Distributed Diagnosis Under Bounded-Delay Communication of Immediately Forwarded Local Observations

Wenbin Qiu and Ratnesh Kumar (wqiu, rkumar@iastate.edu)
Dept. of Elec. & Comp. Eng., Iowa State University, Ames, IA 50011

Abstract— **In this paper, we study *distributed* failure diagnosis under k-bounded communication delay, where each local site transmits its observations to other sites immediately after each observation, and which is received within at most k more event executions of the plant. This work extends our prior work on *decentralized* failure diagnosis [16] that did not allow any communication among the local sites. A notion of *joint$_k^{iop}$-diagnosability* is introduced so that any failure can be diagnosed within a bounded delay of its occurrence by one of the local sites using its own observations and the k-bounded delayed observations received from other local sites. The local sites communicate among each other using an "immediate observation passing (iop)" protocol, forwarding any observation immediately up on its occurrence. We construct models for k-bounded communication delay, and use them to extend the system and non-fault specification models for capturing the effect of bounded-delay communication. Using the extended system and specification models, the distributed diagnosis problem under the immediate observation passing protocol is then converted to a decentralized diagnosis problem. Results from [16] are applied for verifying joint$_k^{iop}$-diagnosability, and for synthesizing local diagnosers. Methods by which complexity of testing joint$_k^{iop}$-diagnosability and of on-line diagnosis can be reduced are presented.**

Keywords: Discrete event systems, distributed failure diagnosis, joint-diagnosability, communication delay

I. Introduction

Failure diagnosis is an active area of research, and has received considerable attention in the literature. A failure is a deviation from an expected or desired behavior. Various approaches have been proposed for failure diagnosis, including fault-trees, expert systems, neural networks, fuzzy logic, bayesian networks, and analytical redundancy [15]. These are broadly categorized into non-model based (where observed behavior is matched to known failures), and model based (where observed behavior is compared against model predictions for any abnormality). For discrete event systems (DES) a certain model based approach for failure diagnosis was proposed in [18], and extended in [17], [9], [10], [8], [5], [21]. The application of DES failure diagnosis includes heating, ventilation, and air conditioning systems [19], transportation systems [12], [6], communication networks [2], [1], [13], manufacturing systems [3], [14], digital circuits [11], and power systems [7].

The problem of *decentralized* diagnosis of discrete event systems was first considered as one special case of *distributed* diagnosis in [5], where local diagnosers do not directly communicate with each other, and there is no delay in the communication channels between the coordinator and the local diagnosers. In that paper, "lack of fully ambiguous traces" was stated as a sufficient condition for decentralized diagnosis to be equivalent to that of centralized one, and an algorithm was presented for verifying the "lack of fully ambiguous traces". The algorithm is based upon structural properties of global (centralized) and local (decentralized) diagnosers, and has an exponential complexity in the size of the system owing to the exponential size of the diagnosers. The one-step out-of-order effect of communication delays was explored in [4].

In a previous work [16], we studied distributed diagnosis involving no communication among local diagnosers. A notion of *codiagnosability* was introduced to capture the fact that the occurrence of any failure must be diagnosed within bounded delay by at least one local diagnoser using its own observations of the system execution. Polynomial algorithms were provided for (i) testing codiagnosability, (ii) computing the delay bound of diagnosis, (iii) off-line synthesis of diagnosers, and (iv) on-line diagnosis using them.

In this paper, we study the distributed failure diagnosis problem under k-bounded communication delay. To formulate the way information is exchanged among local sites, we first present a dynamic system model of a general communication protocol. We then restrict our attention to a specific protocol, the *immediate observation passing (iop)* protocol, where each local site transmits its observations to other sites immediately after each observation, and the transmitted observation is received within at most k more event executions of the plant. The communication channel is assumed to be lossless and order-preserved, but incurs a bounded delay. A similar setting has been considered for distributed *control* in the work of Tripakis [20].

A notion of *joint$_k$-diagnosability* is introduced so that any failure can be diagnosed within a bounded delay of its occurrence by one of the local sites using its own observations and the delayed observations received from other local sites communicating among each other using a general protocol. The joint$_k$-diagnosability under the immediate observation passing protocol is denoted *joint$_k^{iop}$-diagnosability*. We construct models for k-bounded communication delay, and use them to extend the system and non-fault specification models for capturing the effect of bounded-delay communication. Using the extended system

The research was supported in part by the National Science Foundation under the grants NSF-ECS-0099851, NSF-ECS-0218207, NSF-ECS-0244732, NSF-EPNES-0323379, and NSF-ECS-0424048, and a DoD-EPSCoR grant through the Office of Naval Research under the grant N000140110621.

