2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Fault Diagnosis and Prognosis in Partially-Observed Discrete Event Systems with Delayed Observations

Jiwei Wang, Simone Baldi, Wenwu Yu, Xiang Yin

Abstract— **Fault diagnosis and prognosis in discrete event systems are studied in the scenario where the observations are possibly received with delay. To address this scenario, two conditions for diagnosis and prognosis with delayed observations are proposed, where we show that the state-of-the-art notion of prognosability must be revised to avoid conservativeness. Diagnosability and prognosability conditions are then verified by introducing a delay observer and a new verification function. Theoretical analysis indicates the effectiveness of the verification method for fault diagnosis and prognosis in the system.**

I. INTRODUCTION

Fault analysis is fundamental for system reliability and maintenance. Discrete event systems (DES) [1] provide a useful framework for fault analysis: structured DES approaches to fault analysis have been proposed using observations of the system to detect (fault diagnosis) or predict (fault prognosis) anomalies and deviations from healthy operation. Notions of diagnosability [2] and prognosability [3] have been presented for DES, where the former determines the ability to timely detect faults, while the latter determines the ability to forecast future faults based on current system conditions. Prognosability is also called predictability in some literature.

With DES encompassing applications across various domains [4], [5], the literature has proposed several settings for DES fault diagnosis and prognosis, spanning decentralized scenarios [6], [7], robustness notions [8], [9], fuzzy systems [10], [11], fault tolerant control [12], [13], among others. The interested reader can refer to [14], [15] for more details. A fundamental setting in fault analysis is handling observations received with delay. Such issues have been addressed in network systems [16]–[18], and distributed scenarios with transmission delay [19]–[21], introducing notions of codiagnosability and coprognosability. Yet, these results do not encompass the delayed-observation scenario considered in this work. This point will be discussed making use of a suitable example (cf. Example 2&3 in this manuscript).

To consider the delayed-observation scenario, one needs to extend state-of-the-art notions of diagnosability in K steps, such as K-diagnosability [22]. We do this by introducing a new K^T-diagnosability, where T is a delay-dependent parameter. We show that the state-of-the-art notion of prognosability [3] cannot be extended to the delayed-observation scenario. We thus propose a notion of prognosability that, while being equivalent to the state-of-the-art prognosability in the absence of delays, it can be extended to the delayed-observation scenario, resulting in T-constrained prognosability. We finally propose a delay observer structure to record the delays required for diagnosis and prognosis, which in turn leads to a new function to verify K^T-diagnosability and T-constrained prognosability.

The remainder of this paper is organized as follows. In Section II, the partially-observed DES is introduced. Section III recalls the state-of-the-art notions for diagnosability and prognosability, and proposes novel definitions of K^T-diagnosability and T-constrained prognosability. In Section IV, we verify such conditions by introducing the delay observer and the verification function. Section V concludes this paper.

II. PRELIMINARIES

Let us start by considering a finite event set E as an alphabet, enabling us to interpret the *concatenation* of word strings within the alphabet as finite sequences of events in E. Let E^* denote the set of all finite strings over E. The length of a string $s \in E^*$ is denoted as $|s|$, and let ϵ represent the empty string with $|\epsilon| = 0$. A *language* L constitutes a collection of event strings, derived from events in E. The *prefix-closure* of a language L is defined as $\overline{L} = \{s \in E^* | \exists t \in E^*, \text{ s.t. } st \in L\}$, and L is *prefix-closed* if $L = \overline{L}$. The *post-language* of L following a string s is denoted as $L \backslash s = \{s' \in E^* \mid ss' \in L\}$. We say that a language L is *live* if, for every $s \in L$, there exists an $e \in E$ such that $se \in L$, representing that any string in L can be extended to any length.

Automata are a common framework for manipulating languages, used to model DES. Let us consider the finite automaton

$$G = (X, E, \alpha, X_0), \tag{1}$$

where X is the set of finite states, E is the set of finite events, $\alpha : X \times E^* \to 2^X$ is the transition function describing the transition of an event string, $X_0 \subseteq X$ is the set of possible initial states. The language generated by G from state $x \in X$ is denoted by $\mathcal{L}(x, G) = \{s \in E^* | \alpha(x, s)!\}$, where ! indicates that the string s "is defined", meaning that it can occur starting from state x. If $x_0 \in X_0$, we simply denote

This work was supported in part by the Jiangsu Provincial Scientific Research Center of Applied Mathematics grant BK20233002, and in part by the National Natural Science Foundation of China grants 62150610499, 62073074, 62073076, 62233004, 62173226.

J. Wang is with School of Cyber Science and Engineering, Southeast University, Nanjing 210096, China (e-mails: jwwang@seu.edu.cn).

S. Baldi and W. Yu are with School of Mathematics, Southeast University, Nanjing 210096, China (e-mails: {simonebaldi,wwyu}@seu.edu.cn).

X. Yin is with the Department of Automation and Key Laboratory of System Control and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: yinxiang@sjtu.edu.cn).

$\alpha(x_0, s)$ as $\alpha(s)$ and $\mathcal{L}(x_0, G)$ as $\mathcal{L}(G)$.

