Automatica 83 (2017) 199–205

Contents lists available at ScienceDirect

Automatica

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

Brief paper

Verification complexity of a class of observational properties for modular discrete events systems☆

Xiang Yin [a,1], Stéphane Lafortune [b]

[a] Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China
[b] Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA

ARTICLE INFO

Article history:
Received 11 July 2016
Received in revised form 11 April 2017
Accepted 23 April 2017

Keywords:
Computational complexity
PSPACE-completeness
Modular diagnosis
Discrete event systems

ABSTRACT

A modular discrete event system is modeled by a set of module automata running synchronously. In this paper, we investigate the complexity of the verification problems of three different properties, diagnosability, predictability, and detectability, for partially-observed modular discrete event systems. We first show that deciding diagnosability for modular discrete event systems is PSPACE-complete when the number of modules is unbounded. Then we show that deciding predictability and detectability for modular discrete event systems are both PSPACE-hard problems. These results reveal that in order to verify these properties for the complete system, exploring the state space of the monolithic model may be unavoidable, in the worst case.

1. Introduction

Discrete Event Systems (DES) are widely used in the modeling and analysis of the high-level logical behavior of complex automated systems. An important problem in DES is to verify whether or not a system, or plant, satisfies some property. In practice, the system is composed of several *modules* running synchronously. Finite-state automata are widely used representations for these modules in which case synchronization is captured by common event labels. However, analyzing the global behavior of the entire system is not an easy task, since the size of the monolithic system representation grows exponentially fast (in the worst case) as the number of modules increases. In this paper, we investigate the complexity of several property verification problems for DES with modular representations.

Many different system-theoretic properties have been investigated in the context of DES models. In this paper, we consider three different properties: diagnosability (Sampath, Sengupta, Lafortune, Sinnamohideen, & Teneketzis, 1995), predictability (Genc & Lafortune, 2009), and detectability (Shu, Lin, & Ying, 2007). Diagnosability is related to the fault diagnosis problem and it requires that a fault event be diagnosed unambiguously within a bounded delay after its occurrence. Similarly to diagnosability, predictability requires that an inevitable fault event be predicted unambiguously before its occurrence. Detectability is related to the state estimation problem and it requires that the exact system state be identified after a finite number of observations. When the monolithic model of the system is given, all of these three properties can be verified in polynomial time w.r.t. the size of the monolithic model; see, e.g., Genc and Lafortune (2009), Jiang, Huang, Chandra, and Kumar (2001), Shu et al. (2007) and Yoo and Lafortune (2002).

When the system is modeled by a set of modules running synchronously, computing the monolithic model, whose state space is exponentially large w.r.t. the number of modules, may be infeasible. In order to avoid computing the monolithic model, many different modular approaches have been proposed for different problems; see, e.g., Feng and Wonham (2008), Gummadi, Singh, and Sreenivas (2011), Hill, Cury, de Queiroz, Tilbury, and Lafortune (2010), Komenda, van Schuppen, Gaudin, and Marchand (2008), Leduc, Lawford, and Wonham (2005), Saboori and Hadjicostis (2010), and Schmidt and Cury (2012). In particular, in the context of fault diagnosis, several approaches have been proposed based on exploiting the modular structure of the system; see, e.g., Contant, Lafortune, and Teneketzis (2006), Debouk, Malik, and Brandin (2002), Fabre, Benveniste, Haar, and Jard (2005), Pencolé (2004), Ramírez-Treviño, Ruiz-Beltrán, Rivera-Rangel, and López-Mellado (2007), Ricker and Fabre (2000), Schmidt (2013), Ye and Dague (2010) and Zhou, Kumar, and Sreenivas (2008). However, these

☆ This work was partially supported by NSF grants CCF-1138860 (Expeditions in Computing project ExCAPE: Expeditions in Computer Augmented Program Engineering), CNS-1421122, and CNS-1446298. The material in this paper was not presented at any conference. This work was done when the first author was at the University of Michigan. This paper was recommended for publication in revised form by Associate Editor Christoforos Hadjicostis under the direction of Editor Christos G. Cassandras.

E-mail addresses: xiangyin@umich.edu (X. Yin), stephane@umich.edu (S. Lafortune).
[1] Fax: +1 7347638041.

http://dx.doi.org/10.1016/j.automatica.2017.06.013
0005-1098/

approaches either impose additional assumptions on the structure of the system or consider stronger properties than *centralized* diagnosability, i.e., diagnosability of the monolithic model. Whether or not it is possible to verify centralized diagnosability without exploring the entire exponential state space is still unclear.

In this paper, we show that verifying centralized diagnosability for a modular DES is in fact PSPACE-complete when the number of modules is unbounded. Our result relies on reductions from the deterministic finite-state automata intersection problem and the Büchi emptiness problem, which are known to be PSPACE-complete problems (Kozen, 1977). Moreover, we show that deciding predictability and deciding detectability for modular DES are both PSPACE-hard. These results reveal that in order to verify these properties for the complete system, exploring the state space of the monolithic model may be unavoidable, in the worst case.

In the DES literature, many complexity results for different problems have been established; see, e.g., Cassez (2012), Gohari and Wonham (2000), Rohloff and Lafortune (2005), Rohloff, Yoo, and Lafortune (2003), Saboori and Hadjicostis (2012), Su (2014) and Tsitsiklis (1989). Here, we briefly review some works that are closely related to the present paper and compare their results with ours. In Gohari and Wonham (2000) and Rohloff and Lafortune (2005), the *supervisor existence problem for modular DES* was studied. In particular, the authors of Gohari and Wonham (2000) showed that the supervisor existence problem for modular DES is NP-hard. This result was further improved in Rohloff and Lafortune (2005), in which the authors showed that the supervisor existence problem is in fact PSPACE-complete. However, in the present paper, we consider the complexity of the diagnosis problem, not of the control problem. In Rohloff et al. (2003), the *decentralized control problem for monolithic DES* was studied. Specifically, the authors showed that deciding coobservability, the key property in the decentralized control problem, is PSPACE-complete. In Cassez (2012), the author showed that deciding codiagnosability, a key property in the decentralized diagnosis problem, is PSPACE-complete for both timed and untimed automata. In contrast, in the present paper, we consider the *centralized diagnosis problem for modular systems*, which is clearly a different and incomparable problem. Recall that in the decentralized diagnosis problem, there is a monolithic system monitored by *a set* of agents, while in the centralized diagnosis problem there is only one diagnoser. To the best of our knowledge, the complexity of centralized fault diagnosis for modular DES is still open. Moreover, we also consider the complexity of fault prediction and state detection for modular DES, which have also not been investigated in the literature.

The remainder of this paper is organized as follows. In Section 2, we present the model of the system to be analyzed. Some basic concepts from complexity theory are reviewed in Section 3, for the sake of completeness of this paper. In Section 4, we show that deciding diagnosability for a modular DES is PSPACE-complete by reducing the deterministic finite-state automata intersection problem to the diagnosability verification problem. Using similar reductions, in Section 5, we show that deciding predictability and detectability are both PSPACE-hard for modular DES. Finally, we conclude the paper in Section 6.