0-7803-9098-9/05/$25.00 ©2005 AACC

and specification models, the distributed diagnosis problem under the immediate observation passing protocol is then converted to a decentralized diagnosis problem of [16]. Results from [16] are applied for verifying joint$_k^{iop}$-diagnosability, and synthesizing local diagnosers. A way by which complexity of testing joint$_k^{iop}$-diagnosability and of on-line diagnosis can be reduced is presented.

II. Notation and Preliminaries

Given an event set Σ, Σ^* denotes the set of all finite length event seqences over Σ, including the zero length event sequence ε. A member of Σ^* is a *trace* and a subset of Σ^* is a *language*. Given a language $L \subseteq \Sigma^*$, it is said to be *prefix-closed* if $L = pr(L)$, where $pr(L) := \{s \in \Sigma^* | \exists t \in \Sigma^* \text{ s.t. } st \in L\}$. A DES is modeled as a finite automaton $G = (X, \Sigma, \alpha, x_0)$, where X is the set of states, Σ is the finite set of events, $x_0 \in X$ is the *initial state*, and $\alpha : X \times \overline{\Sigma} \to 2^X$ is the *transition function* with $\overline{\Sigma} := \Sigma \cup \{\varepsilon\}$. G is said to be *deterministic* if $|\alpha(\cdot, \cdot)| \leq 1$ and $|\alpha(\cdot, \epsilon)| = 0$; otherwise, it is called *nondeterministic*. The *generated language* of G is given by, $L(G) := \{s \in \Sigma^* | \alpha(x_0, s) \neq \emptyset\}$. A *path* in G is a sequence of transitions $(x_1, \sigma_1, x_2, \cdots, \sigma_{n-1}, x_n)$, where $\sigma_i \in \overline{\Sigma}$ and $x_{i+1} \in \alpha(x_i, \sigma_i)$ for all $i \in \{1, \cdots, n-1\}$. Such a path is called a *cycle* if $x_1 = x_n$.

Given two automata $G_1 = (X_1, \Sigma_1, \alpha_1, x_{0,1})$ and $G_2 = (X_2, \Sigma_2, \alpha_2, x_{0,2})$, the *synchronous composition* of G_1 and G_2 is defined as, $G_1 \| G_2 = (X_1 \times X_2, \Sigma_1 \cup \Sigma_2, \alpha, (x_{0,1}, x_{0,2}))$, where α is defined as follows: $\forall (x_1, x_2) \in X_1 \times X_2, \sigma \in \Sigma_1 \cup \Sigma_2, \alpha((x_1, x_2), \sigma) :=$

$$\begin{cases} \alpha_1(x_1, \sigma) \times \alpha_2(x_2, \sigma) & \text{if } \sigma \in \Sigma_1 \cap \Sigma_2; \\ \alpha_1(x_1, \sigma) \times \{x_2\} & \text{if } \sigma \in \Sigma_1 - \Sigma_2; \\ \{x_1\} \times \alpha_2(x_2, \sigma) & \text{if } \sigma \in \Sigma_2 - \Sigma_1. \end{cases}$$

Given a state $x \in X$, the *ε-closure* of x, denoted $\varepsilon_G^*(x) \subseteq X$, includes all state that can be reached from state x by zero or more ε transitions, and is recursively defined as: $x \in \varepsilon_G^*(x)$; $x' \in \varepsilon_G^*(x) \Rightarrow \alpha(x', \varepsilon) \subseteq \varepsilon_G^*(x)$. The domain of the state transition function α can be extended from $X \times \overline{\Sigma}$ to $X \times \Sigma^*$ recursively as follows: $\forall x \in X, s \in \Sigma^*, \sigma \in \Sigma, \alpha(x, \varepsilon) = \varepsilon_G^*(x)$; $\alpha(x, s\sigma) = \varepsilon_G^*(\alpha(\alpha(x, s), \sigma))$.

