R-step Relative Predictability of Decentralized Failure Prognosis in Discrete-Event Systems

Rui Zhao, Fuchun Liu

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

Abstract: Recently, failure prediction of discrete-event systems (DESs) has received increasing attention. In this paper, the problem of relative predictability of decentralized failure prognosis in DESs is investigated. The notion of r-step relative copredictability is formalized under the decentralized framework to capture the feature that the occurrence of at least a failure event can be predicted prior to r steps at most based on at least one local observation. It is deducted that the relative copredictability is weaker than copredictability and relative predictability. In order to achieve the prediction performance of a decentralized system, the necessary and sufficient condition for verifying the relative copredictability is presented. And a polynomial-complexity algorithm is developed to test the relative copredictability and compute the boundary number of steps prior to the occurrences of failure events. Furthermore, some examples are provided to illustrate the presented results. It is worth noting that the reported work generalizes the main results of relative predictability from the centralized systems to the decentralized setting and extends the results of copredictability introduced by Fuchun Liu to general cases.

Keywords: predictability, discrete event systems, decentralized failure prognosis.

1. INTRODUCTION

This paper investigates the relative predictability of decentralized failure prognosis of DESs. The diagnosis of DESs which aims to timely detect the failures that cannot be observed directly according to the local behaviors of the system has been widely investigated (Sampath et al., 1995; Liu, 2015; Yao and Feng, 2016; Deng and Qiu, 2017; Zhao et al., 2017; Keroglou and Hadjicostis, 2018; Masopust and Yin, 2019; Vianaand and Basilio, 2019) during the past two decades. The study of failure prediction (or prognosis) is inspired by the problem of failure diagnosis. Compared with failure diagnosis that focuses on identifying the failures after their occurrences, failure prediction is to accurately forecast failures prior to their occurrences based on the partial observation of the system. Recently, failure prediction of DESs has received increasing attention and many different approaches focused on the centralized framework (Genc and Lafortune, 2009; Takai, 2012; Chang et al., 2013; Chen and Kumar, 2014; Yokotani et al., 2016) have been proposed, where there is a single site for collecting all the information about the system and there is only a prognoser performing failure prediction. Genc and Lafortune initiated the study of predictability of DESs and provided its formal definition and verification method (Genc and Lafortune, 2009). Takai discussed the robustness of failure prediction and introduced a robust prognoser to predict the occurrences of failures for given a set of possible DES models. The notion of AAS-predictability was introduced by Chang, Dong et al. and a necessary and sufficient condition for the property was presented (Chang et al., 2013). The framework introduced by Genc and Lafortune (Genc and Lafortune, 2009) was generalized to the case of stochastic models in (Chen and Kumar, 2014), and the notion of Sm-prognosability was formulated to capture the ability of predicting the occurrences of failures prior to at least m steps in the setting of stochastic DESs. Yokotani et al. developed a theoretical framework for abstraction-based failure prognosis of partially observed DESs (Yokotani et al., 2016).

However, for many complex large-scale systems, the centralized failure prediction may not always be appropriate, and instead failure prediction needs to be performed by decentralized sites where prediction information is collected. In the decentralized setting, there is a family of local prognosers running at several sites processing local observation. Each local prognoser may only observe part of the dynamic behavior of the system and make local prognostic decision based on its own observation. In (Takai and Kumar, 2011, 2012; Yin and Li, 2016, 2019; Liu, 2019), decentralized failure prediction under different architectures was studied. The decentralized prognoser defined by a map to prognostic decision "1" or "φ" was employed to perform coprediction (Takai and Kumar, 2011), where "1" means a failure is inevitable and"φ" means either a failure is not inevitable or its inevitability is not known. The notion of joint prognosability was introduced by Takai to describe the property that the occurrence of each failure can be predicted by at least one local prognoser (Takai and Kumar, 2012). The notion of k-reliable coprognosability was proposed (Yin and Li, 2016) as the necessary and sufficient condition for the existence of a decentralized prognoser under the presence of unreliable local prognostic decisions. Yin and Li proposed two novel decentralized protocols (Yin and Li, 2019) for the purpose of fault prognosis. In (Liu, 2019), the notion of copredictability of DESs was formalized to capture the feature that the occurrences of failure events can be predicted in advance based on at least one local observation and the coverifier-based approach was developed to verify the predictability.

Inspired by the wide-spread adaptability of decentralized framework, the paper aims to deal with relative predictability of DESs in the setting of decentralized framework. The work is a continuation of the previous work related to relative predictability of DESs (Zhao et al., 2019). It also draws on the study of the work (Liu, 2019) and its main contributions are as follows: 1) formalizing the notion of r-step relative copredictability of DESs; 2) introducing some new concepts such as failure branch, predictive vector and step vector; 3) deriving the necessary and sufficient condition of the property; 4) designing a polynomial-time verification algorithm.

The results presented in the article are mainly related to the work of following references: Takai and Kumar, 2012, Yin and Li, 2019, Liu, 2019 and Zhao et al., 2019. However, the distributed prognosis discussed by Takai (Takai and Kumar, 2012), Yin (Yin and Li, 2019) and Liu (Liu, 2019) is a completely joint predictable property, which is clearly different from the relative copredictability introduced by this paper. The completely joint predictable property requires that the occurrence of each failure event can be predicted by at least one local site, whereas relative copredictability only requires that the occurrence of a failure event can be predicted by at least one local observation. It should be pointed out that the authors of this article (Zhao et al., 2019) have studied the problem of relative predictability of failure event occurrences, but we dealt with the issue under the centralized framework. This paper extends these results to distributed framework. Moreover, the notion of failure branch is redefined so that it can be applied to more general DESs; the concept of predictive vector is introduced to accurately describe the relative copredictability of a DES. Through the value of the predictive vector, which failure branches in the system are copredictable can be known clearly. In addition, this paper presents a polynomial-time algorithm for accurately computing the predictive vector and the boundary number of steps prior to the occurrences of failure events. It's worth noting that the algorithm is also used to verifying copredictability (Liu, 2019) and relative predictability (Zhao et al., 2019). Therefore, the research results of this paper have wide applicability.

