Automatica 69 (2016) 375–379

Contents lists available at ScienceDirect

Automatica

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

Technical communique

Decentralized fault prognosis of discrete event systems with guaranteed performance bound☆

Xiang Yin [a,1], Zhaojian Li [b]

[a] Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA
[b] Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA

ARTICLE INFO

Article history:
Received 25 August 2015
Received in revised form
22 December 2015
Accepted 23 February 2016

Keywords:
Discrete-event systems
Fault prognosis
Decentralized architectures

ABSTRACT

We study the problem of decentralized fault prognosis of partially-observed discrete event systems. In order to capture the prognostic performance issue in the prognosis problem, we propose two new criteria: (1) all faults can be predicted K steps ahead; and (2) a fault will occur for sure within M steps once a fault alarm is issued; and we refer to (M, K) as the performance bound of the prognostic system. A necessary and sufficient condition for the existence of a decentralized supervisor satisfying these two criteria is provided, which is termed as (M, K)-coprognosability. A polynomial-time algorithm for the verification of (M, K)-coprognosability is also proposed. Finally, we show that the proposed approach is applicable to both disjunctive and conjunctive architectures. Our results generalize previous work on decentralized fault prognosis.

1. Introduction

Fault prognosis is an important issue for safety-critical systems. Recently, the problem of fault prognosis has received considerable attention in the Discrete-Event System (DES) literature; see, e.g., Cassez and Grastien (2013), Chang, Dong, Ji, and Tong (2013), Chen and Kumar (2015), Genc and Lafortune (2009), Jéron, Marchand, Genc, and Lafortune (2008), Khoumsi and Chakib (2009), Khoumsi and Chakib (2012), Kumar and Takai (2010), Lefebvre (2014a), Lefebvre (2014b), Nouioua, Dague, and Ye (2014), Takai and Kumar (2011), Takai (2015), Ye, Dague, and Nouioua (2013), and Yin and Lafortune (2015). In Jéron et al. (2008) and Genc and Lafortune (2009), the fault prognosis (or prediction) problem was first studied for the centralized partially-observed DES, where the notion of predictability was introduced. In Kumar and Takai (2010), the authors studied the decentralized fault prognosis problem under the disjunctive architecture, where the notion of coprognosability was proposed. Particularly, a system is coprognosable if and only if there exists a decentralized prognoser that can predict fault correctly. The decentralized prognosis problem has been further studied recently under the conjunctive architecture (Khoumsi & Chakib, 2012) and the inference-based architecture (Takai & Kumar, 2011). Roughly speaking, in the disjunctive architecture, a global fault alarm is issued iff *one* local agent issues a fault alarm. While in the conjunctive architecture, a global fault alarm is issued iff *all* local agents issue a fault alarm. In the inference architecture, multilevel inference for each local agent is used in order to issue a global fault alarm.

Most of the previous work on decentralized prognosis are based on two criteria: "no missed alarm" and "no false alarm", where the former requires that any fault can be predicted prior to its occurrence and the latter requires that a fault will happen for sure once an alarm is issued. However, these two criteria do not care how early or how late the fault alarm is issued. In practice, once a fault alarm is issued, some procedures will be taken in order to protect the system. Since the protection could be costly, one may not want to take it unless it is necessary. On the other hand, the protection may require certain amount of time to set up. Therefore, we also need to guarantee that the fault alarm can be issued in time before certain threshold.

In this note, we investigate the problem of decentralized fault prognosis of DES. Two new prognostic performance criteria are proposed in order to capture the "timing" issue. Specifically, we require that (1) any fault can be predicted K steps prior to its occurrence; and (2) if an alarm is issued, then a fault will occur for sure within M steps from the alarm. We refer to this integer pair (M, K) as the performance bound of the

☆ The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Carlo Fischione under the direction of Editor André L. Tits.

E-mail addresses: xiangyin@umich.edu (X. Yin), zhaojli@umich.edu (Z. Li).
[1] Tel.: +1 7348343243.

http://dx.doi.org/10.1016/j.automatica.2016.03.015
0005-1098/

prognostic system. The contributions of this note are as follows. First, we extend the previous work on the decentralized prognosis problem by proposing the notion of (M, K)-performance bound that takes the prognostic performance issue into account. Second, we provide the necessary and sufficient condition for the existence of a decentralized prognoser that achieves this performance bound. A polynomial-time algorithm for the verification of the existence condition is also provided. Third, we show that the proposed approach is applicable to both disjunctive and conjunctive architectures. We note that the conjunctive prognosis was initially studied by Khoumsi and Chakib (2012); however, to the best of knowledge, no verification algorithm for conjunctive coprognosability is provided so far. As a special case of our notion, now it can be effectively verified by the algorithm proposed in this note.

