CEAI, Vol.22, No.1 pp. 32-41, 2020 Printed in Romania

Relative Codiagnosability of Decentralized Failure Diagnosis Systems

Rui Zhao, Fuchun Liu Jr

School of Computers, Guangdong University of Technology, Guangzhou 510006
China (Tel: +8613160809171; e-mail: zhaorui118204@163.com, corresponding author: fliu2011@163.com).

Abstract: Due to the practical and theoretical importance, failure diagnosis has received considerable attention in the recent years. This paper investigates the relative diagnosability of Discrete-event Systems (DESs) under the decentralized framework. The notion of relative codiagnosability is formalized to capture the feature that there exists at least a failure event which can be detected based on at least one local observation. It is deduced that relative codiagnosability is weaker than codiagnosability and relative diagnosability. In order to achieve the performance of a decentralized failure diagnosis system, the necessary and sufficient conditions for verifying the relative codiagnosability are presented and a polynomial-time algorithm is developed to test the relative codiagnosability by introducing opacity. Furthermore, some examples are provided to illustrate the presented results. It is worth noting that the reported work extends the idea of codiagnosability to general cases and generalizes the main results of relative diagnosability in the centralized observation to the decentralized setting.

Keywords: diagnosability, decentralized fault diagnosis, discrete-event systems.

1. INTRODUCTION

DESs have been successfully applied to provide a formal treatment of many technological and engineering systems such as automated manufacturing systems, artificial intelligence, computer networks, transportation systems, communication protocols, robot coordination systems, process control and power systems. Due to the practical and theoretical importance, failure diagnosis of DESs, which aims to timely identify and exactly characterize the occurrences of incipient faults that may not be directly observed by the sensors from the expected or desired behaviours, has received considerable attention in the recent years (Basile et al., 2009; Carvalho et al., 2013; Chen et al., 2014; Li et al., 2015; Cabral et al., 2015; Yao and Feng, 2016; Deng and Qiu, 2016; Geng et al., 2017). In particular, (Sampath et al., 1995) proposed an approach, in which a diagnoser was constructed to perform the on-line detection and isolation of failure events and off-line verification of the diagnosability property of a system. And (Lin, 2011) studied the diagnosability property by introducing opacity (Saboori and Hadjicostis, 2012, 2014; Zhang et al., 2015; Wu and Lafortune, 2016; Yin and Lafortune, 2017). In addition, a number of variations of diagnosability focused on centralized framework have been investigated in the works (Thorsley and Teneketzis, 2005; Liu, 2009, 2014, 2015; Biswas, 2012;), where there is a single site to collect all the information about a system and there is only a diagnoser performing fault detection.

For many large complex distributed systems, the centralized failure diagnosis framework may not always be appropriate, and instead failure diagnosis may need to be performed at some decentralized sites where diagnosis information is collected. In the decentralized setting, there is a family of local diagnosers running at several sites processing the local observations. Each local diagnoser may only observe the part of the dynamic behaviour of the system, and diagnosis is performed dispersedly at each local site where diagnosis information is collected. Up until now, more and more researchers have devoted to the decentralized failure diagnosis of DESs (Qiu and Kumar, 2006; Liu et al., 2008; Ran et al., 2018; Moreira et al., 2011; Chen and Kumar, 2013; Sayed-Mouchaweh and Lughofer, 2015; Yin and Lafortune, 2017; Deng and Qiu, 2017; Pérez-Zuñiga and Chanthery, 2018). In the work of (Qiu and Kumar, 2006), the notion of decentralized failure diagnosis is formalized by introducing the definition of codiagnosability that requires that a failure can be detected by one of the diagnosers within a bounded delay. Diagnosability ensures that the diagnoser based on the centralized system model will always be able to diagnose all failures without ambiguity, while codiagnosability guarantees that all failures are diagnosed in a decentralized manner using several local diagnosers. However, most complex engineering systems are difficult to satisfy the conditions of codiagnosability, so they are often identified as non-codiagnosable. In the case, the diagnoser approach proposed by (Sampath et al., 1995) cannot be used as a diagnosis tool for on-line failure detection. To solve the problem, we consider the problem of relative diagnosability in the framework of decentralized failure diagnosis.

In the previous work, we have formalized the relative diagnosability (Zhao et al., 2017) and the relative predictability (Zhao et al., 2019) of DESs in the centralized setting. This paper is a continuation of the work (Zhao et al., 2017, 2019) and its three main contributions are as follows.

First, the notion of relative codiagnosability of DESs is formalized under the decentralized framework to describe the property that the occurrence of at least a failure event can be detected based on at least one local observation. And the relationship with the notion of codiagnosability introduced by (Qiu and Kumar, 2006) is analysed. It is deducted that relative codiagnosability is weaker than codiagnosability and codiagnosability can be viewed as a special case of the relative codiagnosability. Moreover, the notion is different from the copredictability (Liu, 2019), which is proposed to capture the feature of copredictable DESs that the occurrences of failure events can be predicted based on at least one local observation. In addition, the relative codiagnosability is clearly different from the k-reliable copredictability (Liu, 2018) which means that prediction performance will not be degraded even when m-k local agents are unavailable.

Second, codiagnosable rate is defined to accurately describe the relative codiagnosability. If the codiagnosable rate of a system equals one, then the system is codiagnosable and it is considered in the previous works (Qiu and Kumar, 2006; Liu et al., 2008; Ran et al., 2018; Moreira et al., 2011; Chen and Kumar, 2013; Sayed-Mouchaweh and Lughofer, 2015; Yin and Lafortune, 2017; Deng and Qiu, 2017); If the codiagnosable rate is greater than 0 and less than 1, then the system is relatively codiagnosable; If the codiagnosable rate equals zero, the system is not relatively codiagnosable. It is worth noting that under the framework proposed by (Qiu and Kumar, 2006), relatively codiagnosable systems are identified as non-codiagnosable, so the diagnoser approach cannot be used for failure diagnosis. But in our framework, relatively codiagnosable systems are considered as partially codiagnosable, and the diagnoser approach can still be chosen as an on-line failure diagnosis tool to detect the occurrences of some failures. Moreover, the codiagnosable rate is also a measure of relatively codiagnosable degree of different systems.

Third, the necessary and sufficient conditions for verifying the relative codiagnosability are presented and a polynomial-time algorithm based on opacity is developed for accurately computing the value of codiagnosable rate. Moreover, the algorithm is also used to deal with the problem of codiagnosability. So our results generalize the important consequences of (Qiu and Kumar, 2006).