2. Preliminaries

Let Σ be a finite set of events. We denote by Σ^* the set of all finite strings over Σ, including the empty string ϵ. A language $L \subseteq \Sigma^*$ is a subset of Σ^*. For any language L, $\overline{L} = \{s \in \Sigma^* : \exists t \in \Sigma^* \text{ s.t. } st \in L\}$ denotes the prefix closure of L. We say that a language L is live if $\forall s \in L, \exists \sigma \in \Sigma : s\sigma \in L$. We denote by L/s the post-language of L after string s, i.e., $L/s = \{t \in \Sigma^* : st \in L\}$. For any string s and a prefix string $t \in \overline{\{s\}}$, we denote by s/t the post-string of s after t, i.e., $t(s/t) = s$.

A DES is modeled as a deterministic finite-state automaton

$$G = (X, \Sigma, \delta, x_0, X_m)$$

where X is the finite set of states, Σ is the finite set of events, $\delta : X \times \Sigma \to X$ is the partial transition function, x_0 is the initial state, and X_m is the set of marked states. The transition function δ is extended to $X \times \Sigma^*$ in the usual manner; see, e.g., Cassandras and Lafortune (2008). We denote by $\mathcal{L}(G) = \{s \in \Sigma^* : \delta(x_0, s)!\}$ the language generated by G, where ! means "is defined". We also denote by $\mathcal{L}_m(G) = \{s \in \Sigma^* : \delta(x_0, s) \in X_m\}$ the language marked by G.

Let $G_1 = (X_1, \Sigma_1, \delta_1, x_{0,1}, X_{m,1})$ and $G_2 = (X_2, \Sigma_2, \delta_2, x_{0,2}, X_{m,2})$ be two automata. The parallel composition of G_1 and G_2, denoted by $G_1 \parallel G_2$, is defined as the accessible part of $(X_1 \times X_2, \Sigma_1 \cup \Sigma_2, \delta_{1\|2}, (x_{0,1}, x_{0,2}), X_{m,1} \times X_{m,2})$, where the transition function is defined by: for any $(x_1, x_2) \in X_1 \times X_2$, for any $\sigma \in \Sigma_1 \cup \Sigma_2$

$$\delta_{1\|2}((x_1, x_2), \sigma) = \begin{cases} (\delta_1(x_1, \sigma), \delta_2(x_2, \sigma)) & \text{if } \sigma \in \Sigma_1 \cap \Sigma_2 \\ (\delta_1(x_1, \sigma), x_2) & \text{if } \sigma \in \Sigma_1 \setminus \Sigma_2 \\ (x_1, \delta_2(x_2, \sigma)) & \text{if } \sigma \in \Sigma_2 \setminus \Sigma_1. \end{cases}$$

In many cases, the monolithic system G is obtained by composing a set of n modules $\{G_1, G_2, \ldots, G_n\}$, i.e., $G = G_1 \| G_2 \| \cdots \| G_n$, where $G_i = (X_i, \Sigma_i, \delta_i, x_{0,i}, X_{m,i}), i = 1, \ldots, n$. Throughout the paper, we assume that all the module automata have the same event set, i.e., $\Sigma_i = \Sigma, \forall i \in \{1, \ldots, n\}$. This assumption is without loss of generality, since we can always add self-loops for events in $\Sigma \setminus \Sigma_i$ at each state of G_i. With this assumption, we have that $\mathcal{L}(G) = \mathcal{L}(G_1) \cap \cdots \cap \mathcal{L}(G_n)$ and $\mathcal{L}_m(G) = \mathcal{L}_m(G_1) \cap \cdots \cap \mathcal{L}_m(G_n)$. Let $\Sigma_o \subseteq \Sigma$ be the set of globally observable events and define $\Sigma_{uo} = \Sigma \setminus \Sigma_o$. The natural projection $P : \Sigma^* \to \Sigma_o^*$ is defined by

$$P(\epsilon) = \epsilon \quad P(s\sigma) = \begin{cases} P(s)\sigma & \text{if } \sigma \in \Sigma_o \\ P(s) & \text{if } \sigma \notin \Sigma_o. \end{cases} \tag{1}$$

The first property that we consider is diagnosability (Sampath et al., 1995). Let $\sigma_f \in \Sigma_{uo}$ be the fault event whose occurrences must be detected by the diagnoser. We write $\sigma_f \in s$, if σ_f occurs at least once in s. We say that $\mathcal{L}(G)$ is diagnosable if any occurrence of σ_f can be unambiguously detected by the diagnoser within a finite number of steps. The formal definition of diagnosability is recalled from Sampath et al. (1995); let $\mathbb{N}$ denote the set of non-negative integers.

Definition 2.1 (*Diagnosability*)**.** A live language $\mathcal{L}(G)$ is said to be diagnosable w.r.t. Σ_{uo} and σ_f if

$$(\exists k \in \mathbb{N})(\forall s\sigma_f \in \mathcal{L}(G))(\forall t \in \mathcal{L}(G)/s\sigma_f)[|t| \geq k \Rightarrow D]$$

where the diagnosability condition D is

$$(\forall w \in \mathcal{L}(G))[P(w) = P(s\sigma_f t) \Rightarrow \sigma_f \in w].$$

It is shown in Jiang et al. (2001) and Yoo and Lafortune (2002) that given the monolithic system G, whether or not $\mathcal{L}(G)$ is diagnosable can be verified in polynomial time w.r.t. the size of G. However, when G is constructed from a set of modules $\{G_1, \ldots, G_n\}$, the size of G is exponential in the number of automaton modules (in the worst case). One question that arises naturally is whether or not we can verify the diagnosability of the entire system in some modular manner without constructing the entire state space of G. Formally, we consider the following problem.

Diagnosis Problem for Modular System (DPM)

- INSTANCE: A set of automata $\{G_1, G_2, \ldots, G_n\}$, a set of unobservable events Σ_{uo} and a fault event $\sigma_f \in \Sigma_{uo}$.
- QUESTION: Whether or not $\mathcal{L}(G_1 \| G_2 \| \cdots \| G_n)$ is diagnosable w.r.t. Σ_{uo} and σ_f?

3. Review of computational complexity

In this section, we briefly review some concepts and results from the theory of computation; see Garey and Johnson (1979) and Hopcroft and Ullman (1979) for detailed coverage of this theory.

We say that a problem is in class P if it can be solved in polynomial time by a deterministic algorithm. A problem is in the class NP if it can be solved in polynomial time by a non-deterministic algorithm. The class PSPACE is the set of problems that can be solved by deterministic algorithms using polynomial amount of space. The class NPSPACE is the set of problems solvable by non-deterministic algorithms using polynomial amount of space. According to Savitch's theorem (Savitch, 1970), PSPACE=NPSPACE. Also, it is well known that P⊆NP⊆PSPACE. However, whether or not these set inclusions are proper are still open problems.