2. Preliminaries

Let Σ be a finite set of events and Σ^* be the set of all finite strings over Σ, including the empty string ϵ. A language $L \subseteq \Sigma^*$ is a set of strings. We denote by $\overline{L}$ the prefix-closure of L, i.e., $\overline{L} = \{s \in \Sigma^* : \exists t \in \Sigma^* \text{ s.t. } st \in L\}$. We denote by $|s|$ the length of a string $s \in \Sigma^*$ with $|\epsilon| = 0$. We denote by L/s the post-language of s, i.e., $L/s := \{t \in \Sigma^* : st \in L\}$. A language L is live if $\forall s \in L, \exists \sigma \in \Sigma : s\sigma \in L$. A DES is modeled by a deterministic finite-state automaton (DFA) $G = (Q, \Sigma, \delta, q_0, Q_m)$, where Q is the finite set of states, Σ is the finite set of events, $\delta : Q \times \Sigma \to Q$ is the partial transition function, $q_0 \in Q$ is the initial state and Q_m is the set of marked states. We write a DFA G as $G = (Q, \Sigma, \delta, q_0)$ if marking is not considered. The transition function δ is extended to $Q \times \Sigma^*$ in the usual manner (see, e.g., Cassandras & Lafortune, 2008). The language generated by G from state q is defined by $\mathcal{L}(G, q) = \{s \in \Sigma^* : \delta(q, s)!\}$, where ! means "is defined". The language marked by G from state q is $\mathcal{L}_m(G, q) = \{s \in \Sigma^* : \delta(q, s) \in Q_m\}$. We write $\mathcal{L}(G, q)$ and $\mathcal{L}_m(G, q)$ as $\mathcal{L}(G)$ and $\mathcal{L}_m(G)$, respectively, when $q = q_0$. Hereafter, we assume w.l.o.g. that $\mathcal{L}(G)$ is live.

In the fault prognosis problem, the goal is to predict whether or not the system will violate some normal behaviors in the future. To this end, we define $H = (Q_H, \Sigma, \delta_H, q_{0,H})$ as the specification automaton that captures the normal behaviors of the system, where $\mathcal{L}(H) \subseteq \mathcal{L}(G)$. We say that H is a sub-automaton of G, denoted by $H \sqsubseteq G$, if $\delta_H(q_{0,H}, s) = \delta(q_0, s)$ for all $s \in \mathcal{L}(H)$. We say that H is a strict sub-automaton of G, denoted by $H \sqsubset G$, if: (1) $H \sqsubseteq G$; and (2) $\forall s \in \mathcal{L}(G) \setminus \mathcal{L}(H) : \delta(q_0, s) \notin Q_H$. Hereafter, we assume w.l.o.g. that the specification automaton $H = (Q_H, \Sigma, \delta_H, q_{0,H})$ is a strict sub-automaton of the system automaton $G = (Q, \Sigma, \delta, q_0)$, i.e., $H \sqsubset G$. Under this assumption, string $s \in \mathcal{L}(G)$ is a non-fault string if and only $\delta(s) \in Q_H$.

In the decentralized fault prognosis (Kumar & Takai, 2010), the system is monitored by a set of agents (or local prognosers) that work as a team in order to predict the fault. We assume that there are n local agents and we denote by $\mathcal{I} = \{1, \ldots, n\}$ the index set. We denote by $\Sigma_{o,i}$ the set of locally observable events of agent $i \in \mathcal{I}$. Then $P_i : \Sigma^* \to \Sigma_{o,i}^*$ is the natural projection defined in the usual manner; see, e.g., (Cassandras & Lafortune, 2008). Each local prognoser $i \in \mathcal{I}$ is defined as the function $\mathcal{A}_i : P_i(\mathcal{L}(H)) \to \{0, 1\}$, where "1" means a fault alarm is issued and "0" means no fault alarm is issued. Each local prognoser sends its local prognostic decision to a coordinator in order to calculate a global prognostic decision. The decentralized prognoser is the function $\{\mathcal{A}_i\}_{i\in\mathcal{I}} : \mathcal{L}(H) \to \{0, 1\}$ defined by: for any string $s \in \mathcal{L}(H)$,

$$\{\mathcal{A}_i\}_{i\in\mathcal{I}}(s) = 1 \Leftrightarrow \exists i \in \mathcal{I} : \mathcal{A}_i(P_i(s)) = 1. \tag{1}$$

In Kumar and Takai (2010), two criteria, "no missed alarm" and "no false alarm", were proposed in order to evaluate a decentralized prognoser. In particular, it was shown that the notion of coprognosability provides the necessary and sufficient condition under which there exists a decentralized prognoser satisfying the above two conditions. We first recall its definition from Kumar and Takai (2010).

Definition 1 (*Coprognosability*)**.** A specification $\mathcal{L}(H)$ is said to be coprognosable w.r.t. $\mathcal{L}(G)$ and $\Sigma_{o,i}, i \in \mathcal{I}$ if $(\exists m \in \mathbb{N})(\forall s \in \mathcal{L}(G) \setminus \mathcal{L}(H))(\exists t \in \overline{\{s\}} \cap \mathcal{L}(H))(\exists i \in \mathcal{I})(\forall u \in P_i^{-1}P_i(t) \cap \mathcal{L}(H))(\forall v \in \mathcal{L}(G)/u)[|v| \geq m \Rightarrow uv \in \mathcal{L}(G) \setminus \mathcal{L}(H)]$.

Remark 1. Intuitively, coprognosability requires that for any fault string, it must have a non-fault prefix such that at least one agent knows for sure that the fault is inevitable in the future. Although the notion of coprognosability guarantees that the fault can be predicted correctly, it does not care how early or how late the fault alarm is issued, i.e., no prognostic performance is guaranteed. However, this issue is very important in many practical applications. For example, in an uninterruptible power system, one may need to predict potential failures and to take some protections before the failure occurs, e.g., starting a backup battery. On the one hand, one may not want that the fault alarm is issued too late, since the backup battery may require several steps to set up. On the other hand, one also does not want that the fault alarm is issued too early, since the backup battery can only support for a limited amount of steps. Therefore, new criteria are needed in order to address the above prognostic performance requirements.

3. Main results

In this section, we propose the notion of (M, K)-coprognosability that quantitatively generalizes the notion of coprognosability by taking the prognostic performance issue into account. First, we define the notion of *performance bound*.