2. PRELIMINARIES

The paper investigates the DES modeled by a deterministic automaton $G=(X,\Sigma,\delta,x_0)$, where X is the set of states, Σ is the finite set of events, a partial function $\delta: X\times\Sigma\to X$ is the transition function and $x_0\in X$ is the initial state. Σ^* is the set of all finite strings over Σ, including the empty string ϵ. For any $s\in\Sigma^*$ and any $x\in X$, use the notation $\delta(x_0,s)!$ to denote that $\delta(x,s)$ is defined. The generated language of G, denoted by L (or $L(G)$), is defined by $L=\{s\in\Sigma^*:\ \delta(x_0,s)!\}$. Let s be a trace originating from x_0, $L/s=\{t\in\Sigma^*:\ st\in L\}$ denotes the post language of L after s and the length of s (number of events including repetitions) is denoted by $|s|$. The event set $\Sigma=\Sigma_o\cup\Sigma_{uo}$, where Σ_o is the set of observable events and Σ_{uo} is the set of unobservable events. Let $\Sigma_f\subseteq\Sigma_{uo}$ be the set of failure events and s_f denote the final event of the trace s, define $\psi(\Sigma_f)$ as the set of all traces of L that end in a failure event, i.e., $\psi(\Sigma_f)=\{s\in L: s_f\in\Sigma_f\}$.

Let $P:\Sigma^*\to\Sigma_o^*$ denote the usual projection operator which is used to filter out the unobservable events from a trace. The inverse projection is $P_L^{-1}(y)=\{s\in L: P(s)=y\}$. Also to avoid unnecessary complexity, the paper assumes that the language L generated by G is live and there are no cycles of unobservable events in G.

Definition 1 (Zhao et al., 2017): A path $(x_0,\sigma_0,x_1,...,\sigma_{n-1},x_n,\sigma_n,x_0)$ of G forms an ultimate cycle, if there are no any other transitions for each state x_l except for $\delta(x_l,\sigma_l)=x_{(l+1)mod(n+1)}$, where $x_l\in X,\sigma_l\in\Sigma$, $l\in\{0,1,2,...,n\}$.

Definition 2 (Zhao et al., 2017): A path of G forms a branch if the path originates from state x_0 and ends in an ultimate cycle.

A system usually can be divided into several branches. B_f represents the set of failure branches that contain at least one failure event. $B_{\sim f}$ represents the set of normal branches that do not contain any failure events..

Definition 3 (Qiu and Kumar, 2006): Let L be the language of G. Assume there are m local projections $P_i:\Sigma^*\to\Sigma_{o,i}^*$, where $i\in\mathrm{M}$ and $\mathrm{M}=\{1,2,...,m\}$. L is said to be codiagnosable w.r.t. $\{P_i\}_{i\in\mathrm{M}}$ if there is a $n_0\in N$ such that

$$(\forall s\in\psi(\Sigma_f))(\exists j\in\mathrm{M})(\forall t\in L/s)[|t|\geq n_0\Rightarrow D_j] \quad (1)$$

Where the diagnosability condition function $D_j(st):\Sigma^*\to\{0,1\}$ is defined as follows:

$$D_j(st)=\begin{cases}1 & \text{if } \forall\omega\in(P_j^{-1}P_j(st)\cap L)\Rightarrow\Sigma_f\in\omega\\ 0 & \text{otherwise}\end{cases} \quad (2)$$

Intuitively, L being codiagnosable means that for any trace s that ends in a failure event and any sufficiently long continuation t of s, there exists at least one site j such that the j th diagnoser can detect the fault among the traces indistinguishable from st within a finite delay.

3. RELATIVE CODIAGNOSABILITY AND ITS NECESSARY AND SUFFICIENT CONDITIONS

3.1 Definition of Relative Codiagnosability

In the section, we consider relative diagnosability of DESs, where there are m local diagnosers to detect the failures. The m local diagnosers are assumed to be independent, namely, without communicating their observations to each other.

Let L be the prefix-closed and live language of G. Now let us give the definition of relative codiagnosability.

Definition 4: A trace $s \in \psi(\Sigma_f)$ is said to be codiagnosable w.r.t. $\{P_i\}_{i \in M}$ if there is a $n_0 \in N$ such that

$$\begin{aligned}&(\exists j \in \mathrm{M})(\forall t \in L/s, |t| \geq n_0) \Rightarrow \\ &(\forall \omega \in L \wedge P_j(\omega) = P_j(st))(\Sigma_f \in \omega)\end{aligned} \tag{3}$$

Definition 5: If there exists a trace s in $\psi(\Sigma_f)$ and s is codiagnosable w.r.t. $\{P_i\}_{i \in M}$, then L is said to be relatively codiagnosable w.r.t. $\{P_i\}_{i \in M}$ and s.

Intuitively, L being relative codiagnosable means that there is at least one trace s in L ending with a failure event and for any sufficiently long continuation t of s, there exists at least one site j such that the j th diagnoser can detect the fault among the traces indistinguishable from st within a finite delay.

Example 1: Consider the automaton $G = (X, \Sigma, \delta, x_0)$ described by Fig.1, where $X = \{x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$, $\Sigma = \{\alpha, \beta, \gamma, \sigma_{uo}, \sigma_f\}$ and σ_f is a failure event. Assume that there are two local diagnosers to detect failures of the system and there are two local projections $P_i : \Sigma^* \rightarrow \Sigma_{o,i}^*$, where $i \in \{1, 2\}$, and $\Sigma_{o,1} = \{\alpha, \beta\}$, $\Sigma_{o,2} = \{\alpha, \gamma\}$.

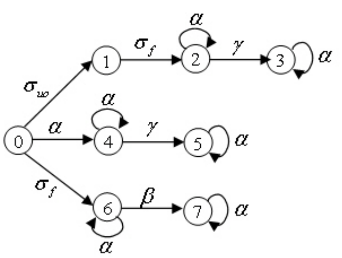


Fig. 1. Automaton G.

Let s be a trace of G and $s \in \psi(\Sigma_f)$, then $s = \sigma_{uo}\sigma_f$ or $s = \sigma_f$.

Case 1: Take $s = \sigma_{uo}\sigma_f$, then $t = \alpha^*\gamma\alpha^*$ satisfies $t \in L/s$ and $|t| \geq n_0 (n_0 \in \mathrm{N})$. If choose the first diagnoser with P_1 to detect the failure event σ_f, then there is a string $\omega = \alpha^*\gamma\alpha^*$ such that $P_1(\omega) = P_1(st) = \alpha^*$, but $\sigma_f \notin \omega$.