Given two problems $\mathcal{A}$ and $\mathcal{B}$, we regard $\mathcal{B}$ as "at least as difficult as $\mathcal{A}$" if one can reduce $\mathcal{A}$ to $\mathcal{B}$ in polynomial time. In other words, this means that if $\mathcal{B}$ can be solved in polynomial time, then $\mathcal{A}$ can also be solved in polynomial time. We say that a problem $\mathcal{P}$ is PSPACE-complete if (i) $\mathcal{P}$ is in PSPACE and; (ii) any problem in PSPACE can be reduced to $\mathcal{P}$ in polynomial time. We say that a problem $\mathcal{P}$ is PSPACE-hard if there exists a PSPACE-complete problem that can be reduced to $\mathcal{P}$ in polynomial time. Therefore, PSPACE-complete problems are regarded as the most difficult problems in PSPACE. In other words, it is highly unlikely that there exists a polynomial-time algorithm for such problems, unless P=NP and NP=PSPACE. If a problem is shown to be PSPACE-complete or PSPACE-hard, then it is good evidence that this problem is computationally intractable.

One important PSPACE-complete problem is the *deterministic finite-state automata intersection problem* (DFA-Int), as proved in Kozen (1977). Hereafter, we will use this problem in order to establish the PSPACE-completeness or hardness of a set of problems. First, we recall the formal definition of DFA-Int from Kozen (1977).

DFA Intersection Problem (DFA-Int)

- INSTANCE: A set of automata $\{G_1, G_2, \ldots, G_n\}$.
- QUESTION: Whether or not $\mathcal{L}_m(G_1 \| G_2 \| \cdots \| G_n) = \emptyset$?

A slightly different version of DFA-Int is the *Büchi emptiness problem* (BEP), which is also a known PSPACE-complete problem. In this case, instead of considering finite strings, we consider infinite strings in the system behavior. Given an automaton G, we denote by $\mathcal{L}_m^\omega(G)$ the set of infinite strings that visit the marked states infinitely often. Then the BEP is defined as follows.

Büchi Emptiness Problem (BEP)

- INSTANCE: A set of automata $\{G_1, G_2, \ldots, G_n\}$.
- QUESTION: Whether or not $\mathcal{L}_m^\omega(G_1 \| G_2 \| \cdots \| G_n) = \emptyset$?

4. PSPACE-Completeness of diagnosis problem for modular systems

In this section, we present the first main result of this paper, namely, that DPM is PSPACE-complete.

First, given a set of automata $\{G_1, G_2, \ldots, G_n\}$ as the instance of DFA-Int, we define a new set of automata $\{G_1^D, G_2^D, \ldots, G_n^D\}$ as follows.

Definition 4.1. Let $\{G_1, G_2, \ldots, G_n\}$ be the instance of DFA-Int, where $G_i = (X_i, \Sigma, \delta_i, x_{0,i}, X_{0,m}), i \in \{1, \ldots, n\}$. For each $i \in \{1, \ldots, n\}$, we define a new automaton $G_i^D = (X_i^D, \Sigma^D, \delta_i^D, x_{0,i}^D, X_{0,m}^D)$, where

- $X_i^D = X_i \cup \{Fault\}$ is the set of states, where *Fault* is a new state that is not in $\cup_{i=1}^n X_i$;
- $\Sigma_i^D = \Sigma \cup \{\sigma_f, \tau\}$ is the set of events, where σ_f and τ are two new events that are not in Σ;
- $x_{0,i}^D = x_{0,i}$ is the initial state;
- $X_{m,i}^D = X_{m,i}$ is the set of marked states;
- $\delta_i^D : X_i^D \times \Sigma^D \to X_i^D$ is the transition function defined by: for any $x \in X_i^D$, for any $\sigma \in \Sigma^D$, we have

$$\delta_i^D(x, \sigma) = \begin{cases} \delta_i(x, \sigma) & \text{if } x \in X_i \wedge \sigma \in \Sigma \\ Fault & \text{if } x \in X_{m,i} \wedge \sigma = \sigma_f \\ x & \text{if } \sigma = \tau. \end{cases} \quad (2)$$

We also define $G^D = (X^D, \Sigma^D, \delta^D, x_0^D, X_m^D) := G_1^D \| G_2^D \| \cdots \| G_n^D$.

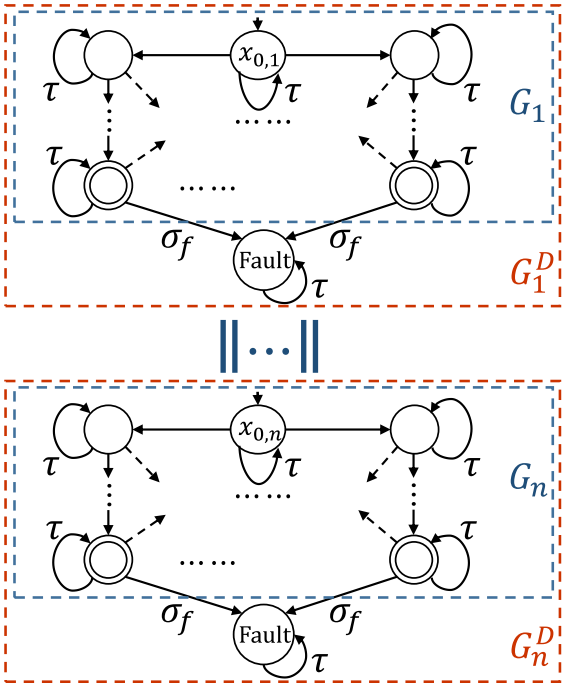


Fig. 1. Conceptual illustration of how to construct $\{G_1^D, \ldots, G_n^D\}$ from $\{G_1, \ldots, G_n\}$.

Intuitively, in order to obtain G_i^D, we add a fault event σ_f leading to the fault state *Fault* at each marked state of G_i. The self-loop τ at each state guarantees that the composed system G^D is live; therefore, diagnosability of $\mathcal{L}(G^D)$ is well-defined. Clearly, the construction of $\{G_1^D, G_2^D, \ldots, G_n^D\}$ can be done in $O(\sum_{i=1}^n |X_i|)$, which is polynomial in the size of the instance of DFA-Int. Fig. 1 provides a conceptual illustration showing how $\{G_1^D, \ldots, G_n^D\}$ are constructed from $\{G_1, \ldots, G_n\}$.

We show that DPM is PSPACE-hard by reducing DFA-Int to an instance of DPM.