Definition 2. Let $M, K \in \mathbb{N}$ be two non-negative integers. A decentralized prognoser $\{\mathcal{A}_i\}_{i\in\mathcal{I}}$ is said to be prognosable with performance bound (M, K) (or a (M, K)-prognoser) if the following two properties hold:

1. Any fault can be alarmed K steps before its occurrence, i.e.,

$$(\forall s \in \mathcal{L}(G) \setminus \mathcal{L}(H))(\exists t v \in \overline{\{s\}} \cap \mathcal{L}(H) : |v| \geq K) \times [\{\mathcal{A}_i\}_{i\in\mathcal{I}}(t) = 1]. \tag{2}$$

2. Fault is guaranteed to occur within M steps once a fault alarm is issued, i.e., for any string $s \in \mathcal{L}(H)$,

$$[\{\mathcal{A}_i\}_{i\in\mathcal{I}}(s) = 1] \Rightarrow (\forall t \in \mathcal{L}(G)/s)[|t| \geq M \Rightarrow st \in \mathcal{L}(G) \setminus \mathcal{L}(H)]. \tag{3}$$

Remark 2. The conditions in Eqs. (2) and (3) generalize the criteria of "no missed alarm" and "no false alarm", respectively, in a quantitative manner by requiring that *when* the fault alarm is issued. Note that these two performance criteria are defined in terms of event steps, i.e., we consider logical prognostic performance criteria.

Before we show the existence condition of a (M, K)-decentralized prognoser, let us first introduce some necessary notations. For each state $q \in Q_H$ in H, we denote by $d_{min}(q)$ the length of the shortest no-fault string from q from which a fault may occur, i.e., $d_{min}(q) = \min_{s\in\mathcal{L}(G,q)\setminus\mathcal{L}(H,q)} |s| - 1$. We assume w.l.o.g. that $d_{min}(q_0) \geq K$; otherwise, (M, K)-coprognosability is violated trivially. Also, we denote by $d_{max}(q)$ the length of the longest non-fault string from q, i.e., $d_{max}(q) = \max_{s\in\mathcal{L}(H,q)} |s|$. Clearly, $d_{max}(q) = \infty$ iff q can reach a cycle of H, i.e., there exists an arbitrarily long non-fault string defined at q. We denote by $\partial_K(H, G)$ the set of states in

H from which a fault behavior may happen in K steps at the soonest, i.e.,

$$\partial_K(H, G) = \{q \in Q_H : d_{min}(q) = K\}. \tag{4}$$

We denote by $\Im_M^<(H, G)$ the set of states in H from which a fault will occur for sure within M steps, i.e.,

$$\Im_M^<(H, G) = \{q \in Q_H : d_{max}(q) \leq M\}. \tag{5}$$

For each local prognoser $i \in \mathcal{I}$, we also denote by $\mathcal{E}_i^H(s)$ Agent i's state estimate of s under $\Sigma_{o,i}$ w.r.t. the state space of H, i.e.,

$$\mathcal{E}_i^H(s) := \{q \in Q_H : \exists t \in \mathcal{L}(H) \text{ s.t. } P_i(s) = P_i(t) \wedge \delta_H(t) = q\}.$$

Intuitively, $\mathcal{E}_i^H(s)$ represents Agent i's knowledge about the system state upon the occurrence of s.

With the notions introduced above, we are now ready to introduce the notion of (M, K)-coprognosability.

Definition 3. $\mathcal{L}(H)$ is said to be (M, K)-coprognosable w.r.t. $\mathcal{L}(G)$ and $\Sigma_{o,i}$, $i \in \mathcal{I}$ if for any string $s \in \mathcal{L}(H)$, we have

$$\delta_H(s) \in \partial_K(H, G) \Rightarrow (\exists i \in \mathcal{I})[\mathcal{E}_i^H(s) \subseteq \Im_M^<(H, G)]. \tag{6}$$

Intuitively, the above definition requires that for any string that ends up with a state in $\partial_K(H, G)$, there must exist at least one agent such that it knows for sure that the current state is in $\Im_M^<(H, G)$. Therefore, it can make a local prognostic decision which is needed. The following result reveals that (M, K)-coprognosability provides the necessary and sufficient condition for the existence of a (M, K)-decentralized prognoser.

Theorem 1. *There exists a (M, K)-decentralized prognoser $\{\mathcal{A}_i\}_{i\in\mathcal{I}}$, if and only if, $\mathcal{L}(H)$ is (M, K)-coprognosable w.r.t. $\mathcal{L}(G)$ and $\Sigma_{o,i}$, $i \in \mathcal{I}$.*

Proof. ($\Leftarrow$) By construction. Let us consider a decentralized prognoser $\{\mathcal{A}_i\}_{i\in\mathcal{I}}$ defined as follows. For each $i \in \mathcal{I}$, we define

$$\mathcal{A}_i(P_i(s)) = 1 \Leftrightarrow \mathcal{E}_i^H(s) \subseteq \Im_M^<(H, G). \tag{7}$$

Next, we show that the properties in Eqs. (2) and (3) are satisfied under the above prognostic strategy.