In practice, the occurrence of certain events may not be detectable. Accordingly, we partition E into the observable events E_o and the unobservable events E_{uo}. Let us now introduce two projection operators. The operator P_{E_o} is utilized for acquiring the observable events in an event string: $\forall s \in \mathcal{L}(G), \forall e \in E : \alpha(se)!$,

$$P_{E_o}(\epsilon) = \epsilon, P_{E_o}(se) = \begin{cases} P_{E_o}(s)e, & \text{if } e \in E_o; \\ P_{E_o}(s), & \text{if } e \notin E_o. \end{cases} \tag{2}$$

Intuitively, for any system trajectory $s \in \mathcal{L}(G)$, $P_{E_o}(s)$ gives the observed events. We extend P_{E_o} to handle a set of event string, that is, $\forall S \subseteq \mathcal{L}(G)$, $P_{E_o}(S) = \{s \in E_o^* | \exists s' \in S, \text{ s.t. } s = P_{E_o}(s')\}$. We then introduce the operator $\zeta^n_{E_o}$: $\forall s \in \mathcal{L}(G)$,

$$\zeta^n_{E_o}(s) = \{s'' \in \overline{s} | \exists s' \in \overline{s} : |s'| \geq |s| - n, \text{s.t.} P_{E_o}(s'') = P_{E_o}(s')\},$$

where $n \in \mathbb{N}$ is the number of steps. Intuitively, $\zeta^n_{E_o}$ collects all prefixes of s which have the same observation as a system trajectory $s' \in \overline{s}$ under the observation ability E_o. Obviously, given $n_1 \geq n_2$, we have $\zeta^{n_2}_{E_o}(s) \subseteq \zeta^{n_1}_{E_o}(s)$.

A. Illustrative example

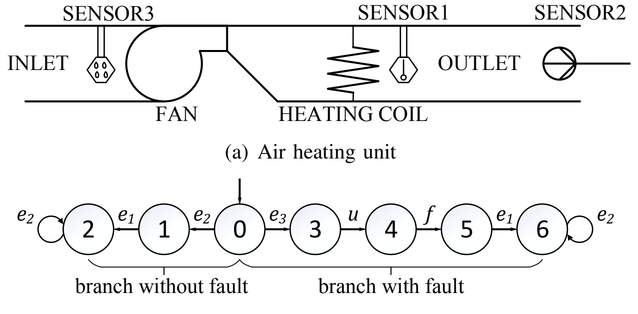
Throughout this paper, we take an air heating unit example to illustrate the key concepts. The example describes a start-up process of the unit via the automaton G in Fig. 1(b), where the system is off initially (in state $x_0 = \{0\}$). In healthy conditions, after turning the air heating unit on, the air flow sensor 2 observes that the flow rate is regular ($e_2, 0 \rightarrow 1$: e_2 occurs and the system state goes from 0 to 1); after some time, the temperature sensor 1 observes that the desired temperature is reached ($e_1, 1 \rightarrow 2$). Then, sensor 2 keeps monitoring the flow rate ($e_2, 2 \rightarrow 2$). However, if the air quality sensor 3 observes an abnormal amount of dust (e.g., possibly due to aging or some leak in the duct) ($e_3, 0 \rightarrow 3$), after some unobservable anomalies (e.g., unusual vibration of the fan) ($u, 3 \rightarrow 4$), the fan may fail or clog ($f, 4 \rightarrow 5$). As a result, sensor 1 observes high temperature ($e_1, 5 \rightarrow 6$), leading to overheating of the coil with no flow. This damage may be irreversible even in the scenario that the fan recovers from the clog and restores the flow rate ($e_2, 6 \rightarrow 6$).

Example 1: (System model). Using the automaton formalism in (1), we have

$$G = (\{0, 1, 2, 3, 4, 5, 6\}, \{e_1, e_2, e_3, u, f\}, \alpha, \{0\}).$$

When the system is in state 5, the only event that can occur is e_1, that is, $\alpha(5, e_1)!$. Let $E_o = \{e_1, e_2, e_3\}$ be the observable events. Then, for the string $e_3ufe_1e_2$ generated by G, we have

$$\begin{aligned} &P_{E_o}(e_3ufe_1e_2) = e_3e_1e_2, \\ &\zeta^0_{E_o}(e_3ufe_1e_2) = \{e_3ufe_1e_2\}, \\ &\zeta^2_{E_o}(e_3ufe_1e_2) = \{e_3, e_3u, e_3uf, e_3ufe_1, e_3ufe_1e_2\}. \end{aligned}$$



(a) Air heating unit

(b) Model G of Air Heating Unit

Fig. 1: Air heating unit and the model G of its start-up process.

III. Diagnosability and Prognosability with Delayed Observation

In this section, we first recall and then extend state-of-the-art notions of K-diagnosability and prognosability.

A. Diagnosis with Delay

Let K be the maximum number of steps permitted from the occurrence of a fault to its diagnosis. Let $G = (X, E, \alpha, x_0)$ be the system model. We introduce a structure of step counter $\Delta : \mathcal{L}(G) \rightarrow \{-1, 0, 1, \ldots, K\}$ to count the number of steps in an event string after a fault event f occurs: $\forall s \in \mathcal{L}(G), \forall e \in E : \alpha(se)! \Rightarrow \Delta(\epsilon) = -1, \Delta(se) =$

$$\begin{cases} \Delta(s), & \text{if } [\Delta(s) = -1 \wedge e \neq f] \vee [\Delta(s) = K]; \\ \Delta(s)+1, & \text{if } [\Delta(s) = -1 \wedge e = f] \vee [0 \leq \Delta(s) < K]; \end{cases} \tag{3}$$

where -1 means no fault happens. By means of Δ, let us recall the notion of K-diagnosability.

Definition 1: (K-diagnosability [22]) For $K \in \mathbb{N}$, a live language $\mathcal{L}(G)$ is K-diagnosable w.r.t. f if $\forall s \in \mathcal{L}(G)$: $\Delta(s) = K$,

$$\forall s' \in \mathcal{L}(G) : P_{E_o}(s') = P_{E_o}(s), \Delta(s') \neq -1. \tag{4}$$

Intuitively, (4) means that any string with the same observation as s must have the fault event f in it, which implies that the fault can be always diagnosed after it occurs within K steps.

Now we consider the scenario that the observed signals are processed and transferred with an unknown delay. We denote $D \geq 0$ as the transmission distance, and $T > 0$ as a coefficient related to transmission efficiency. If the observation site observes an event, then it will be received with a delay of no more than $\lceil \frac{D}{T} \rceil$ steps ($\lceil \cdot \rceil$ rounds the element to the nearest integer towards infinity). We give the following definition for fault diagnosis with delayed observation.