Theorem 1. *DPM is PSPACE-hard.*

Proof. In order to prove this result, we reduce DFA-Int to DPM. Let $\{G_1, G_2, \ldots, G_n\}$ be the instance of DFA-Int and $\{G_1^D, G_2^D, \ldots, G_n^D\}$ be the automata constructed by Definition 4.1. Hereafter, we show that $\mathcal{L}_m(G_1 \| G_2 \| \cdots \| G_n) = \emptyset$ if and only if $\mathcal{L}(G_1^D \| G_2^D \| \cdots \| G_n^D)$ is diagnosable w.r.t. $\Sigma_{uo}^D := \{\sigma_f\}$ and σ_f.

($\Rightarrow$) By contraposition. Suppose that $\mathcal{L}(G^D) = \mathcal{L}(G_1^D \| G_2^D \| \cdots \| G_n^D)$ is not diagnosable w.r.t. $\Sigma_{uo}^D = \{\sigma_f\}$ and σ_f. Then we know that there must exist a fault string $s\sigma_f \in \mathcal{L}(G^D)$. Since σ_f is a common event of $\{G_1^D, G_2^D, \ldots, G_n^D\}$, by the property of parallel composition, we know that $\forall i \in \{1, 2, \ldots, n\} : s\sigma_f \in \mathcal{L}(G_i^D)$. Moreover, by the

construction of G_i^D, we know that σ_f is only defined at a marked state $x_i \in X_{m,i}^D$. Therefore, we know that

$$\delta^D(x_0^D, s) = (\delta_1^D(x_{0,i}^D, s), \ldots, \delta_n^D(x_{0,i}^D, s)) \in X_{m,1}^D \times \cdots \times X_{m,n}^D = X_m^D.$$

Also, by the construction in Definition 4.1, we know that $P_\tau(s) \in \mathcal{L}(G_i)$, where $P_\tau : (\Sigma \cup \{\tau\})^* \to \Sigma^*$ is the natural projection erasing event τ. Since $\forall t \in \mathcal{L}(G_i^D) : \delta_i(x_{0,i}, P_\tau(t)) = \delta_i^D(x_{0,i}^D, t)$ and $X_{m,i} = X_{m,i}^D, \forall i \in \{1, \ldots, n\}$, we have that

$$\delta(x_0, P_\tau(s)) = (\delta_1(x_{0,i}, P_\tau(s)), \ldots, \delta_n(x_{0,i}, P_\tau(s))) \in X_{m,1} \times \cdots \times X_{m,n} = X_m.$$

Therefore, we know that $\mathcal{L}_m(G_1 \| G_2 \ldots \| G_n) \neq \emptyset$.

($\Leftarrow$) We also show this direction by contraposition. Suppose that $\mathcal{L}(G) = \mathcal{L}_m(G_1 \| G_2 \| \cdots \| G_n) \neq \emptyset$. We know that there exists a string $s \in \mathcal{L}(G)$ such that

$$\delta(x_0, s) = (\delta_1(x_{0,i}, s), \ldots, \delta_n(x_{0,i}, s)) \in X_m.$$

Since $\mathcal{L}(G_i) \subseteq \mathcal{L}(G_i^D), \forall t \in \mathcal{L}(G_i) : \delta_i(x_{0,i}, t) = \delta_i^D(x_{0,i}^D, t)$ and $X_{m,i} = X_{m,i}^D, \forall i \in \{1, \ldots, n\}$ we have that

$$\delta^D(x_0^D, s) = (\delta_1^D(x_{0,i}^D, s), \ldots, \delta_n^D(x_{0,i}^D, s)) \in X_m^D.$$

Since each individual state component of $\delta^D(x_0^D, s)$ is marked, by the construction of G_i^D, we know that string $s\sigma_f\tau^k \in \mathcal{L}(G_i^D), \forall i \in \{1, \ldots, n\}, \forall k \in \mathbb{N}$, i.e., $s\sigma_f\tau^k \in \mathcal{L}(G^D)$ for any $k \in \mathbb{N}$. Also, we know that $s\tau^k \in \mathcal{L}(G^D)$ for any $k \in \mathbb{N}$, since $s\tau^k \in \mathcal{L}(G_i^D), \forall i \in \{1, \ldots, n\}, \forall k \in \mathbb{N}$. Therefore, we have that

$$(\forall k \in \mathbb{N})(\exists s\sigma_f \in \mathcal{L}(G^D))(\exists \tau^k \in \mathcal{L}(G^D)/s\sigma_f : |t| \geq n)$$
$$(\exists s\tau^k \in \mathcal{L}(G^D))[P(s\tau^k) = P(s\sigma_f\tau^k) \wedge \sigma_f \notin s\tau^k]$$

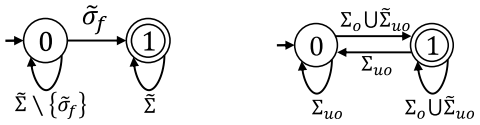
which means that $\mathcal{L}(G^D) = \mathcal{L}(G_1^D \| G_2^D \| \cdots \| G_n^D)$ is not diagnosable w.r.t. $\Sigma_{uo}^D = \{\sigma_f\}$ and σ_f. □

In order to show PSPACE-completeness, we need to show that DPM is in PSPACE. Our strategy for this proof is to reduce DPM to the Büchi emptiness problem (BEP), a known PSPACE-complete problem. First, we recall the procedure for the verification of diagnosability for a monolithic system G. Here, we adopt the approach proposed in Cassez and Tripakis (2008), Cassez (2012) and Moreira, Jesus, and Basilio (2011), where only standard parallel composition is used without constructing the verifier automaton as proposed in Jiang et al. (2001) and Yoo and Lafortune (2002).

Let G be the system to be diagnosed. Let $\tilde{\Sigma}_{uo} := \{\tilde{\sigma} : \sigma \in \Sigma_{uo}\}$ be a new set of events and define $\tilde{\Sigma} = \Sigma \cup \tilde{\Sigma}_{uo} = \Sigma_o \cup \Sigma_{uo} \cup \tilde{\Sigma}_{uo}$. In order to verify the diagnosability of $\mathcal{L}(G)$, we construct four automata: G^N, G^F, G^L and G^I defined as follows.

- $G^N = (X^N, \Sigma \setminus \{\sigma_f\}, \delta^N, x_0, X^N)$ represents the normal behavior of the system, which is defined as the accessible part of G after removing all fault transitions in G.
- $G^F = (X, \Sigma_o \cup \tilde{\Sigma}_{uo}, \delta^N, x_0, X)$ is the automaton obtained by renaming any event $\sigma \in \Sigma_{uo}$ in G by $\tilde{\sigma}$.
- $G^L = (\{0, 1\}, \tilde{\Sigma}, \delta^L, 0, \{1\})$ is the automaton shown in Fig. 2(a) with two states that capture the occurrence of the fault event (renamed as $\tilde{\sigma}_f$).
- $G^I = (\{0, 1\}, \tilde{\Sigma}, \delta^I, 0, \{1\})$ is the automaton shown in Fig. 2(b) in which a marked state is reached if and only if an event in $\Sigma_o \cup \tilde{\Sigma}_{uo}$, i.e., the event set of G^F, occurs.