First, for any fault string $s \in \mathcal{L}(G) \setminus \mathcal{L}(H)$, there exists a prefix $t \in \overline{\{s\}} : t \in \partial_K(H, G)$. Since $\mathcal{L}(H)$ is (M, K)-coprognosable, we know that $(\exists i \in \mathcal{I})[\mathcal{E}_i^H(t) \subseteq \Im_M^<(H, G)]$. Therefore, $(\exists i \in \mathcal{I})[\mathcal{A}_i(P_i(t)) = 1]$, i.e., $\{\mathcal{A}_i\}_{i\in\mathcal{I}}(t) = 1$. Since $t \in \partial_K(H, G)$, we know that any K-step extension of t is still in $\mathcal{L}(H)$, which means that $\exists v : tv \in \overline{\{s\}} \cap \mathcal{L}(H) \wedge |v| \geq K$. Therefore, Eq. (2) holds. Second, we show Eq. (3) holds by contradiction. We assume that Eq. (3) does not hold, which implies that $(\exists s \in \mathcal{L}(H) : \{\mathcal{A}_i\}_{i\in\mathcal{I}}(s) = 1)(\exists t \in \mathcal{L}(G)/s)[|t| \geq M \wedge st \in \mathcal{L}(H)]$. However, by $\{\mathcal{A}_i\}_{i\in\mathcal{I}}(s) = 1$, we know that $(\exists i \in \mathcal{I})[\delta_H(s) \in \mathcal{E}_i^H(s) \subseteq \Im_M^<(H, G)]$, i.e., $d_{max}(\delta_H(s)) < M$. Therefore, we have $(\forall t \in \mathcal{L}(G)/s)[|t| \geq M \Rightarrow st \notin \mathcal{L}(H)]$, which is a contradiction. Hence, Eq. (3) also holds.

($\Rightarrow$) By contradiction. We assume that there exists a decentralized prognoser $\{\mathcal{A}_i\}_{i\in\mathcal{I}}$ but $\mathcal{L}(H)$ is not (M, K)-coprognosable, which means that $(\exists s \in \mathcal{L}(H) : \delta_H(s) \in \partial_K(H, G))(\forall i \in \mathcal{I})[\mathcal{E}_i^H(s) \not\subseteq \Im_M^<(H, G)]$. For the above strings s, we know that $\forall i \in \mathcal{I}, \exists s_i \in \mathcal{L}(H) : P_i(s) = P(s_i) \wedge \delta_H(s_i) \notin \Im_M^<(H, G)$. Since $P(s) = P(s_i)$, we know that $\forall t \in \overline{\{s\}}, \exists t_i \in \overline{\{s_i\}} : P_i(t) = P_i(t_i)$. Moreover, since $\delta_H(s_i) \notin \Im_M^<(H, G)$, we know that $\forall t_i \in \overline{\{s_i\}} : \delta_H(t_i) \notin \Im_M^<(H, G)$. By Eq. (2), we know that $(\exists i \in \mathcal{I})(\exists t \in \overline{\{s\}} \cap \mathcal{L}(H))[\mathcal{A}_i(t) = 1]$. For the above agent i, since $P_i(t) = P_i(t_i)$, we know that $\mathcal{A}_i(t_i) = 1$, which implies that $\{\mathcal{A}_i\}_{i\in\mathcal{I}}(t_i) = 1$. However, $\delta_H(t_i) \notin \Im_M^<(H, G)$, which means that a M-step extension of t_i need not be fault. This contradicts Eq. (3). Therefore, $\mathcal{L}(H)$ is (M, K)-coprognosable. □

Remark 3. Note that the "($\Leftarrow$)" direction of the above proof is constructive. In fact, it tells us how to *synthesize* the desired decentralized prognoser when it exists. For each agent $i \in \mathcal{I}$, it just needs to remember its state-estimate $\mathcal{E}_i^H(s)$ and update it when a new locally observable event occurs. The new state estimate can be computed in polynomial-time w.r.t. the size of H. In other words, once (M, K)-coprognosability is verified off-line (we will discuss its verification later), the synthesis can be done online in a distributed manner, i.e., each local agent does not need to know the other agents' strategy.

Hereafter, we present a polynomial-time algorithm for the verification of (M, K)-coprognosability. For the sake of simplicity, we assume that $n = 2$. It can be extended to an arbitrary number of agents in a trivial manner.

Recall that we assume that $H \sqsubset G$. Let $\Sigma_{o,1}$ and $\Sigma_{o,2}$ be the sets of locally observable events for Agents 1 and 2, respectively. We define $\Sigma^+ = \Sigma \cup \{\epsilon\}$. Then the (M, K)-verifier V is defined as the DFA $V = (X_V, \Sigma_V, f_V, x_{0,V}, X_{m,V})$, where $X_V = Q_H \times Q_H \times Q_H$ is the set of states, $\Sigma_V = \Sigma^+ \times \Sigma^+ \times \Sigma^+$ is the set of events, $x_{0,V} = (q_0, q_0, q_0)$ is the initial state, $X_{m,V}$ is the set of marked states defined by

$$X_{m,V} := \{(q_1, q_2, q_3) \in X_V : q_1 \in \partial_K(H, G) \wedge q_2, q_3 \notin \Im_M^<(H, G)\}. \tag{8}$$

$f_V : X_V \times \Sigma_V \to X_V$ is the partial (deterministic) transition function defined as follows:

For $\sigma \in \Sigma_{o,1} \cap \Sigma_{o,2}$

$$f_V((q_1, q_2, q_3), (\sigma, \sigma, \sigma)) = (\delta_H(q_1, \sigma), \delta_H(q_2, \sigma), \delta_H(q_3, \sigma)).$$

For $\sigma \in \Sigma_{o,1} \setminus \Sigma_{o,2}$

$$f_V((q_1, q_2, q_3), (\sigma, \sigma, \epsilon)) = (\delta_H(q_1, \sigma), \delta_H(q_2, \sigma), \delta_H(q_3, \epsilon))$$
$$f_V((q_1, q_2, q_3), (\epsilon, \epsilon, \sigma)) = (\delta_H(q_1, \epsilon), \delta_H(q_2, \epsilon), \delta_H(q_3, \sigma)).$$