The remainder of the paper is organized as follows. In section 2, the necessary background on DESs is presented. In section 3, the notions of r-step relative copredictability is introduced. In section 4, an algorithm is presented for verifying the property. Finally, in section 5, conclusions are drawn.

2. PRELIMINARIES

A DES is modeled as an automaton $G=(Q,\Sigma,\delta,q_0)$, where Q is the finite set of states, Σ is the set of events, $\delta: Q\times\Sigma\to Q$ is the transition function and $q_0\in Q$ is the initial state. Σ^* is the set of all finite strings over Σ, including the empty string ϵ. Given an event $\sigma\in\Sigma$ and a string $s\in\Sigma^*$, if σ appears at least once in s, then denote it as $\sigma\in s$. The language of G denoted by $L(G)$ or L is defined as $L=\{s\in\Sigma^*:\delta(q_0,s)\in Q\}$. Given a trace $s(s\in L)$, $\overline{s}$ is the the prefix-closure of s, $L/s=\{t\in\Sigma^*: st\in L\}$ denotes the post language of L after s and $\|s\|$ represents the length of s. Under partial observation, the event set is partitioned into two disjoint subsets $\Sigma=\Sigma_o\cup\Sigma_{uo}$, where Σ_o and Σ_{uo} are respectively the observable and unobservable event sets. When a string of events occurs, the sequence of observable events is filtered by the usual projection $P:\Sigma^*\to\Sigma_o^*$. And the inverse projection is $P_L^{-1}(y)_{y\in\Sigma_o^*}=\{s\in L: P(s)=y\}$.

Let $\Sigma_f=\{\sigma_f\}\subseteq\Sigma$ denote the set of failure events which are to be predicted. For set Σ_f, $|\Sigma_f|$ represents the number of failure events contained in the set.

$L_f=L\cap\Sigma^*\Sigma_f$ denotes the set of all traces of L that end in a failure event and $L_{\sim f}=L\cap(\Sigma-\Sigma_f)^*$ denotes the set of all traces in L containing no failure events. Define $L'_{\sim f}=\{s\in L_{\sim f}:(\exists t\in L/s)\sigma_f\in t\}$ as a subset of $L_{\sim f}$ in which the post language of L after s contains failure events and $L''_{\sim f}=\{s\in L_{\sim f}:(\forall t\in L/s)\ \sigma_f\notin t\}$ as another subset of $L_{\sim f}$ in which the post language of L after s contains no failure events. $L_{\sim f}=L'_{\sim f}\cup L''_{\sim f}$.

Let q be a state of G. The feasible event set of q is denoted by $\Gamma(q)=\{\sigma\in\Sigma:\delta(q,\sigma)\in Q\}$.

Definition 1 (Liu, 2019): Let L be the language generated by plant G with a failure event set Σ_f. Assume there are m local projections $P_i:\Sigma^*\to\Sigma_{o,i}^*$, where $i\in I$ and $I=\{1,2,...,m\}$. L is said to be copredictable w.r.t. Σ_f and $\{P_i\}_{i\in I}$ if

$$(\exists n\in\mathbb{N})(\forall s\in L_f)(\exists j\in I)(\exists t\in\overline{s}\cap L_{\sim f})[\mathbb{R}_j(t)=1],$$

where $\mathbb{N}$ is the set of natural number and the condition $\mathbb{R}_j(t)$ is defined as follows:

$$\mathbb{R}_j(t)=\begin{cases}1 & if\,(\forall v\in L/(P_j^{-1}(P_j(t))\cap L_{\sim f}))(\|v\|\geq n\Rightarrow(\sigma_f\in v))\\ 0 & otherwise\end{cases}\tag{1}$$

Intuitively, L being copredictable means that for each failure event, there is at least one location to predict its future occurrence by limited observable events.

3. R-STEP RELATIVE COPREDICTABILITY

In this section, the definitions of relative copredictability and predictive vector are given firstly. Then based on the definitions, the formal description of r-step relative copredictability is introduced.

3.1. Definition of relative copredictability

Let $G=(Q,\Sigma,\delta,q_0)$ be a plant with a failure event set Σ_f. Assume there are m local projections $P_i:\Sigma^*\to\Sigma_{o,i}^*$, where $i\in I$ and $I=\{1,2,...,m\}$. The m local projections are supposed to be independent, namely, without communicating their observation to each other.

Definition 2: Let L be the language generated by G with Σ_f. A trace $s \in L_f$ is said to be copredictable w.r.t. Σ_f and $\{P_i\}_{i\in I}$ if

$$(\exists n \in \mathbb{N})(\exists t \in \overline{s} \cap L_{\sim f})(\exists j \in I)(\forall v \in L/(P_j^{-1}(P_j(t)) \cap L_{\sim f}))$$
$$[\|v\| \geq n \Rightarrow (\sigma_f \in v)].$$

Remark 1: Here, t is referred to as predictive prefix of s. Suppose $t_1, t_2, ..., t_l$ are predictive prefixes of s w.r.t. P_j, where $l \in \mathbb{N}$. If for i=1, 2,..., l, $\exists k \in \{1,2,...,l\}$ such that $\|t_k\| \leq \|t_i\|$, then t_k is called the shortest predictive prefix of s w.r.t. P_j, denoted by $t^j(s)$.

Definition 3: Let L be the language generated by G with Σ_f. If there is a trace s $(s \in L_f)$ and s is copredictable w.r.t. Σ_f and $\{P_i\}_{i\in I}$, then L is said to be relatively copredictable w.r.t. Σ_f and $\{P_i\}_{i\in I}$.

Intuitively, L being relative copredictable means that there is at least one failure trace where the future occurrences of failure events can be inferred based on at least one local observation.

Remark 2: Relative predictability (Zhao et al., 2019) of DESs can be viewed as a special case of the relative copredictability with one location (i.e. $m=1$).

Remark 3: Comparing with Definition 1, it is deduced that copredictability introduced by Liu is a special case of the relative copredictability. If L is copredictable, it must be relatively copredictable, but not vise versa.

Example 1: Consider the plant G described by Fig.1, where $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}\}$, q_0 is the initial state, $\Sigma = \{a,b,c,d,u,f\}$ and $\Sigma_f = \{f\}$ is a failure event set. Assume that there are two local projections $P_i : \Sigma^* \to \Sigma_{o,i}^*$, where $i = 1,2$, and $\Sigma_{o,1} = \{a,b,c\}$, $\Sigma_{o,2} = \{a,c,d\}$.