Similarly, if choose the second diagnoser with P_2 to detect the failure, then there is a string $\omega = \alpha^*\gamma\alpha^*$ such that $P_2(\omega) = P_2(st) = \alpha^*\gamma\alpha^*$, but $\sigma_f \notin \omega$.

In this case, the occurrence of σ_f contained in s cannot be detected by either of two diagnosers.

Case 2: Take $s = \sigma_f$, then $t = \alpha^*\beta\alpha^*$ satisfies $t \in L/s$ and $|t| \geq n_0 (n_0 \in \mathrm{N})$. If choose the first diagnoser with P_1 to detect the failure event σ_f, then there is no string ω in the G such that $P_1(\omega) = P_1(st) = \alpha^*\beta\alpha^*$, but $\sigma_f \notin \omega$.

However, if choose the second diagnoser with P_2 to detect the failure, then there is a string ω in the G such that $P_2(\omega) = P_2(st) = \alpha^*$, but $\sigma_f \notin \omega$.

In the case, the occurrence of σ_f contained in s can be detected by the first diagnoser.

Since there is a trace $s = \sigma_f$ in G and s is codiagnosable w.r.t. $\{\mathrm{P}_1, \mathrm{P}_2\}$, it is shown that L generated by G is relative codiagnosable w.r.t. $\{\mathrm{P}_1, \mathrm{P}_2\}$ by **Definition 5**. But by **Definition 3**, L is identified as non-codiagnosabe w.r.t. $\{\mathrm{P}_1, \mathrm{P}_2\}$.

Remark 1: Comparing Definition 3 with Definition 5, it is known that relative codiagnosability is weaker than codiagnosability (Qiu and Kumar, 2006) and codiagnosability can be viewed as a special case of the relative codiagnosability.

That is, if the language L is codiagnosable, it must be relatively codiagnosable, but not vice versa.

Remark 2: Relative diagnosability of DESs in the centralized setting (Zhao et al., 2017) can be viewed as a special case of the relative codiagnosability of DESs under the decentralized framework with $m = 1$.

Proposition 1: If there exist $i_0 \in \mathrm{M}$, such that L is relative diagnosable w.r.t. P_{i_0}, then L is relative codiagnosable w.r.t. $\{P_i\}_{i \in M}$.

Proof:

If there is a $i_0 \in \{1, 2, ..., m\}$ such that L is relative diagnosable w.r.t. P_{i_0}, then there exist a trace $s \in \psi(\Sigma_f)$ such that

$$\begin{aligned}&(\exists n_0 \in N)(\forall t \in L/s, |t| \geq n_0) \Rightarrow \\ &(\forall \omega \in L \wedge P_{i_0}(\omega) = P_{i_0}(st))(\Sigma_f \in \omega)\end{aligned} \tag{4}$$

If take $j = i_0$, then there is a $n_0 \in N$ such that

$$\begin{aligned}&(\exists j \in \{1, 2, ..., m\}(\forall t \in L/s, |t| \geq n_0) \Rightarrow \\ &(\forall \omega \in L \wedge P_j(\omega) = P_j(st))(\Sigma_f \in \omega)\end{aligned} \tag{5}$$

By Definition 5, L is relative codiagnosable w.r.t. $\{P_i\}_{i \in M}$.

Definition 6: Let $b \in B_f$ be a branch of G, L' be the language generated by b, and $s \in (L' \cap \psi(\Sigma_f))$. If the trace s

is codiagnosable w.r.t. $\{P_i\}_{i \in M}$, then branch b is said to be codiagnosable w.r.t. $\{P_i\}_{i \in M}$.

Definition 7: Let $|B_f|$ be the number of failure branches in G and k be the number of codiagnosable branches in B_f. Then the codiagnosable rate μ of the language L generated by G is defined as

$$\mu = \frac{k}{|B_f|} \tag{6}$$

Remark 3: If $\mu = 1$, the language L is codiagnosable; If $0<\mu<1$, the language L is relatively codiagnosable; If $\mu = 0$, the language L is not relatively codiagnosable.

3.2 Necessary and Sufficient Conditions of the Relative Codiagnosability

Let $G_d = (Q, \Sigma_o, \delta_d, q_0)$ be the diagnoser (Sampath et al., 1995) of G, where Q is the set of diagnoser states, $\delta_d : Q \times \Sigma_o \to Q$ is the transition function of diagnoser states, and q_0 is the initial diagnoser state. State $q \in Q$ is of the form $q = \{(x_1, l_1), \ldots, ((x_n, l_n)\}$, where $x_i \in X$, $l_i \in \{N, F\}$ and $i \in \{1, 2, \ldots, n\}$. A diagnoser state $q = \{(x_1, l_1), \ldots, (x_m, l_m)\} \in Q$ is: F-certain, if for each $i \in \{1, 2, \ldots, n\}$, $l_j = F$.

Theorem 1: Assume there are m local diagnosers G_{d_i} of G, where $i \in \mathrm{M}$. L is relatively condiagnosable w.r.t. $\{P_i\}_{i \in M}$ iff there exists $i_0 \in \{1, 2, ..., m\}$ such that there is at least a F-certain state in its diagnoser $G_{d_{i_0}}$.

Poof:

Necessity: Assume that L is relatively condiagnosable w.r.t. $\{P_i\}_{i \in M}$. By contradiction it will be shown that there is at least a F-certain state in the $G_{d_{i_0}}$. It is further assumed that there is no F-certain state exists in the diagnoser $G_{d_{i_0}}$. This implies that no any failure events can be detected through the construction of the diagnoser. It follows that L is not relatively condiagnosable w.r.t. $\{P_i\}_{i \in M}$, which is a contradiction to the intended hypothesis. Consequently, if L is relatively condiagnosable, there is at least an F-certain state in its diagnoser $G_{d_{i_0}}$.

Sufficiency: Assume that there exists $i_0 \in M$ such that there is at least a F-certain state in the diagnoser $G_{d_{i_0}}$. Let $s \in \psi(\Sigma_f)$ and $\delta(x_0, s) = x$. Pick any $t_1 \in P_{i_0}(L/s)$, then $\delta(x_0, st_1) = x_1$. Correspondingly, $\delta_{d_{i_0}}(q_0, P_{i_0}(st_1)) = q_1$ in $G_{d_{i_0}}$. Since st_1 contains failure events, it follows that $(x_1, l_1) \in q_1$ with $l_1 = F$. Assume q_1 is F-certain, then $\omega \in P_{i_0}^{-1}[P_{i_0}(st_1)] \Rightarrow \Sigma_f \in \omega$. Therefore, L is relatively condiagnosable.