For $\sigma \in \Sigma_{o,2} \setminus \Sigma_{o,1}$

$$f_V((q_1, q_2, q_3), (\sigma, \epsilon, \sigma)) = (\delta_H(q_1, \sigma), \delta_H(q_2, \epsilon), \delta_H(q_3, \sigma))$$
$$f_V((q_1, q_2, q_3), (\epsilon, \sigma, \epsilon)) = (\delta_H(q_1, \epsilon), \delta_H(q_2, \sigma), \delta_H(q_3, \epsilon)).$$

For $\sigma \in \Sigma_{uo}$

$$f_V((q_1, q_2, q_3), (\sigma, \epsilon, \epsilon)) = (\delta_H(q_1, \sigma), \delta_H(q_2, \epsilon), \delta_H(q_3, \epsilon))$$
$$f_V((q_1, q_2, q_3), (\epsilon, \sigma, \epsilon)) = (\delta_H(q_1, \epsilon), \delta_H(q_2, \sigma), \delta_H(q_3, \epsilon))$$
$$f_V((q_1, q_2, q_3), (\epsilon, \epsilon, \sigma)) = (\delta_H(q_1, \epsilon), \delta_H(q_2, \epsilon), \delta_H(q_3, \sigma)).$$

Remark 4. The above construction follows the idea of the twin machine studied in the literature for the purpose of fault diagnosis; see, e.g., (Jiang, Huang, Chandra, & Kumar, 2001; Moreira, Jesus, & Basilio, 2011; Qiu & Kumar, 2006; Yoo & Lafortune, 2002). Roughly speaking, the (M, K)-verifier is constructed according to the following two rules: (1) if the system's component moves, i.e., the first component of event $e \in \Sigma_v$ is not ϵ, then any agent $i \in \{1, 2\}$ such that $\sigma \in \Sigma_{o,i}$ should move together with the system, i.e., the $i+1$th component of event $e \in \Sigma_V$ is not ϵ; and (2) for each agent $i \in \{1, 2\}$ if $\sigma \notin \Sigma_{o,i}$, then the corresponding component can move by itself. Therefore, the (M, K)-verifier essentially tracks the system's execution and strings that look identical for each agent. Specifically, the first component in $\mathcal{L}(V)$ is a string generated by the system and the second (respectively, third) component is a string for Agent 1 (respectively, Agent 2) that has the same projection as the system's string under its local projection.

The following result states how to use the (M, K)-verifier to verify (M, K)-coprognosability.

Theorem 2. *A specification language $\mathcal{L}(H)$ is (M, K)-coprognosable iff $\mathcal{L}_m(V) = \emptyset$.*

Proof. ($\Rightarrow$) By contrapositive. Suppose that $\mathcal{L}_m(V) \neq \emptyset$, i.e., $\exists s = (s_1, s_2, s_3) \in \mathcal{L}(V)$ such that $f_V(x_{0,V}, s) \in X_m$, i.e., $\delta_H(s_1) \in \partial_K(H, G)$ and $\delta_H(s_2), \delta_H(s_3) \notin \Im_M^<(H, G)$. By the construction of V, we have $P_1(s_1) = P_1(s_2)$. To see this, assume that $P_1(s_1) \neq P_2(s_2)$. Then, we know that $\exists(s_1', s_2', s_3')(\sigma_1, \sigma_2, \sigma_3) \in \overline{\{s\}}$ where $(s_1', s_2', s_3') \in \Sigma_V^*$, $(\sigma_1, \sigma_2, \sigma_3) \in \Sigma_V$, such that $P_1(s_1') = P_2(s_2')$ but $P_1(\sigma_1) \neq P_2(\sigma_2)$. However, in the construction of V, we see that $(\sigma_1, \sigma_2, \sigma_3)$ is defined at $f(x_{0,V}, (s_1', s_2', s_3'))$ only if $P_1(\sigma_1) = P_2(\sigma_2)$. Therefore, $P_1(\sigma_1) \neq P_2(\sigma_2)$ is not possible. Moreover, $P_1(s_1) = P_1(s_2)$ implies $\{\delta_H(s_1), \delta_H(s_2)\} \subseteq \mathcal{E}_i^H(s_1)$. However, since $\delta_H(s_2) \notin \Im_M^<(H, G)$, we have that $\mathcal{E}_1^H(s_1) \not\subseteq \Im_M^<(H, G)$. Similarly, we have that $\mathcal{E}_2^H(s_1) \not\subseteq \Im_M^<(H, G)$. Therefore, $\mathcal{L}(H)$ is not (M, K)-coprognosable.

($\Leftarrow$) By contrapositive. Suppose that $\mathcal{L}(H)$ is not (M, K)-coprognosable. Then $\exists s \in \mathcal{L}(H) : \delta_H(s) \in \partial_K(H, G)$ and $\forall i \in \mathcal{I} : \mathcal{E}_i^H(s) \not\subseteq \Im_M^<(H, G)$. For Agent 1, since $\mathcal{E}_1^H(s) \not\subseteq \Im_M^<(H, G)$, we know that there must exist a string s_1 such that $P_1(s) = P_1(s_1)$ and $\delta_H(s_1) \notin \Im_M^<(H, G)$. Similarly, for Agent 2, there also exists a string s_2 such that $P_2(s) = P_2(s_2)$ and $\delta_H(s_2) \notin \Im_M^<(H, G)$. Since $P_1(s) = P_1(s_1)$ and $P_2(s) = P_2(s_2)$, by the construction of V, we know that state $(\delta_H(s), \delta_H(s_1), \delta_H(s_2))$ can be reached. Moreover, it is a marked state since $\delta_H(s) \in \partial_K(H, G)$ and $\delta_H(s_i) \notin \Im_M^<(H, G)$, $\forall i = 1, 2$. Therefore, $\mathcal{L}_m(V) \neq \emptyset$. □

The following example illustrates how to verify (M, K)-coprognosability by using the (M, K)-verifier.