When the system execution is observed by a global observer, we can define a *global observation mask*, $M : \Sigma \to \overline{\Lambda}$ with $M(\varepsilon) = \varepsilon$, where $\overline{\Lambda} := \Lambda \cup \{\varepsilon\}$ and Λ is the set of *observed symbols*. The definition of M can be extended from events to event sequences inductively as follows: $M(\varepsilon) = \varepsilon$; $\forall s \in \Sigma^*, \sigma \in \Sigma, M(s\sigma) = M(s)M(\sigma)$. Given an automaton G and mask M, $M(G)$ is the automaton G with each transition (x, σ, x') of G replaced by $(x, M(\sigma), x')$. The *local observation mask* for a site i is defined as $M_i : \Sigma \to \overline{\Lambda}_i$ $(i \in I = \{1, \cdots, m\})$ with $M_i(\epsilon) = \epsilon$, where m is the number of local observers, $\overline{\Lambda}_i := \Lambda_i \cup \{\varepsilon\}$ and Λ_i is the set of locally observed symbols.

Given a *specification* model $R = (Y, \Sigma, \beta, y_0)$, the *completed specification model* $\overline{R}$ is defined as $\overline{R} := (\overline{Y}, \Sigma, \overline{\beta}, y_0)$, where $\overline{Y} := Y \cup \{F\}$, and $\overline{\beta}$ is defined as: $\forall \overline{y} \in \overline{Y}, \sigma \in \Sigma$,

$$\overline{\beta}(\overline{y}, \sigma) := \begin{cases} \beta(\overline{y}, \sigma), & \text{if } [\overline{y} \in Y] \wedge [\beta(\overline{y}, \sigma) \neq \emptyset], \\ F, & \text{if } [\overline{y} = F] \vee [\beta(\overline{y}, \sigma) = \emptyset]. \end{cases}$$

Definition 1: [16] Let L be the prefix-closed language generated by a system and K be a prefix-closed specification language with $K \subseteq L$. Assume there are m local sites $(I = \{1, \cdots, m\})$. (L, K) is said to be *codiagnosable* with respect to $\{M_i\}$ if

$$(\exists n \in \mathcal{N})(\forall s \in L - K)(\forall st \in L, |t| \geq n \text{ or } st \text{ deadlocks})$$

$$\Rightarrow (\exists i \in I)(\forall u \in L, M_i(u) = M_i(st) \Rightarrow u \in L - K).$$

III. Communication Protocols

Figure 1 shows the architecture of a distributed failure diagnosis system with two local sites. Site i contains three modules: observation mask M_i, communication protocol i, and diagnoser i. The protocol module for site i decides how to share information among various diagnosers. The diagnoser module for site i performs failure diagnosis based on the local observations and the communicated information received from other sites j.

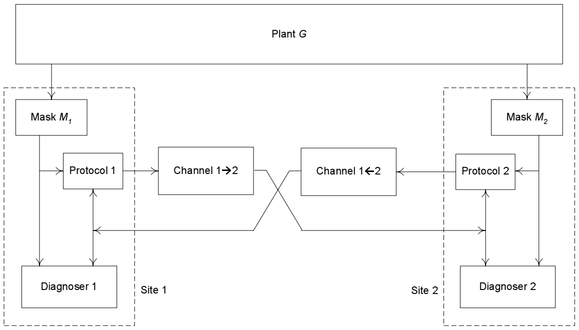


Fig. 1. Architecture of a *distributed* failure diagnosis system

A communication protocol is a causal (prefix-preserving) map from history of all information received to history of all information transmitted. The communication protocol at site i can be implemented as a dynamical system as shown in Figure 2.

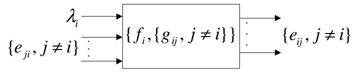


Fig. 2. Model of a general communication protocol

The inputs to this dynamical system consist of the *local observations* $\lambda_i \in \Lambda_i$, and the *communicated informations* from other diagnosers $\{e_{ji} | j \neq i\}$. The output of this dynamical system is the *information to be transmitted* to other diagnosers $\{e_{ij} | j \neq i\}$. The protocol maintains an internal state e_i, called the *protocol state*. The formats of e_i, e_{ji}, and e_{ij} $(j \neq i)$ are specific to a protocol. Formally, a general communication protocol is given by