For the system G described in the Example 1, construct two local diagnosers G_{d_1} and G_{d_2} by $P_i : \Sigma^* \to \Sigma_{o,i}^*$ ($i \in \{1, 2\}$) as that in Fig. 2. and Fig. 3., where $\Sigma_{o,1} = \{\alpha, \beta\}$, $\Sigma_{o,2} = \{\alpha, \gamma\}$. Note that there is a F-certain state $7F$ in the diagnoser G_{d_1}, therefor the language L generated by the system G is relatively codiagnosable w.r.t. $\{P_1, P_2\}$ by the Theorem 1..

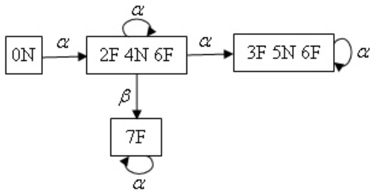


Fig. 2. The local diagnoser G_{d_1}.

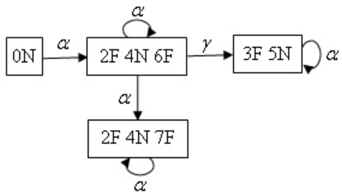


Fig. 3. The local diagnoser G_{d_2}.

4. TEST ALGORITHMS BASED ON THE OPACITY

4.1 Relation between Opacity and Relative Codiagnosability

Over the last decade, opacity as a general information flow property has become a very fertile field of research (Lin, 2011; Saboori and Hadjicostis, 2011, 2012, 2013; Yin and Lafortune, 2017). In order to meet different types of privacy requirements in the context of DESs, several notions of opacity have been studied such as language-based opacity (Lin, 2011), K-step opacity (Saboori and Hadjicostis, 2011), infinite-step opacity (Saboori and Hadjicostis, 2012) and initial-state opacity (Saboori and Hadjicostis, 2013). The opacity in this paper refers to the language-based opacity proposed by Lin (2011).

Definition 8 (Lin, 2011): Given two languages $L_1, L_2 \subseteq L$ and a general observation mapping $\theta : \Sigma^* \to \Sigma^*$, L_1 is strongly opaque with respect to L_2 and θ if $L_1 \subseteq \theta^{-1}\theta(L_2)$; L_1 is weakly opaque with respect to L_2 and θ if $L_1 \cap \theta^{-1}\theta(L_2) \neq \phi$.

The diagnosability was reformulated by the notion of opacity as follows.

Let $L_f = L \cap \Sigma^* \Sigma_f \Sigma^*$ be the set of strings which contain at least one failure event and $L_{\sim f} = L \cap (\Sigma - \Sigma_f)^*$ be the set of strings which do not contain any failure events. It is known that $L = L_{\sim f} \bigcup L_f$. Let $\Sigma^{>n}$ be the set of strings with the length greater than N , then $L_f^{>n} = L \cap \Sigma^* \Sigma_f \Sigma^{>n}$ is the set of all strings which contain a failure event and its subsequent at least n events.

Definition 9 (Lin, 2011): A prefix-closed and live language $L(G)$ is diagnosable with respect to P and Σ_f if there exists a positive integer n such that $(\forall s \in L_f^{>N})$ $(\forall s' \in L(G) \bigcap P^{-1}P(s)) s' \notin L_{\sim f}$.

Likewise, the relative codiagnosability can also be defined by the opacity.

Proposition 2: L is relatively codiagnosable w.r.t. $\{P_i\}_{i\in M}$ and s , if a positive integer n exists such that

$$\begin{aligned}&(\exists j \in M)(s \in L_f^{>n})(\forall s' \in L \cap P_j^{-1} P_j(s))\\ &s' \notin L_{\sim f}\end{aligned} \tag{7}$$

Proof: It is clear by **Definition 9** and **Definition 5**.

Theorem 2: Let $s \in L_f^{>n}$. L is relatively codiagnosable w.r.t. $\{P_i\}_{i\in M}$ and s iff a positive integer n and a $j \in M$ exist such that $L_f^{>n}$ is weak opaque and not strong opaque w.r.t. $L_{\sim f}$ and P_j .

Proof:

$L_f^{>n}$ being weak opaque and not strong opaque with respect to $L_{\sim f}$ and P_j means $L_f^{>n} \bigcap P_j^{-1} P_j(L_{\sim f}) \neq \phi$ and $L_f^{>n} \cap P_j^{-1} P_j(L_{\sim f}) \neq L_f^{>n}$. Therefore,

$L_f^{>n} \bigcap P_j^{-1} P_j(L_{\sim f}) \neq \phi$ and

$$L_f^{>n} \bigcap P_j^{-1} P_j(L_{\sim f}) \neq L_f^{>n} \Leftrightarrow (\exists s \in \Sigma^*) s \in L_f^{>n} \wedge s \notin P_j^{-1} P_j(L_{\sim f})$$

$$\Leftrightarrow (\exists s \in \Sigma^*) s \in L_f^{>n} \wedge (\forall s' \in \Sigma^*) P_j(s) = P_j(s') \wedge s' \notin L_{\sim f}$$

$$\Leftrightarrow (\exists s \in \Sigma^*)(\forall s' \in \Sigma^*) P_j(s) = P_j(s') \wedge s \in L_f^{>n} \wedge s' \notin L_{\sim f}$$

$$\Leftrightarrow (\exists s \in L_f^{>n})(\forall s' \in L \cap P_j^{-1} P_j(s)) s' \notin L_{\sim f}$$

According to **Proposition 2**, it follows that L is relatively codiagnosable with respect to $\{P_i\}_{i\in M}$ and s .

4.2 An Test Algorithm for Relative Codiagnosability

According to **Theorem 1**, relative codiagnosability of a DES can be verified by constructing multiple local diagnosers. However, in this way, the value of the codiagnosable rate cannot be obtained. That is, it is not possible to compare the relative codiagnosability of different systems. Therefore, a new algorithm is developed to solve the problem.

Algorithm 1:

Let B_f be the set of failure branches, $B_{\sim f}$ be the set of normal branches and k be the number of branches that are codiagnosable in the B_f .

Initialization:

$B_f = \phi$, $B_{\sim f} = \phi$, $k = 0$.