Example 1. Consider the system G in Fig. 1(a), where state F denotes the single fault state, i.e., the specification automaton H is obtained by removing F from G and taking the accessible part. We assume that $\mathcal{I} = \{1, 2\}$ and let $\Sigma_{o,1} = \{a, o\}$ and $\Sigma_{o,2} = \{b, o\}$, i.e., e and f are both globally unobservable. We consider $M = 4$, $K = 2$ and we want to verifier whether or not the specification is $(4, 2)$-coprognosable. Part of the verifier V of this system is shown in Fig. 1(b). We see that state $(6, 2, 5)$ is reached. However, for state 6, we know that $d_{min}(6) = 2$, i.e., $6 \in \partial_K(H, G)$. Moreover, since both states 2 and 5 are in a cycle of H, we know that $d_{max}(2) = d_{max}(5) = \infty$, i.e., $2, 5 \notin \Im_M^<(H, G)$. Therefore, state $(6, 2, 5)$ is a marked state in V and this system is not $(4, 2)$-coprognosable.

We conclude this section by discussing the complexity of the above verification procedure. For each state $q \in Q_H$, in order to compute $d_{min}(q)$, we just need to find the shortest path from q to a state in $Q \setminus Q_H$, which can be done in $O(|Q_H||\Sigma|)$; see, e.g., (Sedgewick & Wayne, 2011) (p. 652). To compute $d_{max}(q)$, first, we need to computed all strongly connected components, i.e., cycles, which can be done by Kosaraju's algorithm in $O(|Q_H||\Sigma|)$; see, e.g., (Sedgewick & Wayne, 2011) (p. 587). Then, for any state q in a cycle, we assign $d_{max}(q) = \infty$. Next, we take a backwards search from states with infinite value in G, and assign each state reachable in the backwards search an infinite value, since this means that this state can reach a cycle. This step is just s depth-first search, which is still in $O(|Q_H||\Sigma|)$. Finally, we remove all states with infinite value from H and the resulting system is acyclic, since all states in cycles have been removed. Then we used the longest path search algorithm for the acyclic case provided in Sedgewick and Wayne (2011) (p. 661) to determine the values of states remained. The complexity of this step is still $O(|Q_H||\Sigma|)$. In the worst case, the (M, K)-verifier has $|Q_H|^3$ states and $3|\Sigma||Q_H^3|$ transitions. Therefore, constructing V can be done in $O(|\Sigma||Q_H^3|)$. Checking $\mathcal{L}_m(G) = \emptyset$ is linear in the size of V. Therefore, the total worst-time complexity of verifying (M, K)-coprognosability is $O(|\Sigma||Q_H^3|)$.

4. Fault prognosis using conjunctive architecture

In the above development, the global decision of the decentralized prognoser is "1" iff there exists one local agent whose local decision is "1". This decentralized information structure is usually referred to as the *disjunctive architecture*. However, one can also use the *conjunctive architecture* in order to fuse the local decisions. Specifically, under the conjunctive architecture, the decentralized prognoser $\{\mathcal{A}_i\}_{i\in\mathcal{I}} : \mathcal{L}(H) \to \{0, 1\}$ is defined by: for any string $s \in \mathcal{L}(H)$

$$\{\mathcal{A}_i\}_{i\in\mathcal{I}}(s) = 1 \Leftrightarrow \forall i \in \mathcal{I} : \mathcal{A}_i(P_i(s)) = 1. \tag{9}$$

Hereafter, we refer to (M, K)-coprognosability defined in definition (6) as (M, K)-disjunctive coprognosability. We show how the results developed for the disjunctive architecture can be extended to the conjunctive architecture.

Definition 4. $\mathcal{L}(H)$ is said to be (M, K)-conjunctively coprognosable w.r.t. $\mathcal{L}(G)$ and $\Sigma_{o,i}$, $i \in \mathcal{I}$ if for any string $s \in \mathcal{L}(H)$,

$$\delta_H(s) \notin \Im_M^<(H, G) \Rightarrow (\exists i \in \mathcal{I})[\mathcal{E}_i^H(s) \cap \partial_K(H, G) = \emptyset]. \tag{10}$$

Similar to the disjunctive case, the above definition requires that for any string that ends up with a state that is not in $\Im_M^<(H, G)$, there must exist one agent that knows for sure that the current state is not in $\partial_K(H, G)$. Therefore, it will not make a wrong local prognostic decision. The following result reveals that (M, K)-conjunctive coprognosability provides the necessary and sufficient condition for the existence of a (M, K)-decentralized prognoser under the conjunctive architecture.