Note that the event domains of G^N, G^F, G^L and G^I are different; one can simply add self-loops in each automaton such that they have the same event set. It follows from Cassez (2012) and Cassez and Tripakis (2008) that G is diagnosable if and only if $\mathcal{L}_m^\omega(G^N \| G^F \| G^L \parallel G^I) = \emptyset$. Note that a marked state in automaton



(a) G^L. (b) G^I.

Fig. 2. Automata used to verify diagnosability.

$G^N \| G^F \| G^L \parallel G^I$ is in the form of $(x, y, 1, 1), x, y \in X$. Intuitively, $\mathcal{L}_m^\omega(G^N \| G^F \| G^L \parallel G^I) \neq \emptyset$ implies that there exist an arbitrarily long fault string s and a non-fault string t such that $P(s) = P(t)$, i.e., diagnosability is violated.

We are now ready to establish PSPACE-completeness.

Theorem 2. *DPM is PSPACE-complete.*

Proof. We have already shown PSPACE-hardness. In order to show PSPACE-completeness, we reduce DPM to the Büchi emptiness problem, which shows that DPM is in PSPACE. Given $\{G_1, \ldots, G_n\}$, Σ_{uo} and σ_f as the instance of DPM, we construct two sets of automata $\{G_1^N, \ldots, G_n^N\}$ and $\{G_1^F, \ldots, G_n^F\}$ as follows. For each $i \in \{1, \ldots, n\}$, G_i^N is defined as the accessible part of G_i after removing all fault transitions in G_i and marking all remaining states. Also, for each $i \in \{1, \ldots, n\}$, G_i^F is obtained by renaming any event $\sigma \in \Sigma_{uo}$ in G_i by $\tilde{\sigma}$ and marking all the states. Clearly, we have that $G_1^F \| \ldots \| G_n^F = G^F$, where G^F still denotes the automaton obtained by renaming any event $\sigma \in \Sigma_{uo}$ in G by $\tilde{\sigma}$. Hereafter, we show that $G_1^N \| \ldots \| G_n^N = G^N$, where G^N still denotes the accessible part of G after removing all fault transitions in G.

To see this, consider a string $s \in \mathcal{L}(G^N)$ such that $\delta(x_0, s) = (x_1, \ldots, x_n)$. Since only fault transitions are removed in G_i and s is a non faulty string, then for each $i \in \{1, \ldots, n\}$, we have that $s \in \mathcal{L}(G_i^N)$. Hence, $\mathcal{L}(G^N) \subseteq \mathcal{L}(G_1^N) \cap \cdots \cap \mathcal{L}(G_n^N) = \mathcal{L}(G_1^N \| \ldots \| G_n^N)$. Similarly, consider a string $s \in \mathcal{L}(G_1^N \| \ldots \| G_n^N)$. Then we know that $s \in \mathcal{L}(G_i^N) \subseteq \mathcal{L}(G_i), \forall i \in \{1, \ldots, n\}$, which means that $s \in \mathcal{L}(G) = \mathcal{L}(G_i) \cap \cdots \cap \mathcal{L}(G_n)$. Since s is a non faulty string, according to the definition of G^N, we have $s \in \mathcal{L}(G^N)$. Overall, we have that $\mathcal{L}(G_1^N \| \ldots \| G_n^N) = \mathcal{L}(G^N)$. Moreover, since $\delta^N(x_0, s) = (\delta_1^N(x_{0,1}, s), \ldots, \delta_n^N(x_{0,n}, s))$ for any $s \in \mathcal{L}(G_1^N \| \ldots \| G_n^N) = \mathcal{L}(G^N)$ and we only consider the reachable part of G^N, we know that $G_1^N \| \ldots \| G_n^N = G^N$.

Note that constructing $\{G_1^N, \ldots, G_n^N\}$ and $\{G_1^F, \ldots, G_n^F\}$ can be done in $O(\sum_{i=1}^n |\Sigma||X_i|)$, which is polynomial w.r.t. the problem instance. Overall, we know that $\mathcal{L}(G)$ is diagnosable if and only if

$$\mathcal{L}_m^\omega(G_1^N \| \ldots \| G_n^N \| G_1^F \| \ldots \| G_n^F \| G^L \parallel G^I) = \emptyset. \tag{3}$$

This is an instance of the Büchi emptiness problem, which is in PSPACE. Therefore, we have proved that DPM is also in PSPACE. □

Remark 1. It is worth noting that, in the above proof, we construct two sets of automata $\{G_1^N, \ldots, G_n^N\}$ and $\{G_1^F, \ldots, G_n^F\}$ and then show that $G_1^F \| \ldots \| G_n^F = G^F$ and $G_1^N \| \ldots \| G_n^N = G^N$. However, we cannot directly use the monolithic G^F and G^N, which are defined in terms of G, since constructing G^F and G^N requires exponential state space. Although checking the emptiness of $G_1^N \| \ldots \| G_n^N \| G_1^F \| \ldots \| G_n^F \| G^L \parallel G^I$ requires exponential time, only polynomial space is required.

According to Definition 2.1, $\mathcal{L}(G)$ is diagnosable if there exists a non-negative integer K such that any fault can be detected within K steps after its occurrence. We say that $\mathcal{L}(G)$ is K-diagnosable if this delay bound K is given a priori. In fact, the PSPACE-hardness result established in Theorem 1 for diagnosability can also be extended to K-diagnosability. Consider the automaton G^D constructed in the proof of Theorem 1; we see that it is diagnosable if and only if it is

K-diagnosable for any non-negative integer K. Therefore, we have the following corollary.

Corollary 3. *For any non-negative integer K, verifying K-diagnosability for modular DES is PSPACE-hard.*

This result is interesting since it is generally believed that checking K-diagnosability is easier that checking diagnosability; the latter requires checking arbitrary long fault strings while the former only requires checking fault strings with finite length bounded by K. However, our results reveal that, for modular DES, specifying the delay K a priori will not simplify the verification problem in general.

Remark 2. Another important property in partially-observed DES is observability (Lin & Wonham, 1988); its verification has been shown to be PSPACE-complete in the modular setting (Rohloff & Lafortune, 2005). Also, it was shown in Wang, Girard, Lafortune, and Lin (2011) and Yin and Lafortune (2015) that diagnosability and observability can be reduced from one to the other in the monolithic setting. However, this equivalence does not directly yield the PSPACE-completeness of diagnosability in the modular setting, since the reduction from diagnosability to observability requires $O(|X|^2) = O(\prod_{i=1}^{n}|X_i|^2)$ state–space exploration, which is no longer a polynomial-time reduction in the modular setting.

5. Further results on predictability and detectability

In this section, we show that the verification of two other important properties, predictability and detectability, are both PSPACE-hard problems for modular DES.