Definition 2: (K^T-diagnosability) For $K \in \mathbb{N}$, $T > 0$ and $D \geq 0$, a live language $\mathcal{L}(G)$ is K^T-diagnosable w.r.t. f and D if $\forall s \in \mathcal{L}(G) : \Delta(s) = K$,

$$\forall s' \in \mathcal{L}(G) : P_{E_o}(\zeta^{\lceil \frac{D}{T} \rceil}_{E_o}(s')) \cap P_{E_o}(\zeta^{\lceil \frac{D}{T} \rceil}_{E_o}(s)) \neq \emptyset, \Delta(s') \neq -1. \tag{5}$$

Remark 1: By comparing (4) with (5), the presence of delay enlarges the state set that the system may be in. If (5) holds, all key events to determine the occurrence of f can be timely received. Intuitively, K^T-diagnosability allows that any faulty string (s satisfying $\Delta(s) = K$) can be distinguished from a healthy string (s satisfying $\Delta(s) = -1$), despite the delay. As T increases, the strings that cannot be distinguished from the string $s : \Delta(s) = K$ become less and less, that is, less strings would satisfy (5). As expected, $K^{T'}$-diagnosability implies K^T-diagnosability when $T' \leq T$ (higher transmission efficiency improves diagnosis ability). Compared to existing work on diagnosability subject to observation delays [18], Definition 2 focuses on determining whether key events for diagnosis can be received in a timely manner. □

Example 2: (K^T-diagnosability in the presence of delay). Consider the air hearing unit in Sect. II-A. Delayed observations may happen when the air heating unit is part of a central energy management system where all observed signals are processed and transferred. Suppose $D = 3$ and $K = 3$. For $T = 1$, the occurrence of the events in E_o will be received in no more than $\lceil \frac{D}{T} \rceil = 3$ steps. Then, for the string $s = e_3ufe_1e_2e_2$ generated by G, we have $\Delta(s) = 3$ and $P_{E_o}(\zeta^3_{E_o}(s)) = \{e_3, e_3e_1, e_3e_1e_2, e_3e_1e_2e_2\}$. Since $P_{E_o}(\zeta^3_{E_o}(s)) \cap P_{E_o}(\zeta^3_{E_o}(e_3)) = \{e_3\}$ and $\Delta(e_3) = -1$, we have that $\mathcal{L}(G)$ is not K^T-diagnosable w.r.t. f and D for $K = 3, T = 1, D = 3$. However, for $T = 2$, we have $\lceil \frac{D}{T} \rceil = 2$ steps and $P_{E_o}(\zeta^2_{E_o}(s)) = \{e_3e_1, e_3e_1e_2, e_3e_1e_2e_2\}$. For any string s' with $\Delta(s') = -1$, we have $P_{E_o}(\zeta^2_{E_o}(s')) \cap P_{E_o}(\zeta^2_{E_o}(s)) = \emptyset$, implying that $\mathcal{L}(G)$ is K^T-diagnosable w.r.t. f and D for $K = 3, T = 2, D = 3$. □

B. Prognosis with Delay

Prognosability allows faults to be prognosed before their occurrence. The literature has introduced the following notion of prognosability.

Definition 3: (Prognosability [3]). A live language $\mathcal{L}(G)$ is prognosable w.r.t. f if $\forall sf \in \mathcal{L}(G) : s \in E^*$,

$$\begin{aligned}&\exists s' \in \bar{s} : \Delta(s') = -1, \text{s.t. } \forall t \in \mathcal{L}(G) :\\ &\Delta(t) = -1 \wedge P_{E_o}(t) = P_{E_o}(s'), \exists n \in \mathbb{N}, \text{s.t.}\\ &\forall t' \in \mathcal{L}(G) \backslash t, [|t'| \geq n \Rightarrow \Delta(tt') \geq 0].\end{aligned} \tag{6}$$

Intuitively, (6) means that any string that has the same observation as $s' \in \bar{s}$ will always lead to the occurrence of f, implying that we definitely know that the fault will happen after observing $P_{E_o}(s')$. Later we will show with an example that Definition 3 cannot handle delay well. We now introduce a different definition of prognosability that, while being equivalent to Definition 3 in the absence of delays, is extendable in the scenario of delayed observation.

Definition 4: (Prognosability). A live language $\mathcal{L}(G)$ is prognosable w.r.t. f if $\forall s \in \mathcal{L}(G) : \Delta(s) = 0$,

$$\begin{aligned}&\exists s' \in \bar{s}, \text{s.t.} \forall t \in \mathcal{L}(G) : \Delta(t) = -1 \wedge P_{E_o}(t) = P_{E_o}(s'),\\ &\exists n \in \mathbb{N}, \text{s.t.} \forall t' \in \mathcal{L}(G) \backslash t, [|t'| \geq n \Rightarrow \Delta(tt') \geq 0].\end{aligned} \tag{7}$$

The following lemma shows that Definition 4 is equivalent to Definition 3.

Lemma 1: Let G in (1) be the system model, f be the fault events. Then, for any $sf \in \mathcal{L}(G)$, (6) holds if and only if for any $s \in \mathcal{L}(G) : \Delta(s) = 0$, (7) holds.