Theorem 3. *Under the conjunctive architecture, there exists a decentralized prognoser $\{\mathcal{A}_i\}_{i\in\mathcal{I}}$ satisfying Eqs. (2) and (3), if and only if, $\mathcal{L}(H)$ is (M, K)-conjunctively coprognosable w.r.t. $\mathcal{L}(G)$ and $\Sigma_{o,i}$, $i \in \mathcal{I}$.*

Proof. ($\Leftarrow$) By construction. Let us consider a decentralized prognoser $\{\mathcal{A}_i\}_{i\in\mathcal{I}}$ as follows. For each $i \in \mathcal{I}$, we define

$$\mathcal{A}_i(P_i(s)) = 1 \Leftrightarrow \mathcal{E}_i^H(s) \cap \partial_K(H, G) \neq \emptyset. \tag{11}$$

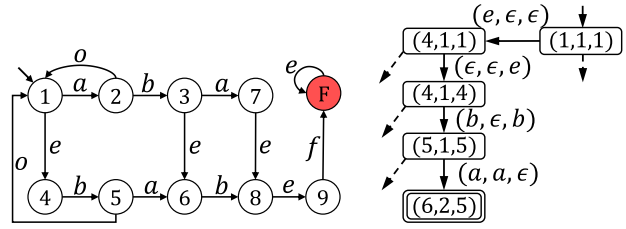
First, for any string $s \in \partial_K(H, G)$, we have $\delta_H(s) \in \mathcal{E}_i^H(s)$, $\forall i \in \mathcal{I}$. Therefore, $\mathcal{E}_i^H(s) \cap \partial_K(H, G) \neq \emptyset$, $\forall i \in \mathcal{I}$ implies that $\mathcal{A}_i(P_i(s)) = 1$, $\forall i \in \mathcal{I}$. Hence, $\{\mathcal{A}_i\}_{i\in\mathcal{I}}(P_i(s)) = 1$ and Eq. (2) holds. Second, we show Eq. (3) holds by contradiction. We assume that Eq. (3) does not hold, which implies that $\exists s \in \mathcal{L}(H) : [\{\mathcal{A}_i\}_{i\in\mathcal{I}}(s) = 1]$ and $(\exists t \in \mathcal{L}(G)/s)[|t| \geq M \wedge st \in \mathcal{L}(H)]$. This implies that $\delta_H(s) \notin \Im_M^<(H, G)$. By Definition 4, we know that $\exists i \in \mathcal{I} : \mathcal{E}_i^H(s) \cap \partial_K(H, G) = \emptyset$, which means that at least one agent's decision is "0". Therefore, $\{\mathcal{A}_i\}_{i\in\mathcal{I}}(s) = 0$, which is a contradiction.

($\Rightarrow$) By contradiction. We assume that there exists $\{\mathcal{A}_i\}_{i\in\mathcal{I}}$ but $\mathcal{L}(H)$ is not (M, K)-conjunctively coprognosable, i.e., $(\exists s \in \mathcal{L}(H) : \delta_H(s) \notin \Im_M^<(H, G))(\forall i \in \mathcal{I})[\mathcal{E}_i^H(s) \cap \partial_K(H, G) \neq \emptyset]$. For the above strings s, we know that $\forall i \in \mathcal{I}, \exists s_i \in \mathcal{L}(H) : P_i(s) = P_i(s_i) \wedge \delta_H(s_i) \in \partial_K(H, G)$. Since $\{\mathcal{A}_i\}_{i\in\mathcal{I}}$ cannot predict fault more than M steps head, by Eq. (3), we know that $\{\mathcal{A}_i\}_{i\in\mathcal{I}}(s) = 0$, i.e., $\forall i \in \mathcal{I} : \mathcal{A}_i(P_i(s)) = 0$. Since $P_i(s) = P_i(s_i)$, we know that $\mathcal{A}_i(P_i(s_i)) = \mathcal{A}_i(P_i(s)) = 0$, i.e., $\{\mathcal{A}_i\}_{i\in\mathcal{I}}(s_i) = 0$. However, we know that $\delta_H(s_i) \in \partial_K(H, G)$, which means that a fault can occur from $\delta_H(s_i)$ within K steps. This contradicts the condition in Eq. (2). □

Remark 5. To verify (M, K)-conjunctive coprognosability, we can still use the (M, K)-verifier proposed in the previous section. In this case, instead of defining the set of marked states $X_{m,V}$ according to Eq. (8), one can define $X_{m,V}$ by

$$X_{m,V} := \{(q_1, q_2, q_3) \in X_V : q_1 \notin \Im_M^<(H, G) \wedge q_2, q_3 \in \partial_K(H, G)\}.$$

It is easy to verify that $\mathcal{L}(H)$ is (M, K)-conjunctively coprognosable iff $\mathcal{L}_m(V) = \emptyset$ for the modified V. The proof is omitted here since it is similar to the proof of Theorem 2.



(a) G. (b) V.

Fig. 1. An illustrative example for (M, K)-coprognosability.

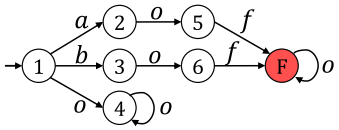


Fig. 2. For the above system, where state F denotes the single fault state. We assume that $\mathbb{I} = \{1, 2\}$ and let $\Sigma_{o,1} = \{a, o\}$ and $\Sigma_{o,2} = \{b, o\}$. We consider $M = 1$ and $K = 0$. Then we have that $\Im_M^<(H, G) = \{2, 3, 5, 6\}$ and $\partial_K(H, G) = \{5, 6\}$. Let us consider string o. Then we know that $\delta_H(o) = 4 \notin \Im_M^<(H, G)$. However, $\mathcal{E}_1^H(o) \cap \partial_K(H, G) = \{4, 6\} \cap \{5, 6\} \neq \emptyset$ and $\mathcal{E}_2^H(o) \cap \partial_K(H, G) = \{4, 5\} \cap \{5, 6\} \neq \emptyset$. Therefore, the system is not $(1, 0)$-conjunctive coprognosable. However, it is $(1, 0)$-disjunctive coprognosable, since a fault alarm can be issued by Agent 1 if event a occurs and by Agent 2 if event b occurs.