$P^{gen} := \{P_i^{gen}, i \in I\}$, where each P_i^{gen} is modeled by a set of maps $\{f_i, \{g_{ij}, j \neq i\}\}$ as follows:

$$P_i^{gen} : \begin{cases} e_i = f_i(e_i, \lambda_i, \{e_{ji}, j \neq i\}) \\ e_{ij} = g_{ij}(e_i, \lambda_i, \{e_{ji}, j \neq i\}) \end{cases} \quad (1)$$

f_i is the *protocol state update map* at site i, which updates the protocol state based on its current value and newly received information, λ_i or $\{e_{ji}, j \neq i\}$. (Events in $\{\lambda_i\} \cup \{e_{ji}, j \neq i\} \cup \{e_{ij}, j \neq i\}$ occur asynchronously.) g_{ij} is the *protocol-output map* at site i, which determines the information to be transmitted to site j ($j \neq i$). The set of protocols of the form specified in (1) is denoted $\mathcal{P}^{gen}$. The setting of decentralized diagnosis involving no communication can be represented by a "null-communication" protocol, $P^{\emptyset}$, for which the output is always null.

If a protocol allows diagnosers to transmit only their local observations, it is called an *observation passing protocol*, denoted $P^{op} := \{P_i^{op}, i \in I\}$. The dynamic model of P_i^{op} is captured by a set of maps $\{f_i, \{g_{ij}, j \neq i\}\}$ defined as follows. ("$\backslash$" denotes the "after" operation.)

$$P_i^{op} : \begin{cases} e_i = e_i\lambda_i \backslash \{e_{ij}, j \neq i\} \\ e_{ij} = g_{ij}(e_i, \lambda_i) \leq e_i\lambda_i \end{cases} \quad (2)$$

where the protocol state e_i is a "vector" whose j^{th} entry stores observations that are not yet transmitted to site j, and e_{ij} is the newly transmitted observation to site j, which is a prefix of $e_i\lambda_i$, the concatenation of the observation trace not yet transmitted and the newly arrived observation. The class of protocols of the form specified in (2) comprise the class of observation passing protocols, denoted $\mathcal{P}^{op}$.

When $g_{ij}(e_i, \lambda_i) = e_i\lambda_i = \epsilon\lambda_i = \lambda_i$ in (2), the protocol model simplifies to the one given in (3), and the corresponding protocol is called the *immediate observation passing protocol*, denoted $P^{iop} := \{P_i^{iop}, i \in I\}$, where P_i^{iop} is defined as follows:

$$P_i^{iop} : \begin{cases} e_i = \epsilon \\ e_{ij} = \lambda_i \end{cases} \quad (3)$$

In this protocol any local observation is transmitted immediately to other sites, and there is no observation that is not transmitted but stored as the protocol state. Therefore, the protocol state update map f_i is trivial, and the information to be transmitted e_{ij} simply equals the local observation λ_i in the protocol output map g_{ij}. The distributed diagnosis architecture under the immediate observation passing protocol is shown in Figure 3, where the operations of masking and delaying are interchanged without affecting the diagnosis result.

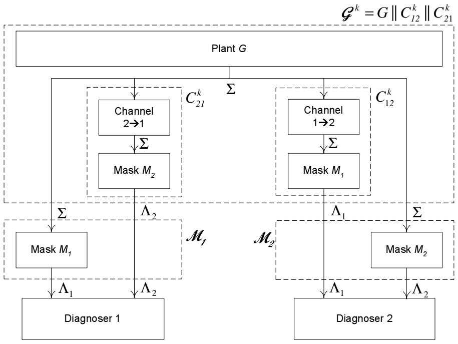


Fig. 3. Distributed diagnosis architecture under protocol P^{iop}

IV. Communication and Extended Models

It is clear from Figure 3 that P^{iop}-based distributed diagnosis can be converted to a decentralized diagnosis having an extended plant $\mathcal{G}^k$, and local diagnosers having the extended observation masks $\{\mathcal{M}_i\}$. The extended plant is given by $\mathcal{G}^k = G\|C^k_{12}\|C^k_{21}$, where C^k_{ij} models the k-bounded delaying and masking operation. We call C^k_{ij} to be the *k-delaying&masking* model. The *extended plant* $\mathcal{G}^k$ "generates" events in $\Sigma \cup \Lambda_1 \cup \Lambda_2$, which are observed by diagnoser i through an extended observation mask $\mathcal{M}_i : \Sigma \cup \Lambda_1 \cup \Lambda_2 \to \overline{\Lambda}_1 \cup \overline{\Lambda}_2$ defined as follows:

$$\mathcal{M}_i(\sigma) := \begin{cases} M_i(\sigma), & \sigma \in \Sigma; \\ \sigma, & \sigma \in \Lambda_j \ (j \neq i); \\ \epsilon, & \sigma \in \Lambda_i. \end{cases} \quad (4)$$