1) Compute the value of $\left|B_f\right|$:

According to **definition 2**, find all branches in G first. Then divide these branches into B_f and $B_{\sim f}$. Let $\left|B_f\right|$ denote the number of branches in the B_f , compute the value of $\left|B_f\right|$.

2) Construct automata for each failure branch after projections.

Let B_f contain n failure branches, i.e. $B_f = \{b_1, b_2, ..., b_n\}$. For each failure branch after projections $P_j(L(b_i))$ ($i \in \{1,2,3,...,n\}$ and $j \in M$), construct automaton $G_{1i}^{j'} = (X_{1i}^j, \Sigma_{1i}^j, \delta_{1i}^{j'}, x_0)$, where X_{1i}^j is the set of initial state x_0 and states in which at least one observable event arrives, Σ_{1i}^j is the set of observable events contained in b_i and $\delta_{1i}^{j'} : X_{1i}^j \times \Sigma_{1i}^j \to X_{1i}^j$ is the transition function.

3) Construct automata for all normal branches after projections.

Let $B_{\sim f}$ contain l failure branches, i.e. $B_f = \{b_1, b_2, ..., b_l\}$. For each normal branch after projections $P_j(L(b_i))$ ($i \in \{1,2,3,...,l\}$ and $j \in M$), construct automaton $G_{2i}^{j'} = (X_{2i}^j, \Sigma_{2i}^j, \delta_{2i}^{j'}, x_0)$, where X_{2i}^j is the set of initial state x_0 and states in which at least one observable event arrives and Σ_{2i}^j is the set of observable events contained in the $b_{\sim i}$. $\delta_{2i}^{j'} : X_{2i}^j \times \Sigma_{2i}^j \to X_{2i}^j$ denotes the transition function.

The automaton $G_2^{j'}$ of $P_j(L_{\sim f})$ can be computed by $G_{2i}^{j'}$: $G_2^{j'} = (X_2^{j'}, \Sigma_2^{j'}, \delta_2^{j'}, x_0)$, where $X_2^{j'} = \bigcup X_{2i}^j$, $\Sigma_2^{j'} = \bigcup \Sigma_{2i}^j$ ($i \in \{1,2,3,...,l\}$), and $\delta_2^{j'} : X_2^{j'} \times \Sigma_2^{j'} \to X_2^{j'}$ is the transition function.

4) Verify if $P_j(L(b_i)) \bigcap P_j(L_{\sim f}) \neq P_j(L(b_i))$.

For verifying if $P_j(L(b_i)) \bigcap P_j(L_{\sim f}) \neq P_j(L(b_i))$, the product of $G_{1i}^{j'}$ and $G_2^{j'}$ will be constructed.

Let $G_{3i}^j = (X_{3i}^j, \Sigma, \delta_{3i}^j, x_0)$, then

$G_{3i}^j = (G_{1i}^{j'} \times G_2^{j'}) = (X_{oi}^j \times X_o^j, \Sigma_o^j, \delta_{1i}^{j'} \times \delta_2^{j'}, (x_0, x_0))$,

$L(G_{3i}^j) = L(G_{1i})^{j'} \bigcap L(G_2^{j'}) = P_j(L(b_i)) \bigcap P_j(L_{\sim f})$

For each $i \in \{1,2,3,...,n\}$, check if there exists a j such that $P_j(L(b_i)) \bigcap P_j(L_{\sim f}) \neq P_j(L(b_i))$. If exists, then $k = k+1$.

5) Compute the value of the codiagnosable rate.

The codiagnosable rate: $\mu = k / \left|B_f\right|$.

if $\mu = 1$: $L(G)$ is codiagnosable;

if $0 < \mu < 1$: $L(G)$ is relatively codiagnosable;

if $\mu = 0$: $L(G)$ is not relatively codiagnosable.

Now, discuss the computational complexity of the algorithm.

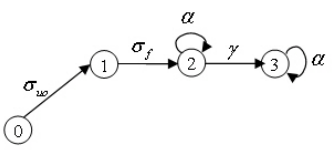
For the system $G = (X, \Sigma, \delta, x_0)$, the number of feasible transitions from a reachable state $x \in X$ is $|\Sigma|$ in the worst case. So the computational complexity of the first step is $O(|Q||\Sigma|)$. In the second step, because automata $G_{1i}^{j'}$ has $\left|X_{1i}^j\right|$ states and every state has $\left|\Sigma_{1i}^j\right|$ feasible transitions at most, so the computational complexity of construction of $G_{1i}^{j'}$ is $O(\left|X_{1i}^j\right|\left|\Sigma_{1i}^j\right|)$. Similarly for the third step, the computational complexity of the construction of $G_{2j}^{j'}$ is $O(\left|X_2^{j'}\right|\left|\Sigma_2^{j'}\right|)$. The computational complexity of product in the fourth step is $O(\left|X_{1i}^j\right|\left|X_2^j\right|\left|\Sigma_2^j\right|)$.

Overall, the computational complexity of the algorithm is polynomial.

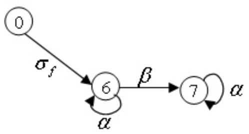
4.3 Illustrative Examples

Example 2: Consider again G shown in Fig.1 again. Example 1 shows that that L is relative codiagnosable w.r.t. $\{P_1, P_2\}$ by using the Definition Approach. In the following, we verify the conclusion by **Algorithm 1**.

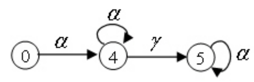
The first step of the algorithm is to compute the value of $\left|B_f\right|$. As Fig.1 shows,, the system G has three branches which are shown in Fig. 4 (a), (b) and (c) respectively. Notice that b_1 and b_2 contain a fault event, but b_3 contains no fault events, so $B_f = \{b_1, b_2\}$ and $B_{\sim f} = \{b_3\}$. Obviously, there are two branches in B_f, therefore $\left|B_f\right| = 2$.



(a) *The failure branch* b_1.



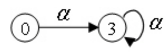
(b) *The failure branch* b_2.



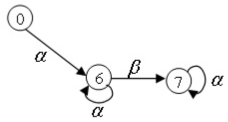
(c) *The normal branch* b_3 *or* $G_2^{2'}$ *of* $P_2(L_{\sim f})$.

Fig. 4. Branches of G.