The above results generalize the conjunctive prognosis studied in Khoumsi and Chakib (2012) in twofold. First, in Khoumsi and Chakib (2012), only the lower bound K is guaranteed for the performance of the prognostic system while our results guarantee both lower bound K and upper bound M. Second, although the notion of conjunctive coprognosability was introduced in Khoumsi and Chakib (2012), there is no verification algorithm provided so far in the literature. Clearly, as a special case of (M, K)-conjunctive coprognosability, conjunctive coprognosability can also be effectively verified by using the (M, K)-verifier as we discussed in the above remark.

Remark 6. In fact, one can verify that the system in Fig. 1, which is not $(4, 2)$-disjunctively coprognosable, is $(4, 2)$-conjunctively coprognosable. It is also not difficult to find a system that is not (M, K)-conjunctively coprognosable but (M, K)-disjunctively coprognosable; an example for this is provided in Fig. 2. Therefore, (M, K)-disjunctive coprognosability and (M, K)-conjunctive coprognosability are *incomparable*. By comparing Eqs. (7) and (11), we see that two different prognostic strategies are used for these two different architectures, respectively. For the disjunctive case, the strategy we take is "alarm as early as possible", since according to Eq. (7), a fault alarm is issued immediately when one agent knows for sure that the fault is inevitable within M steps. However, for the conjunctive case, the strategy we take is "alarm as late as possible", since according to Eq. (11), such a fault alarm will not be issued until a state in $\partial_K(H, G)$ is reached. Clearly, we cannot postpone the alarm any more; otherwise, the condition in Eq. (2) will be violated.

5. Conclusion

In this note, we extended previous work on the problem of decentralized fault prognosis by taking the prognostic performance issue into account. The notion of (M, K)-coprognosability was introduced as the necessary and sufficient condition for the existence of a decentralized prognoser satisfying certain time constraints. A polynomial-time algorithm for the verification of (M, K)-coprognosability was provided. We showed that the proposed approach is also applicable to the conjunctive architecture for which no verification algorithm is provided so far. Note that the prognostic performance criteria considered in the paper are logical; investigating numerical performance criteria for real-time DES is also an interesting future direction.

Acknowledgments

The authors would like to thank the anonymous reviewers for their useful comments on improving this paper.

References

Cassandras, C., & Lafortune, S. (2008). *Introduction to discrete event systems* (2nd ed.). Springer.
Cassez, F., & Grastien, A. (2013). Predictability of event occurrences in timed systems. In *Formal modeling and analysis of timed systems* (pp. 62–76). Springer.
Chang, M., Dong, W., Ji, Y., & Tong, L. (2013). On fault predictability in stochastic discrete event systems. *Asian Journal of Control*, *15*(5), 1458–1467.
Chen, J., & Kumar, R. (2015). Stochastic failure prognosability of discrete event systems. *IEEE Transactions on Automatic Control*, *60*(6), 1570–1581.
Genc, S., & Lafortune, S. (2009). Predictability of event occurrences in partially-observed discrete-event systems. *Automatica*, *45*(2), 301–311.
Jéron, T., Marchand, H., Genc, S., & Lafortune, S. (2008). Predictability of sequence patterns in discrete event systems. In *Proc. 17th IFAC World Congress* (pp. 537–543).
Jiang, S., Huang, Z., Chandra, V., & Kumar, R. (2001). A polynomial algorithm for testing diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, *46*(8), 1318–1321.
Khoumsi, A., & Chakib, H. (2009). Multi-decision decentralized prognosis of failures in discrete event systems. In *ACC* (pp. 4974–4981).
Khoumsi, A., & Chakib, H. (2012). Conjunctive and disjunctive architectures for decentralized prognosis of failures in discrete-event systems. *IEEE Transaction on Automatic Science and Engineering*, *9*(2), 412–417.
Kumar, R., & Takai, S. (2010). Decentralized prognosis of failures in discrete event systems. *IEEE Transactions on Automatic Control*, *55*(1), 48–59.
Lefebvre, D. (2014). Fault diagnosis and prognosis with partially observed petri nets. *IEEE Transactions on Systems, Man and Cybernetics: Systems*, *44*(10), 1413–1424.
Lefebvre, D. (2014). Probability of current state and future faults with partially observed stochastic petri nets. In *Euro. Contr. Conf.* (pp. 258–263).
Moreira, M. V., Jesus, T. C., & Basilio, J. C. (2011). Polynomial time verification of decentralized diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, *56*(7), 1679–1684.
Nouioua, F., Dague, P., & Ye, L. (2014). Probabilistic analysis of predictability in discrete event systems. In *25th Int. Workshop on Princ. Diagnosis*.
Qiu, W., & Kumar, R. (2006). Decentralized failure diagnosis of discrete event systems. *IEEE Transactions on Systems, Man and Cybernetics: Part A*, *36*(2), 384–395.
Sedgewick, R., & Wayne, K. (2011). *Algorithms* (4th ed.). Addison-Wesley Professional.
Takai, S. (2015). Robust prognosability for a set of partially observed discrete event systems. *Automatica*, *51*, 123–130.
Takai, S., & Kumar, R. (2011). Inference-based decentralized prognosis in discrete event systems. *IEEE Transactions on Automatic Control*, *56*(1), 165–171.
Ye, L., Dague, P., & Nouioua, F. (2013). Predictability analysis of distributed discrete event systems. In *52nd IEEE CDC* (pp. 5009–5015).
Yin, X., & Lafortune, S. (2015). A general approach for solving dynamic sensor activation problems for a class of properties. In *54th IEEE conference on decision and control* (pp. 3610–3615).
Yoo, T.-S., & Lafortune, S. (2002). Polynomial-time verification of diagnosability of partially observed discrete-event systems. *IEEE Transactions on Automatic Control*, *47*(9), 1491–1495.