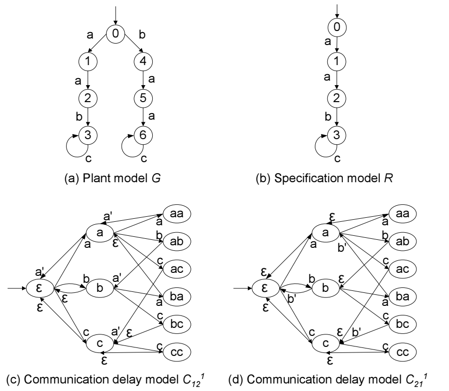
The k-delaying&masking model from diagnoser i to diagnoser j ($i \neq j$) is defined as $C^k_{ij} = (Z^k_{ij}, \Sigma \cup \Lambda_i, \gamma^k_{ij}, z_0)$. Z^k_{ij} is the set of states, which are represented by the event traces executed in the plant but their observed values not yet received by the destination diagnoser. $\Sigma \cup \Lambda_i$ is the event set, where Σ is the set of input events and Λ_i is the set of output events. Without loss of generality, we assume that $\Sigma \cap \Lambda_i = \emptyset$ and $\Lambda_i \cap \Lambda_j = \emptyset$ $(i \neq j)$ (otherwise, we can simply rename some of the symbols). $z_0 = \epsilon$ is the initial state. The transition function γ^k_{ij} is defined as follows.

- "Arrival" due to an event execution in the plant: $\forall z \in Z^k_{ij}, \forall \sigma \in \Sigma$, if $|z| \leq k$, then $\gamma^k_{ij}(z, \sigma) = z\sigma$,
- "Departure" due to a reception at the destination diagnoser: $\forall z \in Z^k_{ij}, \forall \lambda_i \in \Lambda_i$, if $M_i(head(z)) = \lambda_i$, then $\gamma^k_{ij}(z, \lambda_i) = z \backslash head(z)$,
- Undefined, otherwise,

where $head(z)$ is the first event in trace z, and the after operator "$\backslash$" in $z \backslash head(z)$ returns the trace after removing the initial event $head(z)$ from the trace z.

Having introduced the model C^k_{ij}, we next introduce several "extended" models as follows: *extended plant model*: $\mathcal{G}^k = G\|C^k_{12}\|C^k_{21}$; *extended specification model*: $\mathcal{R}^k = R\|C^k_{12}\|C^k_{21}$; *refined extended specification model*: $\overline{\mathcal{R}}^k = \overline{R}\|C^k_{12}\|C^k_{21}$; *refined extended plant model*: $\overline{\mathcal{G}}^k = \mathcal{G}^k\|\overline{\mathcal{R}}^k = G\|\overline{R}\|C^k_{12}\|C^k_{21}$; *extended local specification model*: $\mathcal{R}^k_i = R\|C^k_{ji}$; *refined extended local specification model*: $\overline{\mathcal{R}}^k_i = \overline{R}\|C^k_{ji}$. Note that in the above construction, any event in the set Σ is synchronized among all participating components, while any event in the set $\Lambda_1 \cup \Lambda_2$ is executed asynchronously.