Proof: As necessity is obvious, let us only consider sufficiency. Suppose that for any $s \in \mathcal{L}(G) : \Delta(s) = 0$, (7) holds. We first show that for any $s \in \mathcal{L}(G) : \Delta(s) = 0$, (6) holds. Consider s' in (7). If $\Delta(s') = -1$, then (6) holds directly. If $\Delta(s') \neq -1$, then there exists $s_1 \in \overline{s'} \subseteq \bar{s}$ such that $\Delta(s_1 f) = 0$, which implies that

$$\begin{aligned}&\exists s_2 \in \overline{s_1 f}, \text{s.t.} \forall t \in \mathcal{L}(G) : \Delta(t) = -1 \wedge P_{E_o}(t) = P_{E_o}(s_2),\\ &\exists n \in \mathbb{N}, \text{s.t.} \forall t' \in \mathcal{L}(G) \backslash t, [|t'| \geq n \Rightarrow \Delta(tt') \geq 0].\end{aligned} \tag{8}$$

Obviously, the only case that $\Delta(s_2) \neq -1$ is $s_2 = s_1 f$. Because $P_{E_o}(s_1 f) = P_{E_o}(s_1)$, we can set $s_2 = s_1$ to make (8) hold. Since $s_2 \in \overline{s_1 f} \subseteq \bar{s}$, we conclude that for any $s \in \mathcal{L}(G) : \Delta(s) = 0$, (6) holds. We then show that for any $sf \in \mathcal{L}(G) : s \in E^*$, (6) holds. If $\Delta(s) = -1$, then $\Delta(sf) = 0$, implying that (6) holds. If $\Delta(s) \neq -1$, then there exists $s_1 \in \bar{s}$ such that $\Delta(s_1 f) = 0$, which also implies that (6) holds. ■

Now we extend Definition 4 in the scenario of delayed observation, when the message from the observation site can be received in no more than $\lceil \frac{D}{T} \rceil$ steps.

Definition 5: (T-constrained prognosability). For $T > 0$ and $D \geq 0$, a live language $\mathcal{L}(G)$ is T-constrained prognosable w.r.t. f and D if $\forall s \in \mathcal{L}(G) : \Delta(s) = 0$,

$$\begin{aligned}&\exists s' \in \bar{s}, \text{ s.t. } \forall t \in \mathcal{L}(G) : \Delta(t) = -1 \wedge\\ &P_{E_o}(\zeta^{\lceil \frac{D}{T} \rceil}_{E_o}(t)) \cap P_{E_o}(\zeta^{\lceil \frac{D}{T} \rceil}_{E_o}(s')) \neq \emptyset, \exists n \in \mathbb{N},\\ &\text{s.t. } \forall t' \in \mathcal{L}(G) \backslash t, [|t'| \geq n \Rightarrow \Delta(tt') \geq 0].\end{aligned} \tag{9}$$

Remark 2: Due to the transmission delay, receiving the occurrence of an event before the fault f does not mean that f has not occurred yet. Hence, condition (9) means that any string that may lead to the same received observation as $s' \in \bar{s}$ will always lead to the occurrence of f. If (9) holds, all key events to determine if f will occur can be timely received. In other words, we definitely know that f will occur before its occurrence despite the delay. Note that T'-constrained prognosability implies T-constrained prognosability when $T' \leq T$. Compared to existing work on prognosability subject to observation delays [21], Definition 5 provides a less conservative definition, as illustrated below.

Example 3: (T-constrained prognosability in the presence of delay). Consider the system G in Fig. 1(b) and the string $s = e_3uf$ generated by G. Suppose $D = 3$ and $T = 2$. Then, we know that e_3 will be received within $\lceil \frac{D}{T} \rceil = 2$ steps before the occurrence of f, that is, f can be prognosed. Given $P_{E_o}(\zeta^2_{E_o}(e_3uf)) = \{e_3\}$, we have that the set of strings t satisfying $\Delta(t) = -1 \wedge P_{E_o}(\zeta^2_{E_o}(t)) \cap \{e_3\} \neq \emptyset$ is $\{e_3, e_3u\}$, which will lead to f in 2 steps. Hence, we conclude that $\mathcal{L}(G)$ is T-constrained prognosable w.r.t. f and D for $T = 2$ and $D = 3$. Let us now consider the extension from Definition 3, which is equivalent to [21, Definition 3] in the centralized case: $\forall sf \in \mathcal{L}(G) : s \in E^*$,

$$\begin{aligned}
&\exists s' \in \bar{s} : \Delta(s') = -1, \text{s.t. } \forall t \in \mathcal{L}(G) : \\
&\Delta(t) = -1 \wedge P_{E_o}(\zeta_{E_o}^{\lceil \frac{D}{T} \rceil}(t)) \cap P_{E_o}(\zeta_{E_o}^{\lceil \frac{D}{T} \rceil}(s')) \neq \emptyset, \quad (10)\\
&\exists n \in \mathbb{N}, \text{s.t.} \forall t' \in \mathcal{L}(G) \backslash t, [|t'| \geq n \Rightarrow \Delta(tt') \geq 0].
\end{aligned}$$

For the string $s = e_3uf$, we have that the set of strings $s' \in \bar{s} : \Delta(s') = -1$ is $\{\epsilon, e_3, e_3u\}$ and $\forall s' \in \{\epsilon, e_3, e_3u\}, \epsilon \in P_{E_o}(\zeta_{E_o}^2(s'))$. However, if we consider the string e_2, we have $\epsilon \in P_{E_o}(\zeta_{E_o}^2(e_2))$ and

$$\forall n \in \mathbb{N}, \exists t' \in \mathcal{L}(G) \backslash e_2 : |t'| \geq n, \text{s.t.} \Delta(e_2t') = -1,$$

implying that s does not satisfy (10). In other words, the extension (10) of Definition 3 fails to capture that f can be prognosed. This justifies the introduction of Definition 4 and its extension in Definition 5. □

IV. Verification of Diagnosability and Prognosability

In general, it is intractable to determine if a fault can be diagnosed or prognosed by analysing each event string. It is necessary to embed the delay information into the automaton and develop a feasible method to verify K^T-diagnosability and T-constrained prognosability. This is done by linking the system states to fault events and by building a delay observer structure to handle the delays.