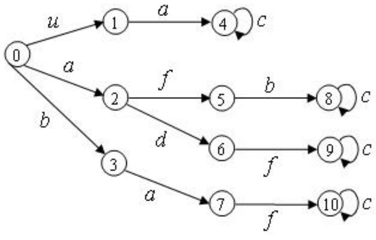


Fig. 1. Plant G of Example 1.

Now the property of copredictability of G will be analysed.

Case 1: Take $s = af$ $(s \in L_f)$ and $t = a$ $(t \in (\overline{s} \cap L_{\sim f}))$. If $j = 1$, then for any $n \in \mathbb{N}$, there is $ua \in (P_1^{-1}(P_1(t)) \cap L_{\sim f})$ and $v = cc^n$ such that $v \in L/ua$ and $\|v\| \geq n$, but $f \notin v$. So s is not predictable w.r.t. Σ_f and P_1. Similarly, it is known that s is not predictable w.r.t. Σ_f and P_2.

In this case, the occurrence of event f contained in s can not be predicted by either of two locations. So it is known that $s = af$ is not copredictable w.r.t. Σ_f and $\{P_i\}_{i\in\{1,2\}}$ by Definition 2.

Case 2: Take $s = adf$ $(s \in L_f)$ and $t = ad$ $(t \in (\overline{s} \cap L_{\sim f}))$. If $j = 1$, then for any $n \in \mathbb{N}$, there is $ua \in (P_1^{-1}(P_1(t)) \cap L_{\sim f})$ and $v = cc^n$ such that $v \in L/ua$ and $\|v\| \geq n$, but $f \notin v$. But if $j = 2$, $(P_2^{-1}(P_2(t)) \cap L_{\sim f}) = \{ad\}$. Then for any $n \in \mathbb{N}$, there is $v = fc^n$ such that $v \in L/ad$, $\|v\| \geq n$ and $f \in v$.

In this case, the occurrence of event f contained in s can not be predicted by the first location, but can be predicted by the second location. So $s = adf$ is copredictable w.r.t. Σ_f and $\{P_i\}_{i\in\{1,2\}}$.

Case 3: Take $s = baf$ $(s \in L_f)$ and $t = ba$ $(t \in (\overline{s} \cap L_{\sim f}))$. If $j = 1$, $(P_1^{-1}(P_1(t)) \cap L_{\sim f}) = \{ba\}$. Then for any $n \in \mathbb{N}$, there is $v = fc^n$ such that $v \in L/ba$, $\|v\| \geq n$ and $f \in v$. But if $j = 2$, then for any $n \in \mathbb{N}$, there is $ua \in (P_2^{-1}(P_2(t)) \cap L_{\sim f})$ and $v = cc^n$ such that $v \in L/ua$ and $\|v\| \geq n$, but $f \notin v$.

In this case, the occurrence of event f contained in s can be predicted by the first location, but can not be predicted by the second location. So $s = baf$ is copredictable w.r.t. Σ_f and $\{P_i\}_{i\in\{1,2\}}$.

By Definition 3, it is known that L is relative copredictable w.r.t. Σ_f and $\{P_i\}_{i\in\{1,2\}}$. But according to Definition 1, L is identified as non-copredictable.

3.2. Definition of predictive vector

Definition 4 (failure states) Let G be a deterministic automaton and $\Sigma_f = \{\sigma_f\}$ be the failure event set. State $q \in Q$ is referred to as a failure state if $\sigma_f \in \Gamma(q)$; And the set of failure states is denoted by Θ.

Definition 5: Path $(q_0, \sigma_0, q_1, ..., \sigma_{n\text{-}1}, q_n)$ of G is referred to as a failure branch if $q_n \in \Theta$; And the set of failure branches is denoted by B_f.

Definition 6: Let $b \in B_f$ be a failure branch of G and $L(b)$ be the language generated by b. Branch b is copredictable w.r.t. Σ_f and $\{P_i\}_{i\in I}$, if $\exists s \in L(b)$ and $s\sigma_f \in L_f$ is copredictable w.r.t. Σ_f and $\{P_i\}_{i\in I}$.

Definition 7: Let $B_f = \{b_1, b_2, ..., b_n\}$ be the set of failure branches of G. Predictive vector ζ of G is defined as:

$$\zeta = (\rho_1, \rho_2, ..., \rho_n). \tag{3}$$

If b_j is copredictable w.r.t. Σ_f and $\{P_i\}_{i\in I}$, where $j = 1,2,...,n$, then ρ_j=1; otherwise, ρ_j=0.

Remark 4: If $\zeta = (1,1,...,1)$, L generated by G is copredictable w.r.t. Σ_f and $\{P_i\}_{i\in I}$; If $\zeta = (0_1,0,...,0)$, the language L is not relatively copredictable w.r.t. Σ_f and $\{P_i\}_{i\in I}$; otherwise, the language L is relatively copredictable w.r.t. Σ_f and $\{P_i\}_{i\in I}$.

The predictive vector describes the copredictability of a decentralized DES in detail. The more 1 in the vector, the more copredictable failure branches exist in the system. That is to say, the system has stronger copredictable performance.

3.3. Definition of r-step relative copredictability

Definition 8: Let L be the language generated by G with Σ_f and trace $s \in L_f$ be copredictable w.r.t. Σ_f and $\{P_i\}_{i\in I}$. Assume the occurrences of failure events contained in s are predictable by projections $P_1, P_2, ..., P_k (k \le m)$ and $t^j(s)$ is the shortest predictive prefix of s w.r.t. P_j, where $j = 1,2,...,k$. (a) If take $r^j(s) = (\min(\|s\|) - \|t^j(s)\| - 1)$, then s is predictable prior to $r^j(s)$ steps at most w.r.t. Σ_f and P_i. (b) If take $r(s) = \min\{r^1(s), r^2(s), ..., r^k(s)\}$, then s is said to be copredictable prior to $r(s)$ steps at most w.r.t. Σ_f and $\{P_i\}_{i\in I}$.