Example 1: A plant model G and a specification model R are shown in Figure 4 (a) and (b), respectively. Suppose the observation masks of two local sites are defined as follows: $M_1(a) = a'$, $M_1(b) = M_1(c) = \epsilon$; and $M_2(b) = b'$, $M_2(a) = M_2(c) = \epsilon$. For delay = 1, Figure 4 (c) and (d) show the models C_{12}^1 and C_{21}^1 respectively. If we follow the trace aba' in the first model C_{12}^1, the states ϵ, a, ab and b are traversed sequentially. This corresponds to the situation in which site 1 sends out its observation a' to site 2 after the occurrence of ab in the plant, whereas event b is executed in the plant but its observation is not yet received at site 2. The refined extended plant model $\overline{\mathcal{G}}^1 = G\|\overline{R}\|C_{12}^1\|C_{21}^1$ and the extended specification model $\mathcal{R}^1 = R\|C_{12}^1\|C_{21}^1$ as shown in Figure 5, and the extended local specification models $\mathcal{R}_1^1 = R\|C_{21}^1$ and $\mathcal{R}_2^1 = R\|C_{12}^1$ are shown in 6.


(a) Plant model G

(b) Specification model R

(c) Communication delay model $C_{12}{}^1$

(d) Communication delay model $C_{21}{}^1$

Fig. 4. System models and communication delay models

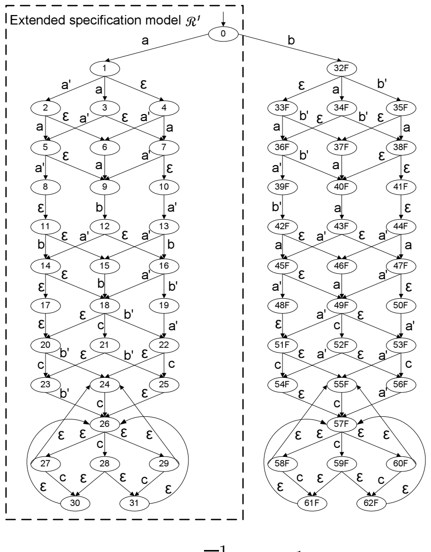


Fig. 5. $\overline{\mathcal{G}}^1$ and $\mathcal{R}^1$

Define a projection $\Pi_\Sigma : \Sigma \cup \Lambda_1 \cup \Lambda_2 \to \Sigma$ as follows:

$$\forall \sigma \in \Sigma \cup \Lambda_1 \cup \Lambda_2, \Pi_\Sigma(\sigma) := \begin{cases} \sigma, & \sigma \in \Sigma; \\ \epsilon, & \sigma \in \Lambda_1 \cup \Lambda_2. \end{cases} \quad (5)$$

This projection can be inductively extended from event to event traces: $\Pi_\Sigma(\epsilon) = \epsilon$; $\forall s \in (\Sigma \cup \Lambda_1 \cup \Lambda_2)^*, \sigma \in \Sigma \cup \Lambda_1 \cup \Lambda_2, \Pi_\Sigma(s\sigma) = \Pi_\Sigma(s)\Pi_\Sigma(\sigma)$. The inverse projection of $\Pi_\Sigma^{-1} : \Sigma^* \to (\Sigma \cup \Lambda_1 \cup \Lambda_2)^*$, is defined as follows:

$$\forall u \in \Sigma^*, \Pi_\Sigma^{-1}(u) := \{s \in (\Sigma \cup \Lambda_1 \cup \Lambda_2)^* \mid \Pi_\Sigma(s) = u\}.$$

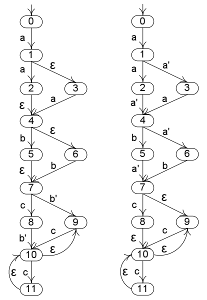


Fig. 6. $\mathcal{R}_1^1$ (left) and $\mathcal{R}_2^1$ (right))

V. JOINT$_k$-DIAGNOSABILITY

Let E_{ij} denote the set of output symbols communicated from site i to site j. Then any symbol received at site i lies in the set $\Lambda_i \cup_{j \neq i} E_{ji}$, called the set of *aggregate observation symbols* at site i. Assuming local sites commute over loss-free, order-preserved, and k-bounded delay channels, the execution of a trace s by the system results in the reception at site i of a sequence of observation symbols in Λ_i interleaved with a sequence of communication symbols in $\cup_{j \neq i} E_{ji}$. Due to the asynchronous nature of the communication channels and the introduction of bounded but random delays by them, execution of a trace s by the system can result in the reception of one of many possible sequences of observed and communicated symbols at site i. Also, any such sequence of observed and communicated symbols arrives in its entirety at site i within a bounded-delay of the execution of trace s.