A. Delay Observer

A delay observer aims to register the delays of the events that help to distinguish the faulty from the healthy strings. To construct a delay observer, a fault automaton structure [23] is utilized to track the number of steps following a fault occurrence

$$\hat{G} = (\hat{X}, E, \hat{\alpha}, \hat{x}_0), \tag{11}$$

where $\hat{X} = X \times \{-1, 0, 1, \ldots, K\}$ comprises the state in X of G along with the fault counting component, denoted as $\hat{x} = (x, |\hat{x}|_f) \in \hat{X}$ where $|\hat{x}|_f$ represents the number of steps following a fault occurrence. The transition function $\hat{\alpha} : \hat{X} \times E \to \hat{X}$ is specified as follows: $\forall \hat{x} = (x, |\hat{x}|_f) \in \hat{X}, e \in E : \alpha(x, e)!$,

$$\hat{\alpha}((x, |\hat{x}|_f), e) = (\alpha(x, e), |\hat{x}|_f + v),$$

where $|\hat{x}|_f \in \{-1, 0, 1, \ldots, K\}$ and v is determined by

$$v = \begin{cases} 0, & \text{if } [|\hat{x}|_f = -1 \wedge e \neq f] \vee [|\hat{x}|_f = K]; \\ 1, & \text{if } [|\hat{x}|_f = -1 \wedge e = f] \vee [0 \leq |\hat{x}|_f < K]. \end{cases}$$

The initial state is $\hat{x}_0 = (x_0, -1)$. Note that $\mathcal{L}(\hat{G}) = \mathcal{L}(G)$. In both diagnosis and prognosis problems, we aim to distinguish two set of states, denoted by $\hat{X}^F$ (faulty) and $\hat{X}^H$ (healthy). In the diagnosis problem, we distinguish

$$\begin{aligned}
\hat{X}^F &= \{\hat{x} \in \hat{X} \mid |\hat{x}|_f = K\}, \\
\hat{X}^H &= \{\hat{x} \in \hat{X} \mid |\hat{x}|_f = -1\},
\end{aligned} \tag{12}$$

while in the prognosis problem, we distinguish

$$\begin{aligned}
\hat{X}^F &= \{\hat{x} \in \hat{X} \mid |\hat{x}|_f = 0\}, \\
\hat{X}^H &= \{\hat{x} \in \hat{X} \mid \forall n \in \mathbb{N}, \exists s \in \mathcal{L}(\hat{x}, \hat{G}) : |s| > n, \\
&\qquad \text{s.t. } \Delta(s) = -1\}.
\end{aligned} \tag{13}$$

The delay observer to determine the delays to be recorded is defined as an automaton

$$\mathcal{O}_{E_o}(\hat{G}) = (Y, E_o, \beta, y_0), \tag{14}$$

where each state $y \in 2^{\hat{X} \times \{0,1,\ldots,\lceil \frac{|a_1a_2|}{T} \rceil, \infty\}}$ represents the state estimation with delay value. The transition function $\beta : Y \times E_o \to Y$ and the initial state y_0 are built as in Algorithm 1. Algorithm 1 collects the initial states with the delay value ∞ into y_0, and uses two iterative procedures to build the delay observer. The procedure $\texttt{RECORD1}(\hat{x}, u, y)$ collects the state estimation for y while appending the delay value to each state in the state estimation and collecting them in y. The procedure $\texttt{RECORD2}(y)$ explores new transitions and observer states while appending the delay value to the "initial" states in the state estimation.

The delay observer has the same structure as a classical observer [1], where the only difference lies in the states with the assigned delay value. Given any $y \in Y$, any pair of two states $(\hat{x}, u), (\hat{x}', u') \in y$ satisfies that there exist $s, s' \in \mathcal{L}(G) : \hat{x} \in \hat{\alpha}(s), \hat{x}' \in \hat{\alpha}(s')$ such that $P_{E_o}(s) = P_{E_o}(s')$. For each $s \in \mathcal{L}(G) : \hat{\alpha}(s) \subseteq \hat{X}^F$, $\mathcal{O}_{E_o}(\hat{G})$ records the delay of the events in E_o that help to distinguish s from the system trajectories $s' \in \mathcal{L}(G) : \hat{\alpha}(s') \cap \hat{X}^H \neq \emptyset$. This is needed to verify K^T-diagnosability and T-constrained prognosability, as it will be clear in the next section.

Example 4: (Building the delay observer). For the system G in Fig. 1(b), we consider $D = 3$, $K = 3$ and $T = 2$, and build the corresponding fault automaton $\hat{G}$ as shown in Fig. 2(a). Given $\lceil \frac{D}{T} \rceil = 2$, $E_o = \{e_1, e_2, e_3\}$ and $\hat{X}^F, \hat{X}^H$ in (12), we run Algorithm 1 to obtain the delay observer $\mathcal{O}_{E_o}^{\text{diag}}(\hat{G})$ shown in Fig. 2(b), where one can see that only the delay of $\mathbf{e_1}$ (denoted with bold) in $e_3uf\mathbf{e_1}e_2e_2$ is recorded. The delay value "0" in $(6, 3, 0)$ indicates that the occurrence of $\mathbf{e_1}$ that helps to distinguish $e_3uf\mathbf{e_1}e_2e_2$ from other system trajectories $s' \in \mathcal{L}(G) : \hat{\alpha}(s') \cap \hat{X}^H \neq \emptyset$ has been received, which implies that the fault can be diagnosed, as shown in Example 2. On the other hand, given $\hat{X}^F, \hat{X}^H$ in (13), we obtain $\mathcal{O}_{E_o}^{\text{diag}}(\hat{G})$ shown in Fig. 2(c) with Algorithm 1, where the delay of $\mathbf{e_3}$ in $\mathbf{e_3}ufe_1e_2e_2$ is recorded. □