Definition 9: Let $b \in B_f$ be a failure branch of , $L(b)$ G be the language generated by b. Suppose $s \in L(b)$, $s\sigma_f \in L_f$ and $s\sigma_f$ is copredictable prior to $r(s)$ steps at most w.r.t. Σ_f and $\{P_i\}_{i\in I}$. Take $r(b) = r(s)$, then b is copredictable prior to $r(b)$ steps at most w.r.t. Σ_f and $\{P_i\}_{i\in I}$. Suppose $s \in L(b)$, $s\sigma_f \in L_f$ and $s\sigma_f$ is not copredictable w.r.t. Σ_f and $\{P_i\}_{i\in I}$, then take $r(b) = r(s) = \infty$ (∞ denotes infinity).

Definition 10: Let L generated by G be copredictable with Σ_f and $\{P_i\}_{i\in I}$. Suppose $B_f = \{b_1, b_2, ..., b_n\}$. (a) Take $\kappa = (r(b_1), r(b_2), ..., r(b_n))$, κ is referred to as step vector of B_f, where if b_i $(i = 1,2,...,n)$ is not copredictable w.r.t. Σ_f and $\{P_i\}_{i\in I}$, then $r(b_i) = \infty$. (b) If take $r = \min\{r(b_1), r(b_2) ..., r(b_i)\}$, then L is said to be copredictable prior to r steps at most w.r.t. Σ_f and $\{P_i\}_{i\in I}$.

4. AN ALGORITHM FOR VERIFYING RELATIVE COPREDICTABILITY

In the section, an algorithm based on the verifier (Jiang et al., 2001; Yoo and Lafortune, 2002) is presented to verify relative copredictability of DESs and to compute the boundary number of steps prior to occurrences of failure events. Furthermore, some examples are provided to illustrate the algorithm.

4.1. A verification algorithms based on a verifier

Algorithm 1:

Let L be the language generated by G with Σ_f. Assume there are m local projections $P: \Sigma^* \to \Sigma^*_{o,i}$, where $i \in I$.

Input:

$G = (Q, \Sigma, \delta, q_0)$, $B_f = \phi$, $k = 0$, $n = 0$.

Step 1: Mark all failure states of G.

Find all failure states of G and mark them with double solid circles.

Step 2: Construct non-failure automaton G_N.

$G_N = G \times A_N$ models the normal behavior of G, where A_N is composed of a single state N with a self-loop labeled with all events in $\Sigma \backslash \Sigma_f$.

Step 3: Put all failure branches of G_N in the set B_f and construct automata for them.

Find all failure branches of G_N by Definition 5 and denote them by $b_1, b_2, ..., b_n$. $B_f = \{b_1, b_2, ..., b_n\}$, where $n = |B_f|$ is the number of failure branches. Let b_i be a branch of B_f and $L(b_i)$ be the language generated by b_i, where $i = 1,2,...,n$. Construct the automata $G_{1i} = (Q_{1i}, \Sigma_{1i}, \delta_{1i}, q_0)$ for $L(b_i)$, where $Q_{1i} = \{q \in Q : q$ is the state in the branch $b_i\}$, $\Sigma_{1i} = \{\sigma \in \Sigma : \sigma \in L(b_i)\}$ and $\delta_{1i} = \delta|_{Q_{1i} \times \Sigma_{1i} \to Q_{1i}}$.

Step 4: Initialize predictive vector and step vector.

Predictive vector $\zeta = (\rho_1, \rho_2, . . . , \rho_n)$, step vector $\kappa = (r(b_1), r(b_2), ..., r(b_n))$, where $\rho_i = 0$, $r(b_i) = \infty$ and $i = 1,2,...,n$.

Step 5: Construct an automaton for $L^*_{\sim f}$.

Construct automaton $G_2 = (Q_2, \Sigma_2, \delta_2, q_0)$ for $L^*_{\sim f}$, where $\Sigma_2 = \{\sigma \in \Sigma : (\exists s \in L^*_{\sim f}) \quad \sigma \in s\}$, $Q_2 = \{q \in Q : q = q_0 \vee (\exists s \in L^*_{\sim f})\ \delta(q_0, s) = q\}$ and $\delta_2 = \delta|_{Q_2 \times \Sigma_2 \to Q_2}$.

Step 6: For each $j \in I$ and $i = 1,2,...,n$, construct verifier automaton V^j_{3i} according to G_2, G_{1i} and P_j.

Based on local projection P_j, automata G_2 and G_{1i}, construct verifier automaton V^j_{3i}. The verifier automaton is a finite-state automaton

$$V^j_{3i} = (Q^j_{Vi}, \Sigma^j_{Vi}, \delta^j_{Vi}, q_{v,0}) \tag{4}$$

where $Q^j_{Vi} \subseteq Q_2 \times Q_{1i}$ is the set of states; $\Sigma^j_{Vi} = \Sigma_2 \cup \Sigma_{1i}$; $q_{v,0} = (q_0, q_0)$; δ^j_{Vi} is the state transition function defined as follows:

for any state $(x_1, x_2) \in Q^j_{Vi}$, if $\sigma \in \Sigma_o$ for the local projection P_j, $\delta_2(x_1, \sigma) \in Q_2$ and $\delta_{1i}(x_2, \sigma) \in Q_{1i}$, then

$$\delta^j_{Vi}((x_1, x_2), \sigma) = (\delta_2(x_1, \sigma), \delta_{1i}(x_2, \sigma));$$

if $\sigma \in \Sigma_{uo}$ for the local projection P_j, then

(a) if $\delta_2(x_1, \sigma) \in Q_2$ and $\delta_{1i}(x_2, \sigma)$ is not defined in the G_{1i}, then

$$\delta_{Vi}^j((x_1, x_2), \sigma) = (\delta_2(x_1, \sigma), x_2);$$

(b) if $\delta_2(x_1, \sigma)$ is not defined in the G_2 and $\delta_{1i}(x_2, \sigma) \in Q_{1i}$, then

$$\delta_{Vi}^j((x_1, x_2), \sigma) = (x_1, \delta_{1i}(x_2, \sigma));$$

(c) if $\delta_2(x_1, \sigma) \in Q_2$ and $\delta_{1i}(x_2, \sigma) \in Q_{1i}$, then

$$\delta_{Vi}^j((x_1, x_2), \sigma) = \{(\delta_2(x_1, \sigma), x_2), (x_1, \delta_{1i}(x_2, \sigma)), (\delta_2(x_1, \sigma), \delta_{1i}(x_2, \sigma))\}.$$

Step 7: Check if there exists a state in the V_{3i}^j satisfying the condition (5) given by Theorem 1 below.