5.1. PSPACE-hardness of deciding predictability

The notion of predictability is studied in Genc and Lafortune (2009) as the necessary and sufficient condition under which the fault event can be predicted unambiguously *before* its occurrence (Genc & Lafortune, 2009). First, we recall the definition of predictability from Genc and Lafortune (2009).

Definition 5.1 (*Predictability*)**.** A live language $\mathcal{L}(G)$ is said to be predictable w.r.t. Σ_{uo} and σ_f if

$$\begin{aligned}&(\exists k \in \mathbb{N})(\forall s\sigma_f \in \mathcal{L}(G))(\exists t \in \overline{\{s\}} : \sigma_f \notin t) \\ &(\forall u \in \mathcal{L}(G) : \sigma_f \notin u \wedge P(u) = P(t)) \\ &(\forall v \in \mathcal{L}(G)/u)[|v| \geq k \Rightarrow \sigma_f \in v].\end{aligned} \tag{4}$$

It was shown in Genc and Lafortune (2009) that predictability can be verified in polynomial time when the monolithic system is given. When the system is given as a set of modules, similarly to DPM, we define the prediction problem for modular systems as follows.

Prediction Problem for Modular Systems (PPM)

- INSTANCE: A set of automata $\{G_1, G_2, \ldots, G_n\}$, a set of unobservable events Σ_{uo}, and a fault event $\sigma_f \in \Sigma_{uo}$.
- QUESTION: Whether or not $\mathcal{L}(G_1 \| G_2 \| \cdots \| G_n)$ is predictable w.r.t. Σ_{uo} and σ_f?

The following theorem reveals that PPM is PSPACE-hard.

Theorem 4. *PPM is PSPACE-hard.*

Proof. To prove the result, we reduce DFA-Int to PPM. Let $\{G_1, G_2, \ldots, G_n\}$ be the instance of DFA-Int and $\{G_1^D, G_2^D, \ldots, G_n^D\}$ be the automata constructed by Definition 4.1. Hereafter, we show that $\mathcal{L}_m(G_1 \| G_2 \| \cdots \| G_n) = \emptyset$ if and only if $\mathcal{L}(G_1^D \| G_2^D \| \cdots \| G_n^D)$ is predictable w.r.t. $\Sigma_{uo}^D := \{\sigma_f\}$ and σ_f.

($\Rightarrow$) By contraposition. Suppose that $\mathcal{L}(G^D)$ is not predictable w.r.t. $\{\sigma_f\}$ and σ_f. Then we know that there must exist a fault string $s\sigma_f \in \mathcal{L}(G^D)$. We have shown in the proof of Theorem 1 that $s\sigma_f \in \mathcal{L}(G^D)$ implies that $\delta(x_0, P_\tau(s)) \in X_m$. Therefore, we know that $\mathcal{L}_m(G_1 \| G_2 \ldots \| G_n) \neq \emptyset$.

($\Leftarrow$) Also by contraposition. Suppose that $\mathcal{L}(G) = \mathcal{L}_m(G_1 \| \ldots \| G_n) \neq \emptyset$. We know that there exists a string $s \in \mathcal{L}(G)$ such that $\delta(x_0, s) \in X_m$. As we have shown in the proof of Theorem 1, $\delta(x_0, s) \in X_m$ implies that for any $k \in \mathbb{N}$, $s\sigma_f\tau^k, s\tau^k \in \mathcal{L}(G^D)$. Therefore, for any $k \in \mathbb{N}$ and the same string s, we have $s\sigma_f \in \mathcal{L}(G^D)$ and for any $t \in \overline{\{s\sigma_f\}} : \sigma_f \notin t$, there exists string $s\tau^k/t \in \mathcal{L}(G^D)/t$ such that $|s\tau^k/t| \geq k$ and $\sigma_f \notin s\tau^k/t$. That is, $\mathcal{L}(G^D)$ is not predictable w.r.t. $\{\sigma_f\}$ and σ_f. □

5.2. PSPACE-hardness of deciding detectability

Detectability is a property arising in state estimation of DES (Shu et al., 2007). It requires that the current state of the system be determined unambiguously after a finite number of observations. First, we recall the formal definition of strong detectability from Shu et al. (2007).

Definition 5.2 (*Strong Detectability*)**.** A system G is said to be strongly detectable w.r.t. Σ_{uo} if

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

where $R_G(s) = \{x \in X : \exists t \in \mathcal{L}(G) \text{ s.t. } P(t) = P(s) \wedge \delta(x_0, t) = x\}$.

As for DPM and PPM, we define the strong detection problem for modular DES.

Strong Detection Problem for Modular Systems (DTPM-S)

- INSTANCE: A set of automata $\{G_1, G_2, \ldots, G_n\}$, a set of unobservable events Σ_{uo}.
- QUESTION: Whether or not $G_1 \| G_2 \| \cdots \| G_n$ is strongly detectable w.r.t. Σ_{uo}?

The following result shows that DTPM-S is PSPACE-hard.

Theorem 5. *DTPM-S is PSPACE-hard.*

Proof. We reduce DFA-Int to DTPM-S. Let $\{G_1, G_2, \ldots, G_n\}$ be the instance of DFA-Int and $\{G_1^D, G_2^D, \ldots, G_n^D\}$ be the automata constructed by Definition 4.1. Hereafter, we show that $\mathcal{L}_m(G_1 \| G_2 \| \cdots \| G_n) = \emptyset$ if and only if $G^D = G_1^D \| G_2^D \| \cdots \| G_n^D$ is strongly detectable w.r.t. $\Sigma_{uo}^D := \{\sigma_f\}$.

($\Rightarrow$) Suppose that $\mathcal{L}_m(G) = \mathcal{L}_m(G_1 \| \ldots \| G_n) = \emptyset$. We know that there does not exist a string $s\sigma_f \in \mathcal{L}(G^D)$, since we have shown that $s\sigma_f \in \mathcal{L}(G^D)$ implies that $\delta(x_0, P_\tau(s)) \in X_m$. Therefore, we know that all strings in $\mathcal{L}(G^D)$ are observable, i.e., $\mathcal{L}(G^D) \subseteq (\Sigma \cup \{\tau\})^*$. Clearly, we know that $|R_{G^D}(s)| = 1$ for any $s \in \mathcal{L}(G^D)$, which means that G^D is strongly detectable.