According to the second step of the algorithm, automata $G_{11}^{1'}$ and $G_{12}^{1'}$ for $P_1(L(b_1))$ and $P_1(L(b_2))$ are constructed as that in Fig. 5.(a) and (b), respectively. Automata $G_{11}^{2'}$ and $G_{12}^{2'}$ for $P_2(L(b_1))$ and $P_2(L(b_2))$ are constructed as that in Fig. 6.(a) and (b), respectively.

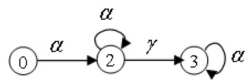


(a) Automaton $G_{11}^{1'}$ of $P_1(L(b_1))$.

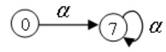


(b) Automaton $G_{12}^{1'}$ of $P_1(L(b_2))$.

Fig. 5. Automata $G_{1i}^{1'}$ ($i \in \{1,2\}$).



(a) Automaton $G_{11}^{2'}$ of $P_2(L(b_1))$.



(b) Automaton G_{12}^{2} of $P_2(L(b_2))$.

Fig. 6. Automata G_{1i}^{2} ($i \in \{1,2\}$).

Next by Step 3, construct automata $G_2^{1'}$ and $G_2^{2'}$ for $P_1(L_{\sim f})$ and $P_2(L_{\sim f})$, which are shown in Fig. 7. and Fig. 4.(c), respectively.

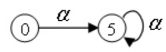
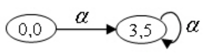


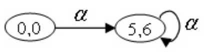
Fig. 7. Automaton $G_2^{1'}$.

Now determine whether $P_1(L(b_i)) \bigcap P_1(L_{\sim f}) \neq P_1(L(b_i))$ based on step 4 of the algorithm. To this end, automaton G_{31}^{1} (i.e., the product of $G_{11}^{1'}$ and $G_2^{1'}$) and automaton G_{32}^{1} (i.e., the product of $G_{12}^{1'}$ and $G_2^{1'}$) are constructed as that depicted by Fig. 8.(a) and Fig. 8.(b). As Fig. 8.(a) shows, $P_1(L_{11}^{1'}) \bigcap P_1(L_2^{'}) = P_1(L_{11}^{'})$, i.e., $P_1(L(b_1)) \bigcap P_1(L_{\sim f}) = P_1(L(b_1))$. But as shown in Fig. 8.(b),

$P_1(L_{12}^{1'}) \cap P_1(L_2^{1'}) \neq P_1(L_{12}^{1'})$ i.e., $P_1(L(b_2)) \bigcap P_1(L_{\sim f}) \neq P_1(L(b_2))$.



(a) Automaton G_{31}^{1} .

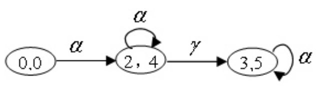


(b) Automaton G_{32}^{1} .

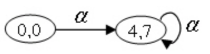
Fig. 8. Automata $G_{3i}^{1}(i=1,2)$.

Likewise, to determine whether $P_2(L(b_i)) \bigcap P_2(L_{\sim f}) \neq P_2(L(b_i))$, automaton G_{31}^{2} (i.e., the product of $G_{11}^{2'}$ and $G_2^{2'}$) and automaton G_{32}^{2} (i.e., the product of $G_{12}^{2'}$ and $G_2^{2'}$) are constructed as that in Fig. 9(a) and Fig. 9(b). As Fig. 9.(a) shows, $P_2(L_{11}^{2'}) \bigcap P_2(L_2^{'}) = P_2(L_{12}^{'})$, i.e., $P_2(L(b_1)) \bigcap P_2(L_{\sim f}) = P_2(L(b_1))$. From Fig. 9.(b), it is known that $P_2(L_{12}^{1'}) \bigcap P_2(L_2^{1'}) = P_2(L_{12}^{1'})$ i.e., $P_2(L(b_2)) \bigcap P_2(L_{\sim f}) = P_2(L(b_2))$.

Therefore, the value of k is 1.



(a) Automaton G_{31}^{2}



(b) Automaton G_{32}^{2}

Fig. 9. Automata $G_{3i}^{2}(i=1,2)$.

The final step is to get the value of the codiagnosable rate. It is clear that the codiagnosable rate $\mu = k / \left| B_f \right| = 1/2 = 0.5$. $0 < \mu < 1$, therefore we have the same conclusion as Example 1 that L is relatively codiagnosable w.r.t. $\{P_1, P_2\}$.

In addition, the result shows that half of the failure events in the system G can be diagnosed by the diagnoser approach proposed in (Sampath et al., 1995). However, according to (Qiu and Kumar, 2006), the language L of G is identified as non-codiagnosable, so the diagnoser approach cannot be used to detect the failure events in the G . In fact, if the codiagnosable rate μ of a language is closer to 1, the diagnoser method can be used to diagnose most of the failure events of the system.

Example 2: Consider the plant G depicted as Fig. 10. It is a DES model of the decentralized delivery manufacturing system shown in Fig. 11., which describes the situation that a robot delivers a batch of semi-finished products from Workshop 1 to two Test stations and another three Workshops.

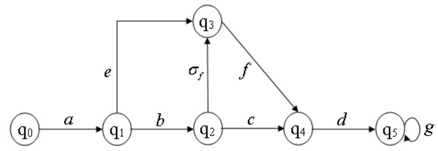


Fig. 10. DES in Example 2.

In Fig. 11., events are eight execution instructions:

a : Teststation1 of Factory1 (F1:Teststation1 for short) receives the products from Workshop1 of Factory1 (F1:Workshop1 for short);

b : Workshop1 of Factory2 (F2:Workshop1 for short) receives the products from F1:Teststation1;

c : Teststation1 of Factory2 (F2:Teststation1 for short) receives the products from F2:Workshop1;

d : Workshop3 of Factory1 (F1:Workshop3 for short) receives the products from F2:Teststation1;

e : Workshop2 of Factory1 (F1:Workshop2 for short) receives the products from F1:Teststation1;;

f : F2:Teststation1 receives the products from F1:Workshop2;

g : Deliver the products from F1:Workshop3 to F1:Workshop3;

σ_f : The products were delivered from F2:Workshop1 to F1:Workshop2 by mistake.

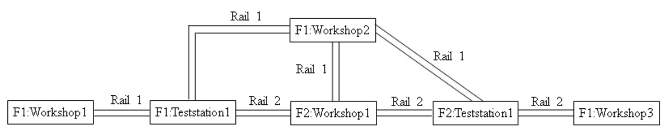


Fig. 11. A delivery manufacturing system.