B. Verification of K^T-diagnosability and T-constrained prognosability

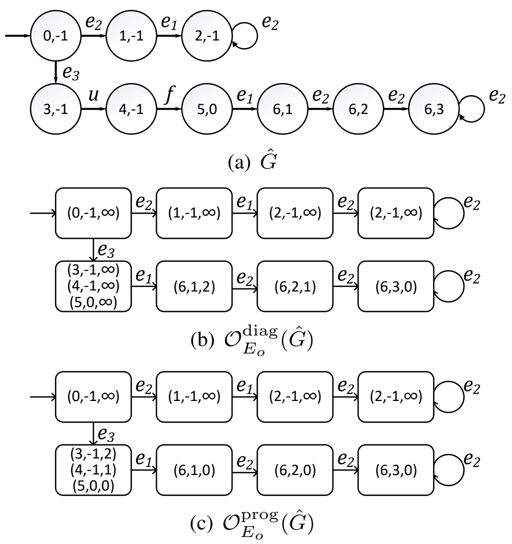
Using all structures above, we initiate the verification process of K^T-diagnosability and T-constrained prognosability.

In Section IV-A, we embed all delays of necessary events into the delay observer, in which all states will be checked by the following verification function. Define the mapping $\psi : y \to \{H, F\}$ as follows: $\forall y \in Y$,

$$\psi(y) = \begin{cases} F, & \text{if } y \in \mathcal{VC}; \\ H, & \text{otherwise}; \end{cases} \tag{15}$$

Algorithm 1 Construction of the delay-observer $\mathcal{O}_{E_o}(\hat{G})$

```
Input: $\hat{G} = (\hat{X}, E, \hat{\alpha}, \hat{X}_0), \lceil \frac{D}{T} \rceil, E_o, \hat{X}^F, \hat{X}^H$;
Output: $\mathcal{O}_{E_o}(\hat{G}) = (Y, E_o, \beta, y_0)$;
 1: $y_0 \leftarrow \emptyset$;
 2: for $\hat{x}_0 \in \hat{X}_0$ do
 3:     $y_0 \leftarrow y_0 \cup (\hat{x}_0, \infty)$; RECORD1$(\hat{x}_0, \infty, y_0)$;
 4: end for
 5: $Y \leftarrow \{y_0\}$; RECORD2$(y_0)$; Return $\mathcal{O}_{E_o}(\hat{G})$;
 6: procedure RECORD1$(\hat{x}, u, y)$
 7:     for $e \in E_{uo} : \hat{\alpha}(\hat{x}, e)!$ do
 8:         for $\hat{x}' \in \hat{\alpha}(\hat{x}, e)$ do
 9:             if $0 < u < \infty$ then
10:                 $u' \leftarrow u - 1$;
11:             else
12:                 $u' \leftarrow u$;
13:             end if
14:             if $(\hat{x}', u') \notin y$ then
15:                 $y \leftarrow y \cup \{(\hat{x}', u')\}$;
16:                 RECORD1$(\hat{x}', u', y)$;
17:             end if
18:         end for
19:     end for
20: end procedure
21: procedure RECORD2$(y)$
22:     $\iota \leftarrow \{\hat{x} \in \hat{X} \mid (\hat{x}, u) \in y\}$; $E^{\text{Rec}} \leftarrow \emptyset$;
23:     for $e \in E_o$ do
24:         $\iota^* \leftarrow \{\hat{x} \in \hat{X} \mid \exists \hat{x}' \in \iota, \text{ s.t. } \hat{x} \in \hat{\alpha}(\hat{x}', e)\}$;
25:         if $\iota^* \cap \hat{X}^H = \emptyset$ then
26:             $E^{\text{Rec}} \leftarrow E^{\text{Rec}} \cup \{e\}$;
27:         end if
28:         $y^e \leftarrow \emptyset$;
29:     end for
30:     for $(\hat{x}, u) \in y$ do
31:         for $e \in E_o : \hat{\alpha}(\hat{x}, e)!$ do
32:             if $u = \infty \wedge e \in E^{\text{Rec}}$ then
33:                 $u' \leftarrow \lceil \frac{D}{T} \rceil$;
34:             else if $0 < u < \infty$ then
35:                 $u' \leftarrow u - 1$;
36:             else
37:                 $u' \leftarrow u$;
38:             end if
39:             for $\hat{x}' \in \hat{\alpha}(\hat{x}, e)$ do
40:                 if $(\hat{x}', u') \notin y^e$ then
41:                     $y^e \leftarrow y^e \cup \{(\hat{x}', u')\}$;
42:                     RECORD1$(\hat{x}', u', y^e)$;
43:                 end if
44:             end for
45:         end for
46:     end for
47:     for $e \in E_o$ do
48:         Add $\beta(y, e) = y^e$ to $\mathcal{O}_{E_o}(\hat{G})$;
49:         if $y^e \notin Y$ then
50:             $Y \leftarrow Y \cup \{y^e\}$; RECORD2$(y^e)$;
51:         end if
52:     end for
53: end procedure
```



(a) $\hat{G}$

(b) $\mathcal{O}_{E_o}^{\text{diag}}(\hat{G})$

(c) $\mathcal{O}_{E_o}^{\text{prog}}(\hat{G})$

Fig. 2: Fault automaton $\hat{G}$, and the delay observers $\mathcal{O}_{E_o}^{\text{diag}}(\hat{G})$ and $\mathcal{O}_{E_o}^{\text{prog}}(\hat{G})$ for fault diagnosis and prognosis with $\lceil \frac{D}{T} \rceil = 2$.

where the condition $y \in \mathcal{VC}$ is defined as:

$$\exists (\hat{x}, u) \in y, \text{ s.t. } \hat{x} \in \hat{X}^F, u > 0. \tag{16}$$

Now we are in the position to verify K^T-diagnosability and T-constrained prognosability with the following theorem.