($\Leftarrow$) We prove this direction by contraposition. Suppose that $\mathcal{L}_m(G) = \mathcal{L}_m(G_1 \| \ldots \| G_n) \neq \emptyset$. We know that there exists a string $s \in \mathcal{L}(G)$ such that $\delta(x_0, s) \in X_m$. As we have shown in the proof of Theorem 1, $\delta(x_0, s) \in X_m$ implies that $s\sigma_f\tau^k \in \mathcal{L}(G^D)$ for any $k \in \mathbb{N}$ and $s\tau^k \in \mathcal{L}(G^D)$ for any $k \in \mathbb{N}$. Since $P(s\sigma_f\tau^k) = P(s\tau^k)$ and $\delta^D(x_0^D, s\sigma_f\tau^k) \neq \delta^D(x_0^D, s\tau^k)$, we know that $|R_{G^D}(s\tau^k)| \geq 2$. Therefore, we know that G^D is not strongly detectable w.r.t. $\Sigma_{uo}^D = \{\sigma_f\}$. □

Note that strong detectability requires that we can *always* determine the current state of the system unambiguously after a finite number of observations. A weaker version of detectability, called weak detectability (Shu et al., 2007), requires that we can determine the current state of the system *for some* string.

Definition 5.3 (*Weak Detectability*)**.** A system G is said to be weakly detectable w.r.t. Σ_{uo} if

$$(\exists k \in \mathbb{N})(\exists s \in \mathcal{L}(G) : |P(s)| \geq k)[|R_G(s)| = 1]. \tag{6}$$

Similarly, we define DTPM-W if strong detectability in DTPM-S is replaced by weak detectability. The following result reveals that DTPM-W is still PSPACE-hard.

Theorem 6. *DTPM-W is PSPACE-hard.*

Proof. Let $\{G_1, G_2, \ldots, G_n\}$ be the instance of DFA-Int and $\{G_1^D, G_2^D, \ldots, G_n^D\}$ be the automata constructed by Definition 4.1. For each G_i^D, we define a new automaton $\tilde{G}_i^D$, by adding a new initial state $\tilde{x}_{0,i}^D$, where $\tilde{x}_{0,i}^D \notin X_i^D$. We add a new transition labeled by a new event τ' from the new initial state $\tilde{x}_{0,i}^D$ to the original initial state $x_{0,i}^D$. Also, we add a self-loop labeled by τ at the new initial state $\tilde{x}_{0,i}^D$. Clearly, the new set of automata $\{\tilde{G}_1^D, \tilde{G}_2^D, \ldots, \tilde{G}_n^D\}$ is polynomial in the size of the instance of DFA-Int. Hereafter, we show that $\mathcal{L}_m(G_1 \| G_2 \ldots \| G_n) \neq \emptyset$ if and only if $\tilde{G}^D = \tilde{G}_1^D \| \tilde{G}_2^D \| \cdots \| \tilde{G}_n^D$ is weakly detectable w.r.t. $\tilde{\Sigma}_{uo}^D := \Sigma \cup \{\tau'\}$, i.e., $\tilde{\Sigma}_o^D = \{\sigma_f, \tau\}$.

($\Rightarrow$) We prove this direction by contraposition. Suppose that $\mathcal{L}_m(G) = \mathcal{L}_m(G_1 \| \ldots \| G_n) = \emptyset$. We know that σ_f will not occur in G^D, which means that σ_f also will not occur in $\tilde{G}^D$. This means that the only event that can be observed in $\tilde{G}^D$ is τ, which is a self-loop event at each state in $\tilde{G}^D$. Therefore, for any string $s \in \mathcal{L}(\tilde{G}^D)$, we have $R_{\tilde{G}^D}(s) = \tilde{X}^D$, where $\tilde{X}^D$ is the set of states in $\tilde{G}^D$. Moreover, by the definition of $\{\tilde{G}_1^D, \tilde{G}_2^D, \ldots, \tilde{G}_n^D\}$, we have $\{(\tilde{x}_{0,1}^D, \ldots, \tilde{x}_{0,n}^D), (x_{0,1}^D, \ldots, x_{0,n}^D)\} \subseteq \tilde{X}^D$. Therefore, $\forall k \in \mathbb{N}, \forall s \in \mathcal{L}(\tilde{G}^D) : |P(s)| \geq k$, we have $|R_{\tilde{G}^D}(s)| = |\tilde{X}^D| \geq 2$, i.e., $\tilde{G}^D$ is not weakly detectable.

($\Leftarrow$) Suppose that $\mathcal{L}_m(G) = \mathcal{L}_m(G_1 \| \ldots \| G_n) \neq \emptyset$. We know that there exists a string $s \in \mathcal{L}(G)$ such that $\delta(x_0, s) \in X_m$. As we have shown in the proof of Theorem 1, $\delta(x_0, s) \in X_m$ implies that $s\sigma_f \in \mathcal{L}(G^D)$. By the definition of $\{\tilde{G}_1^D, \tilde{G}_2^D, \ldots, \tilde{G}_n^D\}$, this further implies that $\tau' s \sigma_f \in \mathcal{L}(\tilde{G}^D)$. In each $\tilde{G}_i^D$, once event σ_f is observed, we know for sure that the current state is *Fault*, i.e., $|R_{\tilde{G}^D}(\tau' s \sigma_f)| = |\{(Fault, \ldots, Fault)\}| = 1$. Therefore, by taking $k = |P(\tau' s \sigma_f)| = 1$, we know that $\tilde{G}^D$ is weakly detectable w.r.t. $\tilde{\Sigma}_{uo}^D := \Sigma \cup \{\tau'\}$. □

Remark 3. We have shown that the verification of predictability and detectability are both PSPACE-hard for modular DES. However, the PSPACE-completeness of these two conditions is still open. One possible direction is to investigate whether or not the verification of these two properties can also be reduced to the Büchi emptiness problem by using techniques similar to that in Cassez (2012) and Cassez and Tripakis (2008). This direction deserves a detailed future study.

6. Conclusion

We have shown that deciding centralized diagnosability for modular DES is PSPACE-complete. We also showed that deciding centralized predictability and centralized (strong and weak) detectability for modular DES are PSPACE-hard problems. These computational intractability results suggest the following future research directions. First, one may be interested in investigating stronger modular versions of these properties that can be verified modularly. In fact, this direction has already been investigated in Contant et al. (2006), Debouk et al. (2002) and Zhou et al. (2008) for diagnosability and one may extend the ideas therein to predictability and detectability. Another direction is to identify sufficient conditions under which the centralized or monolithic versions of these properties can be verified efficiently in some modular manner; see, e.g., Gummadi et al. (2011) for relevant work in the context of modular control.

References

Cassandras, C. G., Lafortune, S. (2008). *Introduction to discrete event systems* (2nd ed.). Springer.

Cassez, F. (2012). The complexity of codiagnosability for discrete event and timed systems. *IEEE Transactions on Automatic Control*, *57*(7), 1752–1764.

Cassez, F., & Tripakis, S. (2008). Fault diagnosis with static and dynamic observers. *Fundamenta Informaticae*, *88*(4), 497–540.

Contant, O., Lafortune, S., & Teneketzis, D. (2006). Diagnosability of discrete event systems with modular structure. *Discrete Event Dynamic Systems: Theory & Applications*, *16*(1), 9–37.