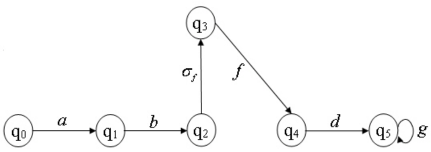
Initially, if the robot executes instruction 1 (i.e., a), then it will travel on Rail 1 and deliver the semi-finished products from F1:Workshop1 (i.e., state q_0) to F1:Teststation1 (i.e., state q_1) for detecting. After that, the robot picks up the products to F2:Workshop1 (i.e., state q_2) or F1:Workshop2 (i.e., state q_3) for further processing under instruction 2 (i.e., b) or instruction 5 (i.e., e) and then to F2:Test Station 1 (i.e., state q_4) for further detecting after executing instruction 3 (i.e., c) or instruction 6 (i.e., f). Finally, the robot will deliver the products from F2:Teststation1 to F1:Workshop3 (i.e., state q_5) for using.

If instruction 3 is executed by mistake at the F2:Workshop1 (i.e., σ_f occurs), the robot will travel on Rail 1 and deliver the products from F2:Workshop1 to F1:Workshop2, where the products will be processed again. Then, the robot picks up the products to Test Station 2 for further detecting under instruction 6 (i.e., f).

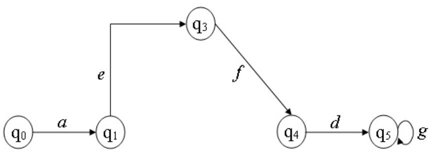
Because instruction 3 (i.e. c) has nothing to do with Factory1, so it is not visible to the Factory1. And instruction 1 (i.e. a) and instruction 5 (i.e. e) have nothing to do with Factory2, so they are not visible to the Factory2. In addition, failure event σ_f are also not visible to Factory1 and Factory2. Here, assume that there are two local projections $P_i : \Sigma^* \rightarrow \Sigma^*_{o,i} (i \in \{1,2\})$, where $\Sigma_{o,1} = \{a,b,d,e,f,g\}$, $\Sigma_{o,2} = \{b,c,d,f,g\}$.

Next verify the codiagnosability of the system through **Algorithm 1**.

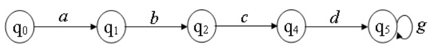
As Fig. 10 shows, the system has three branches which are shown in Fig. 12.(a), Fig. 12.(b) and Fig. 12.(c) respectively. $B_f = \{b_1\}$, therefore $|B_f| = 1$.



(a) The failure branch b_1



(b) The failure branch b_2 (i.e. $G_{21}^{1'}$)



(c) The normal branch b_3

Fig. 12. Branches of the system.

Automata $G_{11}^{1'}$ for $P_1(L(b_1))$ and $G_{11}^{2'}$ for $P_2(L(b_1))$ are constructed as that in Fig. 13. and Fig. 14. Similarly, automata $G_{21}^{1'}$ and $G_{21}^{2'}$ for $P_1(L(b_2))$ and $P_2(L(b_2))$ are constructed as that in Fig. 12(b) and Fig. 15. And automata $G_{22}^{1'}$ and $G_{22}^{2'}$ for $P_1(L(b_3))$ and $P_2(L(b_3))$ are constructed as that in Fig. 16(a) and Fig. 16(b).

According to $G_{21}^{1'}$ and $G_{22}^{1'}$, automaton $G_2^{1'}$ are constructed as that in Fig. 17. Likewise, according to $G_{21}^{2'}$ and $G_{22}^{2'}$, automaton $G_2^{2'}$ are constructed as that in Fig. 18.

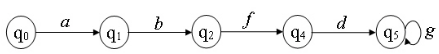


Fig. 13. Automaton $G_{11}^{1'}$ of $P_1(L(b_1))$.

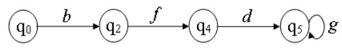


Fig. 14. Automaton $G_{11}^{2'}$ of $P_2(L(b_1))$.

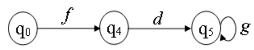
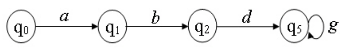
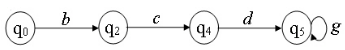


Fig. 15. Automaton $G_{21}^{2'}$.



(a) Automaton $G_{22}^{1'}$ or $G_{32}^{1'}$



(b) Automaton $G_{22}^{2'}$

Fig. 16. Automata $G_{22}^{j'}$ ($j \in \{1,2\}$).

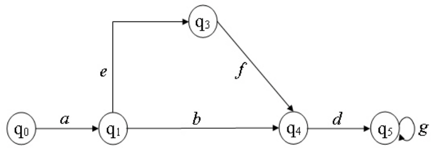


Fig. 17. Automaton $G_2^{1'}$.

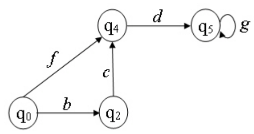


Fig. 18. Automaton $G_2^{2'}$.

Automaton $G_{31}^{1'}$ (i.e., the product of $G_{11}^{1'}$ and $G_2^{1'}$) is constructed as that in Fig. 19. As Fig. 19 shows, $P_1(L_1^{'}) \cap P_1(L_2^{'}) \neq P_1(L_1^{'})$ i.e., $P_1(L(b_1)) \cap P_1(L_{\sim f}) \neq P_1(L(b_1))$. Automaton $G_{31}^{2'}$ (i.e., the product of $G_{11}^{2'}$ and $G_2^{2'}$) is constructed as that in Fig. 20. As we can see from Fig.20, $P_2(L_1^{'}) \cap P_2(L_2^{'}) \neq P_2(L_1^{'})$ i.e., $P_2(L(b_1)) \cap P_2(L_{\sim f}) \neq P_2(L(b_1))$. Therefore, the value of k is 1.

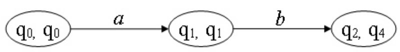


Fig. 19. Automaton $G_{31}^{1'}$.

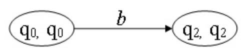


Fig. 20. Automaton $G_{31}^{2'}$.

The codiagnosable rate $\mu = k / |B_f| = 1$. The result indicates that L is codiagnosable. And the example shows that the algorithm is also applicable for the verification of codiagnosability introduced by (Qiu and Kumar, 2006).

5. CONCLUSIONS

In this paper, we investigated the relative diagnosability of the decentralized failure diagnosis system. The notion of relative codiagnosability was formalized, which is weaker than codiagnosability and relative diagnosability. In order to verify whether a system is relatively codiagnosable, the necessary and sufficient conditions of relative codiagnosability were presented. Moreover, an opacity-based algorithm was proposed for calculating the codiagnosable rate and its complexity is polynomial. The reported work generalizes the main results of Qiu (2006). Further issue worthy of consideration is relative diagnosability of fuzzy DESs. We would like to consider it in subsequent work.

