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