To characterize the set of sequences of aggregate observations received at site i under protocol $P^{gen} \in \mathcal{P}^{gen}$ immediately at the time the system has executed a trace s, we define a map $O_i^{gen,k} : \Sigma^* \to 2^{(\Lambda_i \cup_{j \neq i} E_{ji})^*}$, where k is the communication delay bound. We call this map to be the *P^{gen}-based aggregate observations map* for site i under k-bounded communication delay. Similarly, we can define $O_i^{op,k}$ (resp., $O_i^{iop,k}$ and $O_i^{\emptyset}$) to be the *P^{op}(resp., P^{iop} and $P^{\emptyset}$)-based aggregate observations map* for site i under k-bounded communication delay. Since $P^{\emptyset}$ represents the "null communication" protocol, it is obvious that $O_i^{\emptyset}(s) = \{M_i(s)\}$ for any $s \in L(G)$. For protocol P^{iop}, the aggregate observations map $O_i^{iop,k}$ can be formally defined through the extended observation mask $\mathcal{M}_i$ as follows.

Definition 2: The *P^{iop}-based aggregate observations map* for site i under k-bounded communication delay, $O_i^{iop,k} : \Sigma^* \to 2^{(\cup_{j \in I} \Lambda_j)^*}$, is defined as follows: $\forall s \in L(G), O_i^{iop,k}(s) := \mathcal{M}_i(\Pi_\Sigma^{-1}(s) \cap L(\mathcal{G}^k))$.

Definition 3: The *P^{gen}-based indistinguishability predicate* for site i under k-bounded communication delay is defined as follows: $\forall s, t \in L(G)$,

$$\Upsilon_i^{gen,k}(s,t) := \begin{cases} 1 & O_i^{gen,k}(s) \cap O_i^{gen,k}(t) \neq \emptyset \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

The *P^{op} (resp., P^{iop} and $P^{\emptyset}$)-based indistinguishability predicate* for site i under k-bounded communication delay, denoted $\Upsilon_i^{op,k}$ (resp., $\Upsilon_i^{iop,k}$ and $\Upsilon_i^{\emptyset}$), are defined similarly by replacing $O_i^{gen,k}$ with $O_i^{op,k}$ (resp., $O_i^{iop,k}$ and $O_i^{\emptyset}$).

Proposition 1: Let $\Upsilon_i^{iop,k}$ (resp., $\Upsilon_i^{\emptyset}$) be the P^{iop}(resp., $P^{\emptyset}$)-based indistinguishability predicates for site i under k-bounded communication delay.

- $\forall s,t \in L(G)$: $[\Upsilon_i^{\emptyset}(s,t)=1] \Leftrightarrow [M_i(s)=M_i(t)]$, and
- $\forall s,t \in L(G)$: $[\Upsilon_i^{iop,k}(s,t)=1] \Leftrightarrow \exists s' \in \Pi_\Sigma^{-1}(s) \cap L(\mathcal{G}^k), t' \in \Pi_\Sigma^{-1}(t) \cap L(\mathcal{G}^k) : \mathcal{M}_i(s') = \mathcal{M}_i(t')$.

Definition 4: Let L be the prefix-closed language generated by a plant, and K be a prefix-closed specification language ($K \subseteq L$). (L,K) is said to be joint$_k$-diagnosable under P^{gen}, called *joint$_k^{gen}$-diagnosable*, if

$$(\exists n \in \mathcal{N})(\forall s \in L - K)(\forall st \in L, |t| \geq n \text{ or } st \text{ deadlocks}) \Rightarrow$$

$$(\exists i \in I)(\forall u \in L, \Upsilon_i^{gen,k}(st,u)=1 \Rightarrow u \in L-K). \quad (7)$$

The joint$_k$-diagnosability with respect to protocol $P^{op} \in \mathcal{P}^{op}$ and P^{iop}, denoted *joint$_k^{op}$-diagnosability* and *joint$_k^{iop}$-diagnosability*, respectively, are defined by replacing $\Upsilon_i^{gen,k}$ in (7) with $\Upsilon_i^{op,k}$ and $\Upsilon_i^{iop,k}$, respectively.

Theorem 1: Given a plant G and a specification model R with $L(R) \subseteq L(G)$, $(L(G), L(R))$ is joint$_k^{iop}$-diagnosable if and only if $(L(\mathcal{G}^k), L(\mathcal{R}^k))$ is codiagnosable with respect to $\{\mathcal{M}_i\}$.