If the answer is yes for each $j \in I$, then ρ_i=0; otherwise ρ_i=1 and $k = k+1$.

Theorem 1: Let V_{3i}^j be the verifier of branch b_i under the location projection P_j. The branch b_i is not copredictable w.r.t. Σ_f and P_j iff there exists a state $(x_1, x_2) \in Q_{Vi}^j$ such that

$$[x_1 \notin \Theta_i] \wedge [x_2 \in \Theta_i], \quad (5)$$

where Θ_i denotes the set of failure states in the branch b_i.

Proof: (Sufficiency) Suppose there exists a state $(x_1, x_2) \in Q_{Vi}^j$ such that $x_1 \notin \Theta_i$ and $x_2 \in \Theta_i$. By the construction of V_{3i}^j, there exists a string $s \in L_{\sim f}^{*}$ and $t \in L_{\sim f}^{'}$ such that $P(s) = P(t)$, $x_1 = \delta(x_0, s)$ and $x_2 = \delta(x_0, t)$. Since $P(s) = P(t)$, b_i is not copredictable w.r.t. Σ_f and P_j by Definition 2 and Definition 6.

(Necessity) Suppose that b_i is not copredictable w.r.t. Σ_f and P_j. Then it is known that there exists $s \in L_{\sim f}^{'}$ such that $\delta(x_0, s) \in \Theta_i$. And there exists $t \in L_{\sim f}^{*}$ and $P(s) = P(t)$ such that $\delta(x_0, t) \notin \Theta_i$. According to the definition of V_{3i}^j, there exists a state $(x_1, x_2) \in Q_{Vi}^j$ such that $\delta(x_0, s) = x_1$, $\delta(x_0, t) = x_2$ and $[x_1 \in \Theta_i] \wedge [x_2 \notin \Theta_i]$. Thus, the condition(5) holds.

If there is a $j \in I$ such that the condition (5) does not hold, then the branch b_i is copredictable w.r.t. Σ_f and P_j. Let $s \in L(b_i)$ and $s\sigma_f \in L_f$, compute $r^j(s) = (\min(\|s\|) - \|t^j(s)\| - 1)$. And $r^j(b_i) = r^j(s)$. Suppose the occurrences of failure events contained in b_i are predictable by projections P_1, P_2, ..., P_k $(k \le m)$, then $r(b_i) = \min\{r^1(b_i), r^2(b_i) ..., r^k(b_i)\}$.

Step 8: Compute the value of predictive vector ζ, step vector κ and r.

$\zeta = (\rho_1, \rho_2, ..., \rho_n)$ and $\kappa = (r(b_1), r(b_2), ..., r(b_n))$. $r = \min\{r(b_i)\}$, where $i = 1, 2, ..., n$.

Output: ζ and r.

If ζ=(1,1,...,1), then L is copredictable prior to r steps at most w.r.t. $\{P_i\}_{i \in I}$ and Σ_f;

if ζ=(0,0,...,0), then L is not relatively copredictable w.r.t. $\{P_i\}_{i \in I}$ and Σ_f;

otherwise, L is relatively copredictable prior to r steps at most w.r.t. $\{P_i\}_{i \in I}$ and Σ_f.

Now, the computational complexity of the algorithm is discussed.

Given the system $G = (Q, \Sigma, \delta, q_0)$, the number of feasible transitions from a reachable state $q \in Q$ is $|\Sigma|$ in the worst case. So the time complexity of the first step is $O(|Q||\Sigma|)$. The number of states of G is $|Q|$, and the number of feasible transitions of every state is $(|\Sigma| - |\Sigma_f|)$, so the construction of G_N takes $O(|Q|(|\Sigma| - |\Sigma_f|))$ time. In the third step, because automaton G_{1i} has $|Q_{1i}|$ states and every state has $|\Sigma_{1i}|$ feasible transitions at most, so the the time complexity of construction of G_{1i} is $O(|Q_{1i}||\Sigma_{1i}|)$. Likewise, the time complexity of the construction of G_2 in the fifth step is $O(|Q_2||\Sigma_2|)$. The computational complexity of the sixth step and the seventh step is based on a verifier. The construction of the verifier takes time $O(n|Q^2||\Sigma|)$ in the worst case. Overall, the complexity of the algorithm is polynomial.

4.2. Illustrative examples

Example 2: Consider again plant G described in Example 1. Example 1 shows that L is relative copredictable w.r.t. Σ_f and $\{P_1, P_2\}$ by using the Definition 3. In the following, the conclusion will be verified by Algorithm 1.

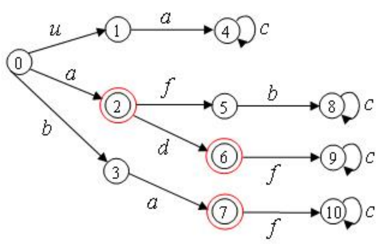


Fig. 2. Plant G of Example 2.

According to Definition 4, it is known that q_2, q_6 and q_7 are failure states of G, which are marked with double solid circle and highlighted by red color as that in Fig.2.

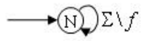


Fig. 3. A_N

A_N is described by Fig.3. Non-failure automaton G_N is constructed based on A_N as shown in Fig.4 .

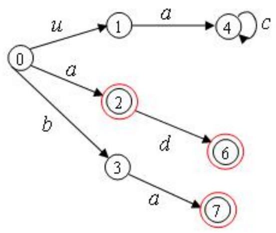
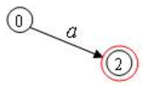


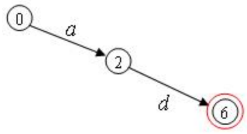
Fig. 4. G_N of Example 2.

As Fig.4 shows, G_N has three failure branches: $b_1=(0,a,2)$, $b_2=(0,a,2,d,6)$ and $b_3=(0,b,3,a,7)$. Put them in the set B_f. $B_f=\{b_1,b_2,b_3\}$ and $n=|B_f|=3$. Let Θ_i denote the set of the failure states in the branch b_i, then $\Theta_1=\{2\}$, $\Theta_2=\{6\}$ and $\Theta_3=\{7\}$.