Debouk, R., Malik, R., & Brandin, B. (2002). A modular architecture for diagnosis of discrete event systems. In *41st IEEE conference on decision and control* (pp. 417–422).

Fabre, E., Benveniste, A., Haar, S., & Jard, C. (2005). Distributed monitoring of concurrent and asynchronous systems. *Discrete Event Dynamic Systems: Theory & Applications*, *15*(1), 33–84.

Feng, L., & Wonham, W. M. (2008). Supervisory control architecture for discrete-event systems. *IEEE Transactions on Automatic Control*, *53*(6), 1449–1461.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness. Vol. 29*. W.H. Freeman.

Genc, S., & Lafortune, S. (2009). Predictability of event occurrences in partially-observed discrete-event systems. *Automatica*, *45*(2), 301–311.

Gohari, P., & Wonham, W. M. (2000). On the complexity of supervisory control design in the RW framework. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *30*(5), 643–652.

Gummadi, R., Singh, N., & Sreenivas, R. S. (2011). On tractable instances of modular supervisory control. *IEEE Transactions on Automatic Control*, *56*(7), 1621–1635.

Hill, R. C., Cury, J. E. R., de Queiroz, M. H., Tilbury, D., & Lafortune, S. (2010). Multi-level hierarchical interface-based supervisory control. *Automatica*, *46*(7), 1152–1164.

Hopcroft, J. E., & Ullman, J. D. (1979). *Introduction to automata theory, languages, and computation*. Addison-Wesley.

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.

Komenda, J., van Schuppen, J. H., Gaudin, B., & Marchand, H. (2008). Supervisory control of modular systems with global specification languages. *Automatica*, *44*(4), 1127–1134.

Kozen, D. (1977). Lower bounds for natural proof systems. In *18th symp. foundations computer science* (pp. 254–266).

Leduc, R., Lawford, M., & Wonham, W. M. (2005). Hierarchical interface-based supervisory control-part II: parallel case. *IEEE Transactions on Automatic Control*, *50*(9), 1336–1348.

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

Moreira, M., Jesus, T. C., & Basilio, J. C. (2011). Polynomial time verification of decentralized diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, *56*(7), 1679–1684.

Pencolé, Y. (2004). Diagnosability analysis of distributed discrete event systems. In *European conference on AI* (pp. 43–47).

Ramírez-Treviño, A., Ruiz-Beltrán, E., Rivera-Rangel, I., & López-Mellado, E. (2007). Online fault diagnosis of discrete event systems. A petri net-based approach. *IEEE Transactions on Automation Science and Engineering*, *4*(1), 31–39.

Ricker, S. L., & Fabre, E. (2000). On the construction of modular observers and diagnosers for discrete-event systems. In *39th IEEE conference on decision and control* (pp. 2240–2244).

Rohloff, K., & Lafortune, S. (2005). PSPACE-completeness of Modular Supervisory Control Problems. *Discrete Event Dynamic Systems: Theory & Applications*, *15*(2), 145–167.

Rohloff, K., Yoo, T.-S., & Lafortune, S. (2003). Deciding co-observability is PSPACE-complete. *IEEE Transactions on Automatic Control*, *48*(11), 1995–1999.

Saboori, A., & Hadjicostis, C. N. (2010). Reduced-complexity verification for initial-state opacity in modular discrete event systems. In *10th int. workshop on discrete event systems* (pp. 78–83).

Saboori, A., & Hadjicostis, C. N. (2012). Verification of infinite-step opacity and complexity considerations. *IEEE Transactions on Automatic Control*, *57*(5), 1265–1269.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., & Teneketzis, D. (1995). Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, *40*(9), 1555–1575.

Savitch, W. J. (1970). Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, *4*(2), 177–192.

Schmidt, K. (2013). Verification of modular diagnosability with local specifications for discrete-event systems. *IEEE Transactions on Systems, Man and Cybernetics*, *43*(5), 1130–1140.

Schmidt, K., & Cury, J. E. R. (2012). Efficient abstractions for the supervisory control of modular discrete event systems. *IEEE Transactions on Automatic Control*, *57*(12), 3224–3229.

Shu, S., Lin, F., & Ying, H. (2007). Detectability of discrete event systems. *IEEE Transactions on Automatic Control*, *52*(12), 2356–2359.

Su, R. (2014). On the complexity of synthesizing a minimum-weighted supervisor under partial observation. *Automatica*, *50*(6), 1725–1729.

Tsitsiklis, J. N. (1989). On the control of discrete-event dynamical systems. *Mathematics of Control, Signals, and Systems*, *2*(2), 95–107.

Wang, W., Girard, A. R., Lafortune, S., & Lin, F. (2011). On codiagnosability and coobservability with dynamic observations. *IEEE Transactions on Automatic Control*, *56*(7), 1551–1566.

Ye, L., & Dague, P. (2010). Diagnosability analysis of discrete event systems with autonomous components. In *European conference on AI* (pp. 105–110).

Yin, X., & Lafortune, S. (2015). Codiagnosability and coobservability under dynamic observations: Transformation and verification. *Automatica*, *61*, 241–252.

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.

Zhou, C., Kumar, R., & Sreenivas, R. S. (2008). Decentralized modular diagnosis of concurrent discrete event systems. In *9th international workshop on discrete event systems* (pp. 388–393.

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. His research interests include supervisory control of discrete-event systems, model-based fault diagnosis, formal methods, security and their applications to cyber and cyber-physical systems. Dr. Yin received the Outstanding Reviewer Award from Automatica in 2016, the Outstanding Reviewer Award from IEEE Transactions on Automatic Control in 2017 and the IEEE Conference on Decision and Control (CDC) Best Student Paper Award Finalist in 2016. He is the co-chair of the IEEE CSS Technical Committee on Discrete Event Systems.

Stéphane Lafortune received the B.Eng. degree from Ecole Polytechnique de Montréal in 1980, the M.Eng. degree from McGill University in 1982, and the Ph.D. degree from the University of California at Berkeley in 1986, all in electrical engineering. Since September 1986, he has been with the University of Michigan, Ann Arbor, where he is a Professor of Electrical Engineering and Computer Science.

Dr. Lafortune is a Fellow of the IEEE (1999).

He received the Presidential Young Investigator Award from the National Science Foundation in 1990 and the George S. Axelby Outstanding Paper Award from the Control Systems Society of the IEEE in 1994 (for a paper co-authored with S.-L. Chung and F. Lin) and in 2001 (for a paper co-authored with G. Barrett).

Dr. Lafortune's research interests are in discrete event systems and include multiple problem domains: modeling, diagnosis, control, optimization, and applications to computer and software systems.

He is the lead developer of the software package UMDES and co-developer of DESUMA with L. Ricker.

He co-authored, with C. Cassandras, the textbook Introduction to Discrete Event Systems—Second Edition (Springer, 2008).

Dr. Lafortune is Editor-in-Chief of the Journal of Discrete Event Dynamic Systems: Theory and Applications.