REFERENCES

Basile, F., Chiacchio, P., and De Tommasi, G. (2009). An efficient approach for online diagnosis of discrete event systems. *IEEE Trans. on Automatic Control*, 54(4), 784–759.

Biswas, S. (2012). Diagnosability of discrete event systems for temporary failures. *Computers and Electrical Engineering,* 38, 1534–1549.

Chen, J., and Kumar, R. (2013). Decentralized failure diagnosis of stochastic discrete event systems. *IEEE International Conference on Automation Science and Engineering (CASE)*, Kuching, 1083–1088.

Carvalho, L. K., Moreira, M. V., Basilio, J. C., and Lafortune, S. (2013). Robust diagnosis of discrete-event systems against permanent loss. *Automatica*, 49, 223–231.

Chen, Z., Lin, F., Wang, C., Wang, L., and Xu, M. (2014). Active diagnosability of discrete event systems and its application to battery fault diagnosis. *IEEE Trans. on Control System Technology*, 22(5), 1892–1898.

Cabral, F. G., Moreira, M. V., Diene, O., and Basilio, J. C. (2015). A Petri Net Diagnoser for Discrete Event Systems Modeled by Finite State Automata. *IEEE Trans. on Automatic Control*, 60(1), 59–70.

Deng, W., and Qiu, D. (2016). State-based Fault Diagnosis of Discrete-event Systems. *2016 28th Chinese Control and Decision Conference (CCDC)*, 5470–5475.

Deng, W., and Qiu, D. (2017). State-Based decentralized diagnosis of bi-fuzzy discrete event systems. *IEEE Transactions on Fuzzy Systems*, 25(4), 854–867.

Geng, X., Ouyang D., and Zhang, Y. (2017). Model-based diagnosis of incomplete discrete-event system with rough set theory. *Science China Information Sciences*, 60, 1–10.

Li, H., Gao, Y., Wu, L., and Lam, H. (2015). Fault detection for T-S fuzzy time-delay systems: delta operator and input-output methods. *IEEE Trans. on Cybernetics*, 45(2), 229–241.

Lin, F. (2011). Opacity of discrete event systems and its applications. *Automatica*, 47, 496–503.

Liu, F., Qiu, D., Xing, H., and Fan, Z. (2008). Decentralized diagnosis of stochastic discrete event system. *IEEE Trans. on Automatic Control*, 53(2), 535–546.

Liu, F., and Qiu, D. (2009). Diagnosability of fuzzy discrete-event systems. *IEEE Trans. on Automatic Control*, 17(2), 372–383.

Liu, F. (2014). Polynomial-time verification of diagnosability of fuzzy discrete event systems. *Science China Information Sciences*, 57(6), 1–10.

Liu, F. (2015). Safe diagnosability of fuzzy discrete-event systems and a polynomial-time verification. *IEEE Trans. on Fuzzy System*, 23(5), 1534–1544.

Liu, F. (2018). Reliable predictability of failure events for decentralized discrete-event systems. *Proceedings of the 37th Chinese Control Conference July 25-27, 2018, Wuhan, China*, 2048-2053.

Liu, F. (2019). Predictability of failure event occurrences in decentralized discrete-event systems and polynomial-time verification. *IEEE Trans. on Automation Science and Engineering*, 16(1), 498–504.

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

Perez-Zuniga, C. G., Chanthery, E., Trav´ e-Massuy` es, L., Satomayor, J., and Artigues, C. (2018). Decentralized diagnosis via Structural Analysis and Integer Programming. *Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2018 Warsaw*, Poland, 29C31 August 2018, 51(24), 168–175.

Qiu, W., and Kumar, R. (2006). Decentralized failure iagnosis of discrete event systems. *IEEE Trans. on Systems, Man, and Cybernetics-Part A: Systems and Humans,* vol. 36(2), 384–395.

Ran, N., Su, H., and Giua, A. (2018). Decentralized failure diagnosis of discrete event systems. *IEEE Trans. on Automatic Control*, 63(4), 1192-1199.

Saboori, A., and Hadjicostis, C. N. (2011). Verification of K-Step Opacity and Analysis of Its Complexity. *IEEE Trans. on Automation Science and Engineering*, 8(3), 549–559.

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

Saboori, A., and Hadjicostis, C. N. (2013). Verification of initial-state opacity in security application of discrete event systems. *IEEE Trans. on Automatic Control*, 57(5), 115–132.

Saboori, A., and Hadjicostis, C. N. (2014). Current-State opacity formulations in probabilistic finite automata. *IEEE Trans. on Automatic Control*, 59(1), 120–133.

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

Sayed-Mouchaweh, M., and Lughofer, E. (2015). Decentrallzed fault diagnosis approach without a global model for fault diagnosis of discrete event systems. *International Journal of Control*, 88, 2228–2241.

Thorsley, D., and Teneketzis, D. (2005). Diagnosability of stochastic discrete-Event systems. *IEEE Trans. on Automatic Control*, 50(4), 476–492.

Wu, Y., and Lafortune, S. (2016). Synthesis of optimal insertion functions for opacity enforcement. *IEEE Trans. on Automatic Control*, 61(3), 571–584.

Yao, L., and Feng, L. (2016). Fault diagnosis and fault tolerant control for non-Gaussian time-delayed singular stochastic Distribution Systems. *International Journal of Control, Automation and Systems*, 14(2), 35–442.

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

Yin, X., and Lafortune, S. (2017). Minimization of sensor activation in decentralized discrete event systems. *IEEE Trans. on Automatic Control, Citation information*: DOI 10.1109/TAC.2017.2783048, 2017.

Zhao, R., Liu, F., and Liu, Z. (2017). Relative diagnosability of discrete-event systems and its opacity-based test algorithm. *International Journal of Control, Automation and Systems*, 15(4), 1693–1700.

Zhao, R., Liu, F., and Tan, J. (2019). Relative predictability of failure event occurrences and its opacity-based test algorithm. *International Journal of Control*, 92(7), 1600–1608.

Zhang, B., Shu, S., and Lin, F. (2015). Maximum information release while ensuring opacity in discrete event systems. *IEEE Trans. on Automation Science and Engineering,* 12(3), 1067–1079.