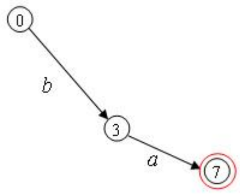
Automata G_{11}, G_{12} and G_{13} are constructed for $L(b_1)$, $L(b_2)$ and $L(b_3)$ as shown in Fig.5(a), Fig.5(b) and Fig.5(c) respectively.



(a) G_{11}



(b) G_{12}



(c) G_{13}

Fig. 5. $G_{1i}(i=1,2,3)$ of Example 2.

Since $n=3$, $\zeta=(\rho_1,\rho_2,\rho_3)$, $\kappa=(r(b_1),r(b_2),r(b_3))$, where ρ_i=0, $r(b_i)=\infty$ and $i=1,2,3$.

Now construct automaton G_2 as that in Fig.6.

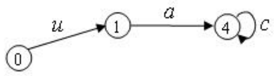


Fig. 6. G_2 of Example 2.

Based on local projections P_1,P_2, automata G_{11} and G_2, construct verifier automata V_{31}^1 and V_{31}^2 as described in Fig.7.

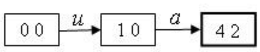
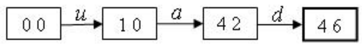
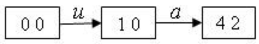


Fig. 7. V_{31}^1 or V_{31}^2

With the similar method, automata V_{32}^1 and V_{32}^2 are constructed as that in Fig.8(a) and Fig.8(b); automata V_{33}^1 and V_{33}^2 are constructed as that in Fig.9(a) and Fig.9(b).

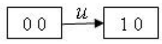


(a) V_{32}^1

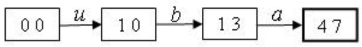


(b) V_{32}^2

Fig. 8. $V_{32}^i(i=1,2)$ of Example 2.



(a) V_{33}^1



(b) V_{33}^2

Fig. 9. $V_{33}^i(i=1,2)$ of Example 2.

As Fig.7 shows, state (4,2) in the V_{31}^1 or V_{31}^2 satisfies condition (5), i.e., $4\notin\Theta_1$, but $2\in\Theta_1$. Therefore, b_1 is not copredictable w.r.t. Σ_f and $\{P_i\}_{i\in\{1,2\}}$, i.e. ρ_1=0.

From Fig.8(a), it is clear that state (4,6) in the V_{32}^1 satisfies condition (5), i.e., $4\notin\Theta_2$, but $6\in\Theta_2$. So b_2 is not predictable w.r.t. Σ_f and P_1. But in Fig.8(b), there is no state of the V_{32}^2 satisfies condition (5), so b_2 is predictable w.r.t. Σ_f and P_2. According to the definition 2 and 6, b_2 is copredictable w.r.t. Σ_f and $\{P_i\}_{i\in\{1,2\}}$, i.e., ρ_2=1. Let $s=ad$, then $s\in \mathrm{L}(b_2)$ and $s\sigma_f\in L_f$. As seen from V_{32}^2, a is not the shortest predictive prefix of the trace $s\sigma_f$ since there is an indistinguishable string ua. Its the shortest predictive prefix is ad, i.e., $t^2(s\sigma_f)=ad$. Therefore, $r^2(s\sigma_f)=(\min(\|s\sigma_f\|)-\|t^2(s\sigma_f)\|-1)=0$ i.e., $r(b_2)=0$.

By a simple inspection Fig.9(a), there is no state in the V_{33}^1 satisfies condition (5), so b_3 is predictable w.r.t. Σ_f and P_1. As Fig.9(b) shows, state (4,7) in the V_{33}^2 satisfies condition (5), i.e., $4\notin\Theta_3$, but $7\in\Theta_3$. So b_3 is not predictable w.r.t. Σ_f and P_2. According to the definition 2 and 6, b_3 is copredictable w.r.t. Σ_f and $\{P_i\}_{i\in\{1,2\}}$, i.e., ρ_3=1. Let $s=ba$, then $s\in L(b_3)$ and $s\sigma_f\in L_f$. As seen from V_{33}^1, b is the shortest predictive prefix of the trace $s\sigma_f$ since there is no indistinguishable string with the string b, i.e., $t^1(s\sigma_f)=b$. Therefore, $r^1(s\sigma_f)=(\min(\|s\sigma_f\|)-\|t^1(s\sigma_f)\|-1)=1$ i.e., $r(b_3)=1$.

According to the above analysis, it is known that $\zeta=(0,1,1)$, $\kappa=(\infty,0,1)$ and $r=\min\{r(b_1),r(b_2),r(b_3)\}=0$.

The result shows that w.r.t. Σ_f and $\{P_i\}_{i\in\{1,2\}}$, failure branch b_1 is not copredictable, failure branch b_2 is copredictable prior to 0 steps at most, while failure branch b_3 is copredictable prior to 1 steps at most. That is to say, two of three failure branches of G are copredictable. Therefore, the same conclusion as Example 1 that L is relatively copredictable w.r.t. Σ_f and $\{P_i\}_{i\in\{1,2\}}$ is drawn. Moreover, we know that L is relatively copredictable prior to 0 steps at most w.r.t. Σ_f and $\{P_i\}_{i\in\{1,2\}}$.

Example 3: Consider the plant G shown in Fig.10, where $\Sigma=\{a,b,c,d,f\}$ with a failure event set $\Sigma_f=\{f\}$ and q_0 is the initial state. Assume that there are two local projections $P_i:\Sigma^*\to\Sigma_{o,i}^*$, where $i=1,2$, $\Sigma_{o,1}=\{a,b,c,d\}$, and $\Sigma_{o,2}=\{a,b,c\}$.

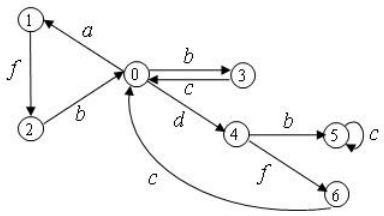


Fig. 10. Plant G in Example 3.

In the following steps, Algorithm 1 will be used to test the copredictability of G .