Theorem 1: Let G in (1) be the system model, f be the fault events, D be the transmission distance, $\mathcal{O}_{E_o}(\hat{G})$ be the delay observer built with Algorithm 1. Then, if $\hat{X}^F$ and $\hat{X}^H$ are as in (12), $\mathcal{L}(G)$ is K^T-diagnosable w.r.t. f and D if and only if

$$\forall y \in Y, \psi(y) = H. \tag{17}$$

If $\hat{X}^F$ and $\hat{X}^H$ are as in (13), $\mathcal{L}(G)$ is T-constrained prognosable w.r.t. f and D if and only if (17) holds.

Proof: Let us only prove the K^T-diagnosability of the theorem, as the proof for T-constrained prognosability follows similar lines.

(Necessity): Seeking a contradiction argument, let us first suppose that $\mathcal{L}(G)$ is K^T-diagnosable w.r.t. f and D while $\exists y \in Y$, s.t. $\psi(y) = F$. Then, we have that y satisfies (16).

We now consider event strings $s \in \mathcal{L}(\hat{G}) : \hat{x}^k \in \hat{\alpha}(s)$ and $s' \in \bar{s} : |s| - |s'| = \lceil \frac{D}{T} \rceil$, where $\hat{x}$ is in (16). Then, we have that $s' \in \zeta_{E_o}^{\lceil \frac{D}{T} \rceil}(s)$. Since $u > 0$, we can obtain from the procedures RECORD1 and RECORD2 (cf. lines 25 and 32) in Algorithm 1 that there exists $(\hat{x}'', \infty) \in \beta(P_{E_o}(s'))$ such that $|\hat{x}''|_f = -1$; otherwise, we have that $\forall (\hat{x}', u') \in \beta(P_{E_o}(s')), |\hat{x}'|_f \geq 0$, which is impossible since the delay ∞ will never appear in such an observer state according to the procedure RECORD2. Then, we have that there exists $s'' \in \mathcal{L}(\hat{G}) : \hat{x}'' \in \hat{\alpha}(s'')$ such that $P_{E_o}(s'') = P_{E_o}(s') \in P_{E_o}(\zeta_{E_o}^{\lceil \frac{D}{T} \rceil}(s'')) \cap P_{E_o}(\zeta_{E_o}^{\lceil \frac{D}{T} \rceil}(s))$. Finally, with $\Delta(s) = K$ and $\Delta(s'') = -1$, we conclude that $\mathcal{L}(G)$ is not K^T-diagnosable w.r.t. f and D, resulting in a contradiction.

(Sufficiency): Seeking a contradiction argument, let us suppose that $\forall y \in Y, \psi(z) = H$ while $\mathcal{L}(G)$ is not K^T-diagnosable w.r.t. f and D. Then, we have that there exists $s, s' \in \mathcal{L}(G) : \Delta(s) = K, \Delta(s') = -1$ such that $P_{E_o}(\zeta_{E_o}^{\lceil \frac{D}{T} \rceil}(s)) \cap P_{E_o}(\zeta_{E_o}^{\lceil \frac{D}{T} \rceil}(s')) \neq \emptyset$.

We first consider the case that there exists $s'' \in \mathcal{L}(\hat{G}) : P_{E_o}(s'') = P_{E_o}(s)$ such that $\Delta(s'') = -1$. According to the procedures `RECORD1` and `RECORD2` (cf. lines 25 and 32) in Algorithm 1, there exists $(\hat{x}, u) \in \beta(P_{E_o}(s))$ such that $|\hat{x}|_f = K$ and $u = \infty$, implying $\psi(\beta(P_{E_o}(s))) = F$, resulting in a contradiction.

We then consider the case that $\forall s'' \in \mathcal{L}(\hat{G}) : P_{E_o}(s'') = P_{E_o}(s), \Delta(s'') \neq -1$. In this case, there must exist $s^k \in \bar{s}, e \in E_o : s^k e \in \bar{s}$ such that $\exists s^{-1} \in \bar{s}'$, s.t. $P_{E_o}(s^{-1}) = P_{E_o}(s^k) \in P_{E_o}(\zeta_{E_o}^{\lceil \frac{D}{T} \rceil}(s')) \cap P_{E_o}(\zeta_{E_o}^{\lceil \frac{D}{T} \rceil}(s))$. Note that $\Delta(s^{-1}) = -1$. From the definition of ζ, we can obtain that $s^k \in \zeta_{E_o}^{\lceil \frac{D}{T} \rceil}(s)$, which implies that $|s| - |s^k e| < \lceil \frac{D}{T} \rceil$ since $e \in \omega(s^k)$ and $|s| - |s^k e| \geq \lceil \frac{D}{T} \rceil$ would lead to $s^k \notin \zeta_{E_o}^{\lceil \frac{D}{T} \rceil}(s)$. According to the procedures `RECORD1` and `RECORD2` (cf. lines 25 and 32) in Algorithm 1, there exists $(\hat{x}_1, u_1) \in \beta(P_{E_o}(s^k e))$ such that $u_1 = \infty$ or $\lceil \frac{D}{T} \rceil$. Since $|s| - |s^k e| < \lceil \frac{D}{T} \rceil$, there exists $(\hat{x}_2, u_2) \in \beta(P_{E_o}(s))$ such that $|\hat{x}_2|_f = K$ and $u_2 > u_1 - \lceil \frac{D}{T} \rceil \geq 0$, implying $\psi(\beta(P_{E_o}(s))) = F$, resulting in a contradiction, which completes the proof. ■

Remark 3: (Complexity analysis). The computational complexity of the proposed verification methods primarily arises from the construction of the delay observer, which has a complexity similar to that of the observer [1] in the literature. Indeed, the only difference between these two observers lies in the addition of an extra delay value in the states.

