) = \{s \in \Sigma_o^* : \exists t \in L \text{ s.t. } P(t) = s\}$. We denote by $P^{-1}$ the inverse projection.

Given a DFA $G$ and a set of states $I \subseteq Q$, we denote by $\mathrm{Acc}_G(I)$ the set of states accessible from some state in $I$, i.e.,

$$\mathrm{Acc}_G(I) = \{q \in Q : \exists q' \in I, \exists s \in \Sigma^* \text{ s.t. } \delta(q', s) = q\}. \tag{2}$$



(a) System $G_1$ (b) System $G_2$ (c) System $G_3$

**Fig. 1.** For both $G_1$ and $G_2$: $\Sigma_o = \{o\}$ and $\Sigma_{uo} = \{a, b\}$.

Let $q \in Q$ be a state in $G$. We denote by $\mathrm{In}_G(q)$ the set of events entering $q$, i.e.,

$$\mathrm{In}_G(q) = \{\sigma \in \Sigma : \exists q' \in Q \text{ s.t. } \delta(q', \sigma) = q\}. \tag{3}$$

Finally, let $s \in P(\mathcal{L}(G))$, we denote by $R_G(s)$ the set of states that can be reached by observing $s$, i.e.,

$$R_G(s) = \{q \in Q : \exists t \in \Sigma^* \text{ s.t. } \delta(q_0, t) = q \wedge P(t) = s\}. \tag{4}$$

## 3. State-Based detectability and trajectory-based detectability

Due to measurement uncertainty, one may not always have a perfect knowledge about the current status of the system. In [4], the notion of (strongly) detectability was introduced in order to capture whether or not we can eventually have a perfect knowledge about the system after finite delay. In this paper, we refer to detectability defined in [4] as *state detectability*. First, we recall its definition.

**Definition 1.** A DFA $G = (Q, \Sigma, \delta, q_0, Q_m)$ is said to be *state detectable* w.r.t. $\Sigma_o$ if

$$(\exists n \in \mathbb{N})(\forall s \in L)[|P(s)| \geq n \Rightarrow |R_G(P(s))| = 1] \tag{5}$$

**Example 1.** Let us consider system $G_1$ shown in Fig. 1(a), where $\Sigma_o = \{o\}$. Clearly, this system is state detectable since $R_G(o^n) = \{5\}$ for any $n \geq 2$. However, system $G_2$ shown in Fig. 1(b) is not state detectable. To see this, for any $n \geq 2$, we can find $o^n$ such that $R_G(o^n) = \{5, 6\}$, i.e., we can never determine the current state of the system precisely.

Intuitively, state detectability says that, after a finite delay, we will know exactly the current state of the system and maintain this ability in the future. However, in some applications, this requirement may be too strong. Therefore, in [4], the notion of periodic state detectability was also proposed, which only requires that we can detect the state of the system periodically.

**Definition 2.** A DFA $G = (Q, \Sigma, \delta, q_0, Q_m)$ is said to be *periodically state detectable* w.r.t. $\Sigma_o$ if

$$\begin{aligned} &(\exists n \in \mathbb{N})(\forall s \in L)(\forall t \in L/s : |P(t)| \geq n) \\ &(\exists t' \leq t)[|R_G(P(st'))| = 1] \end{aligned} \tag{6}$$

**Example 2.** Let us consider system $G_3$ shown in Fig. 1(c) with $\Sigma_o = \{o\}$. This system is not state detectable, since we cannot distinguish states 4 and 5 after observing $oo(ooo)^n$ for any $n \geq 0$. However, it is periodically state detectable, since we always know for sure that the current state is 0 after observing $(ooo)^n$ for any $n \geq 0$.

**Remark 1.** Note that the system automaton $G$ considered in [4] has multiple initial states; say $Q_0 \subseteq Q$. In the above definitions, we only consider the case where the initial state is unique. However,