From Fig.10, it is known that q_1 and q_4 are failure states. Mark them with double circle as that in Fig.11.

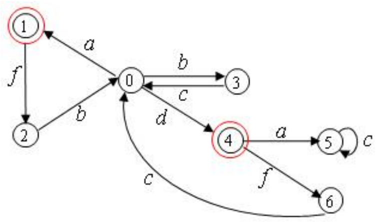


Fig. 11. Plant G of Example 3 with failure states.

G_N is obtained by $G\times A_N$ as shown in Fig.12.

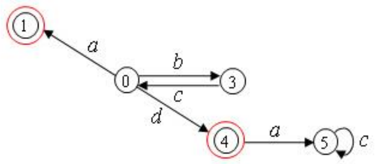
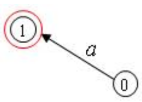
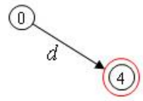


Fig. 12. G_N of Example 3.

As Fig.12 shows, G_N has two failure branches: $b_1=(0,a,1)$, $b_2=(0,d,4)$. Therefore, $B_f=\{b_1,b_2\}$, $n=|B_f|=2$. Construct automata G_{11} and G_{12} as that in Fig.13(a) and Fig.13 (b), respectively.



(a) G_{11}



(b) G_{12}

Fig. 13. G_{1i}(i=1,2) of Example 3.

Because $n=2$, so $\zeta=(\rho_1,\rho_2)$, $\kappa=(r(b_1),r(b_2))$, where $\rho_i=0$, $r(b_i)=\infty$ and $i=1,2$.

Continue to construct automaton G_2 as that in Fig.14.

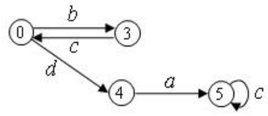
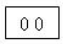
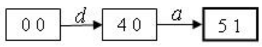


Fig. 14. G_2 of Example 3.

Then, construct automata G_{31}^1 and G_{31}^2 as that in Fig.15(a) and Fig.15 (b), respectively; construct automata G_{32}^1 and G_{32}^2 as that in Fig.16 (a) and Fig.16 (b), respectively.

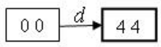


(a) V_{31}^1

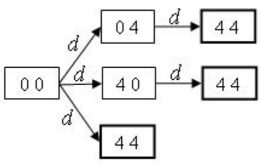


(b) V_{31}^2

Fig. 15. V_{31}^i(i=1,2) of Example 3.



(a) V_{32}^1



(b) V_{32}^2

Fig. 16. V_{32}^i(i=1,2) of Example 3.

As Fig.15 shows, state (5,1) in the V_{31}^2 satisfies condition (5). But in the V_{31}^1, there is no state satisfies condition (5). Therefore, b_1 is copredictable w.r.t. Σ_f and $\{P_i\}_{i\in\{1,2\}}$, i.e. $\rho_1=1$. Let $s=a$, then $s\in L(b_1)$ and $s\sigma_f\in L_f$. As seen from V_{31}^1, a is the shortest predictive prefix of the trace $s\sigma_f$ since there is no indistinguishable string with the string a , i.e. , $t^1(s\sigma_f)=a$. Therefore, $r^1(s\sigma_f)=(\min(\|s\sigma_f\|)-\|t^1(s\sigma_f)\|-1)=0$, i.e., $r(b_1)=0$.

Note the state (4,4) in the V_{32}^1 or V_{32}^2 of Fig.16, in which the first state 4 is the state 4 in G_2, while the second state 4 refers to the failure state 4 in the b_2. So the state (4,4) satisfies condition (5). Therefor, b_2 is not copredictable w.r.t. Σ_f and $\{P_i\}_{i\in\{1,2\}}$, i.e. $\rho_2 = 0$.

Through the algorithm, it is known that $\zeta = (1,0)$, $\kappa=(0,\infty)$ and $r = \min\{r(b_1), r(b_2)\}=0$.

The result indicates that one of two failure branches of G is copredictable. So L generated by the plant G is relatively copredictable prior to 0 steps at most w.r.t. Σ_f and $\{P_i\}_{i\in\{1,2\}}$.

Example 4: Consider the plant G (Liu, 2019) shown in Fig.17, where $\Sigma = \{a,b,e,g,\sigma_f,\sigma_u\}$ with a failure event set $\Sigma_f = \{\sigma_f\}$ and q_0 is the initial state. Assume that there are two local projections $P_i : \Sigma^* \to \Sigma_{o,i}^*$, where $i = 1,2$, $\Sigma_{o1} = \{g,a,b\}$, and $\Sigma_{o2} = \{e,a,b\}$. Liu has shown that L generated by G is copredictable w.r.t. Σ_f and $\{P_i\}_{i\in\{1,2\}}$. Now the conclusion is verified by Algorithm 1.

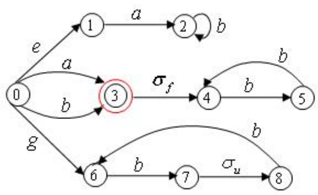


Fig. 17. Plant G in Example 4.

As Fig.17 shows, q_3 is the failure state. And G_N is constructed as that in Fig.18.

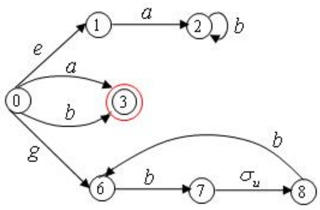
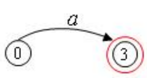


Fig. 18. G_N of Example 4.

There are two failure branch in G_N: $b_1 = (0,a,3)$, $b_2 = (0,b,3)$. So $B_f = \{b_1, b_2\}$. Fig.19 (a) and Fig.19 (b) are the automata corresponding to them. Since $n = |B_f| = 2$, $\zeta = (\rho_1, \rho_2) = (0,0)$ and $\kappa=(r(b_1), r(b_2)) = (\infty,\infty)$.



(a) G_{11}

(b) G_{12}

Fig. 19. Failure branches of G in Example 4.

Automaton G_2 is constructed as that in Fig.20.