Example 5: (Verifying K^T-diagnosability and T-constrained prognosability). Consider the delay observer in Fig. 2(b) and 2(c). We use (15) to check the states in $\mathcal{O}_{E_o}^{\text{diag}}(\hat{G})$ and $\mathcal{O}_{E_o}^{\text{prog}}(\hat{G})$. We know that in diagnosis problem, $\hat{X}^F = \{\hat{x} \in \hat{X} \mid |\hat{x}|_f = 3\}$ while in prognosis problem, $\hat{X}^F = \{\hat{x} \in \hat{X} \mid |\hat{x}|_f = 0\}$. We have $\psi(\{(6, 3, 0)\}) = H$ in $\mathcal{O}_{E_o}^{\text{diag}}(\hat{G})$ and $\psi(\{(3, -1, 2), (4, -1, 1), (5, 0, 0)\}) = H$ in $\mathcal{O}_{E_o}^{\text{prog}}(\hat{G})$, indicating that $\mathcal{L}(G)$ is K^T-diagnosable and T-constrained prognosable w.r.t. f and D for $K = 3$, $T = 2$, $D = 3$ as shown in Example 2 and 3. □

V. Conclusion

This work extended and solved fault diagnosis and prognosis in discrete event systems in the delayed observation scenario. In order to make this extension possible, we proposed a new diagnosability notion, namely K^T-diagnosability, as well as a new prognosability notion, namely T-constrained prognosability, extended from a restated prognosability definition to avoid conservativeness. To handle the delayed observations, a delay observer was constructed, that allows to verify the aforementioned conditions of diagnosability and prognosability. Interesting directions for future work are to consider multiple delays, dynamic observations and stochastic settings.

References

[1] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*. Springer Science & Business Media, 2009.

[2] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 40, no. 9, pp. 1555–1575, 1995.

[3] S. Genc and S. Lafortune, "Predictability of event occurrences in partially-observed discrete-event systems," *Automatica*, vol. 45, no. 2, pp. 301–311, 2009.

[4] F. Lin, "Diagnosability of discrete event systems and its applications," *Discrete Event Dynamic Systems*, vol. 4, no. 2, pp. 197–212, 1994.

[5] J. He, D. Wang, M. Yang, and Y. Hu, "Asynchronous fault diagnosis of stochastic discrete-event systems in industrial applications," *IEEE Sensors Journal*, 2024.

[6] R. Debouk, S. Lafortune, and D. Teneketzis, "Coordinated decentralized protocols for failure diagnosis of discrete event systems," *Discrete Event Dynamic Systems*, vol. 10, no. 1-2, pp. 33–86, 2000.

[7] R. Kumar and S. Takai, "Decentralized prognosis of failures in discrete event systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 48–59, 2009.

[8] X. Yin, J. Chen, Z. Li, and S. Li, "Robust fault diagnosis of stochastic discrete event systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 4237–4244, 2019.

[9] H. Liao, F. Liu, and N. Wu, "Robust predictability of stochastic discrete-event systems and a polynomial-time verification," *Automatica*, vol. 144, p. 110477, 2022.

[10] F. Liu and D. Qiu, "Diagnosability of fuzzy discrete-event systems: A fuzzy approach," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 2, pp. 372–384, 2009.

[11] B. Benmessahel, M. Touahria, and F. Nouioua, "Predictability of fuzzy discrete event systems," *Discrete Event Dynamic Systems*, vol. 27, pp. 641–673, 2017.

[12] A. Paoli, M. Sartini, and S. Lafortune, "Active fault tolerant control of discrete event systems using online diagnostics," *Automatica*, vol. 47, no. 4, pp. 639–649, 2011.

[13] A. T. Watanabe, A. B. Leal, J. E. Cury, and M. H. de Queiroz, "Safe controllability using online prognosis," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 12 359–12 365, 2017.

[14] S. Lafortune, F. Lin, and C. N. Hadjicostis, "On the history of diagnosability and opacity in discrete event systems," *Annual Reviews in Control*, vol. 45, pp. 257–266, 2018.

[15] A. T. Watanabe, R. Sebem, A. B. Leal, and M. d. S. Hounsell, "Fault prognosis of discrete event systems: An overview," *Annual Reviews in Control*, vol. 51, pp. 100–110, 2021.

[16] C. E. Nunes, M. V. Moreira, M. V. Alves, L. K. Carvalho, and J. C. Basilio, "Codiagnosability of networked discrete event systems subject to communication delays and intermittent loss of observation," *Discrete Event Dynamic Systems*, vol. 28, no. 2, pp. 215–246, 2018.

[17] Z. Liu, X. Yin, S. Shu, F. Lin, and S. Li, "Online supervisory control of networked discrete event systems with control delays," *IEEE Transactions on Automatic Control*, vol. 67, no. 5, pp. 2314–2329, 2021.

[18] G. S. Viana, M. V. Alves, and J. C. Basilio, "Codiagnosability of networked discrete event systems with timing structure," *IEEE Transactions on Automatic Control*, vol. 67, no. 8, pp. 3933–3948, 2021.

[19] W. Qiu and R. Kumar, "Distributed diagnosis under bounded-delay communication of immediately forwarded local observations," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 38, no. 3, pp. 628–643, 2008.

[20] J. Wang, S. Baldi, W. Yu, and X. Yin, "Distributed fault diagnosis in discrete event systems with transmission delay impairments," *IEEE Transactions on Automatic Control*, 2024.

[21] S. Takai and R. Kumar, "Distributed failure prognosis of discrete event systems with bounded-delay communications," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1259–1265, 2011.

[22] F. Cassez and S. Tripakis, "Fault diagnosis with static and dynamic observers," *Fundamenta Informaticae*, vol. 88, no. 4, pp. 497–540, 2008.

[23] X. Yin and S. Lafortune, "Codiagnosability and coobservability under dynamic observations: Transformation and verification," *Automatica*, vol. 61, pp. 241–252, 2015.