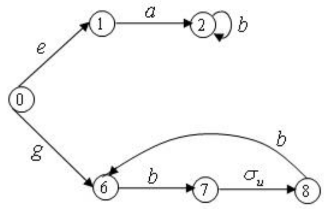
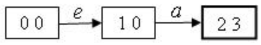
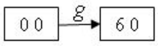


Fig. 20. G_2 of Example 4.

Automata V_{31}^1 and V_{31}^2 are constructed as that in Fig.21(a) and Fig.21 (b), respectively; automata V_{32}^1 and V_{32}^2 are constructed as that in Fig.22 (a) and Fig.22 (b), respectively.

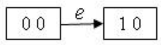


(a) V_{31}^1

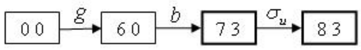


(b) V_{31}^2

Fig. 21. $V_{31}^i (i = 1,2)$ of Example 4.



(a) V_{32}^1



(b) V_{32}^2

Fig. 22. $V_{32}^i (i = 1,2)$ of Example 4.

As Fig.21(b) and Fig.22(a) show, there is no state satisfies condition (5) in the V_{31}^2 and V_{32}^1. Therefore, b_1 and b_2 are copredictable w.r.t. Σ_f and $\{P_i\}_{i\in\{1,2\}}$, i.e. ρ_1=1 and ρ_2=1. Let $s = a$, then $s \in L(b_1)$ and $s\sigma_f \in L_f$. As seen from V_{31}^2, a is the shortest predictive prefix of the trace $s\sigma_f$ since there is no indistinguishable string with the string a, i.e., $t^2(s\sigma_f) = a$. Therefore, $r^1(s\sigma_f) = (\min(\|s\sigma_f\|) - \|t^2(s\sigma_f)\| - 1) = 0$,i.e., $r(b_1) = 0$. Let $s = b$, then $s \in L(b_2)$ and $s\sigma_f \in L_f$. As seen from V_{32}^1, b is the shortest predictive prefix of the trace $s\sigma_f$ since there is no indistinguishable string with the string b, i.e., $t^1(s\sigma_f) = b$. Therefore, $r^2(s\sigma_f) = (\min(\|s\sigma_f\|) - \|t^1(s\sigma_f)\| - 1) = 0$, i.e., $r(b_2) =$

0.

Based on the above analysis, it is easy to know that $\zeta = (\rho_1, \rho_2)$=(1,1), κ=(r(b$_1$),r(b$_2$))=(0,0) and r=min{r(b$_1$),r(b$_2$)}=0.

The result indicates that two failure branches of the plant G are both copredictable prior to 0 steps at most w.r.t. Σ_f and $\{P_i\}_{i\in\{1,2\}}$. So L generated by the plant G is copredictable prior to 0 step at most w.r.t. Σ_f and $\{P_i\}_{i\in\{1,2\}}$. The example shows that Algorithm 1 is also applicable for the verification of copredictability introduced by Liu.

7. CONCLUSIONS

In this paper, the relative copredictability of decentralized DESs is investigated. The notion of relative copredictability was formalized, which is weaker than copredictability and relative predictability. In order to verify whether a system is relatively copredictable, a necessary and sufficient condition of relative copredictability was presented. Moreover, an algorithm of polynomial complexity for calculating the predictive vector and the boundary number of steps prior to occurrences of failures was proposed. This reported work generalized the main results of (Liu, 2019). Further issue worthy of consideration is the problem of relative predictability of fuzzy DESs. We would like to consider it in subsequent work.

ACKNOWLEDGEMENTS

This work was supported by “the National Natural Science Foundation of China” (grand number 61673122) and “the Natural Science Foundation of Guangdong Province” (grand number 2019A1515010548) of China.

REFERENCES

Chang, M., Dong, W., Ji, Y. and Tong, L. (2013). On fault predictability in stochastic discrete event systems. *Asian Journal of Control*, vol. 15(5), 1458-1467.

Chen, J. and Kumar, R. (2014). Failure prognosability of stochastic discrete event systems. *in Proc. 2014 Amer. Control Conf, June*, 2041-2046.

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

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

Jiang, S., Huang, Z., Chandra, V. and Kumar, R. (2001). A polynomial time algorithm for diagnosability of discrete event systems. *IEEE Trans. on Automatic Control*, vol. 46(8), 1318-1321.

Keroglou, C. and Hadjicostis, C. N. (2018). Distributed fault diagnosis in discrete event systems via set intersection refinements. *IEEE Transactions on Automatic Control*, vol. 63(10), 3601-3607.

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

Liu, F. (2019). Predictability of failure event occurrences in decentralized discrete-event systems and polynomialtime verification. *IEEE Trans. on Automation science and engineering* , vol. 16(1), 498-504.

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

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

Takai, S. (2012). Robust failure prognosis of partially observed discrete event systems. *in Proc. 2012 Amer. Control Conf., Montreal, Canada, June*,27-29.

Takai, S. and Kumar, R. (2011). Inference-based decentralized prognosis in discrete event systems. *IEEE Trans. On Automatic Control*, vol. 56(1), 165-171.

Takai, S. and Kumar, R. (2012). Distributed failure prognosis of discrete event systems with bounded-delay communications. *IEEE Trans. on Automatic Control*, vol. 57(5), 1259-1265.

Viana, G. S. and Basilio, J. C. (2019). Codiagnosability of discrete event systems revisited: A new necessary and sufficient condition and its applications. *Automatica*, vol. 101, 354-364.

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*, vol. 14(2), 435-442.

Yin, X. and Li, Z. (2016). Reliable decentralized fault prognosis of discrete-event systems. *IEEE Trans. on Systems, Man, and Cybernetics: Systems*, vol. 46(11), 1598-1603.

Yin, X. and Li, Z. (2019). Decentralized fault prognosis of discrete-event systems using state-estimate-based protocols. *IEEE Transactions on Cybernetics*, vol. 49(4), 1302-1313.

Yokotani, M., Kondo, T. and Takai, S. (2016). Abstraction-based verification and synthesis for prognosis of discrete event systems. *Asian Journal of Control*, vol. 18(4), 1279-1288.

Yoo, T. and Lafortune, S. (2002). Polynomial-time verification of diagnosability of partially-observed discrete-event systems. *IEEE Trans. on Automatic Control*, vol. 47(9), 1491-1495.

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*, vol. 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*, vol. 92(7), 1600-1608.