An implication of Theorem 1 is that the methods presented in [16] for studying decentralized diagnosis can be applied to study P^{iop}-based distributed diagnosis. The application of codiagnosability test in [16, Algorithm 1] would result in the construction of the testing automaton $\overline{\mathcal{G}}^k \times \mathcal{R}^k \times \mathcal{R}^k$ that tracks a trace-triple satisfying $\forall i \in \{1,2\}$, $\mathcal{M}_i(s) = \mathcal{M}_i(u_i), s \in L(\overline{\mathcal{G}}^k) = L(\mathcal{G}), u_i \in L(\mathcal{R}^k)$. Since $\mathcal{M}_i(L(\mathcal{R}^k)) = \mathcal{M}_i(L(\mathcal{R}_i^k))$ as shown in the following proposition, it suffices to construct the testing automaton $\mathcal{T}^k = \overline{\mathcal{G}}^k \times \mathcal{R}_1^k \times \mathcal{R}_2^k$. The advantage is that $\mathcal{T}^k$ has a smaller state space than $\overline{\mathcal{G}}^k \times \mathcal{R}^k \times \mathcal{R}^k$.

Proposition 2: Given a specification model R, define $\mathcal{R}^k := R \| C_{12}^k \| C_{21}^k$ and $\mathcal{R}_i^k := R \| C_{ji}^k$ $(i,j \in \{1,2\}, i \neq j)$. Then, $\mathcal{M}_i(L(\mathcal{R}^k)) = \mathcal{M}_i(L(\mathcal{R}_i^k))$.

Algorithm 1: Consider a deadlock-free plant $G = (X, \Sigma, \alpha, x_0)$ and a specification model $R = (Y, \Sigma, \beta, y_0)$.

1. Construct C_{12}^k and C_{21}^k;
2. Construct the extended plant model $\mathcal{G}^k = G \| C_{21}^k \| C_{21}^k$, the refined extended plant model $\overline{\mathcal{G}}^k = G \| \overline{R} \| C_{12}^k \| C_{21}^k$, the extended specification model $\mathcal{R}^k = R \| C_{12}^k \| C_{21}^k$, and the extended local specification model $\mathcal{R}_i^k = R \| C_{ij}^k$ $(i,j \in \{1,2\}, i \neq j)$;
3. Construct a testing automaton $\mathcal{T}^k = (\mathcal{G} \| \overline{\mathcal{R}}^k) \times \mathcal{R}_1^k \times \mathcal{R}_2^k$. Note that $(\epsilon,\epsilon,\epsilon)$-transition is allowed in the testing automaton if it is not performed as a self loop. $\mathcal{T}^k$ tracks all triplet of traces $s, u_1, u_2 \in (\Sigma \cup \Lambda_1 \cup \Lambda_2)^*$ satisfying the following property: $\forall i \in \{1,2\}, \mathcal{M}_i(s) = \mathcal{M}_i(u_i), s \in L(\overline{\mathcal{G}}^k) = L(\mathcal{G}), u_i \in L(\mathcal{R}_i^k)$;
4. Check the existence of any "offending" cycle in $\mathcal{T}^k$: The system is not joint$_k^{iop}$-diagnosable if and only if any state in a cycle contains the label "F".

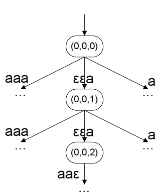


Fig. 7. Testing automata $\mathcal{T}^1$ in Example 2

Example 2: Consider the system introduced in Example 1. Construct the testing automaton as shown in Figure 7. Note at the initial state of $\mathcal{G}$, the transition on a leads the state machine to a "good" region satisfying the specification, where no state is labeled by "F", and there is no possibility of leaving that region. For the failure diagnosis purpose, we do not need to track traces in that "good" region. Thus we omit the corresponding part from the testing automation $\mathcal{T}^1$. It is seen that no "offending" cycle exist in $\mathcal{T}^1$. Thus, the system is joint$_1^{iop}$-diagnosable. For the same system, if the delay bound is $k=2$, then it can be verified that the system is not joint$_2^{iop}$-diagnosable.

VI. Diagnoser Synthesis

In the setting of decentralized diagnosis of [16], a local diagnoser at site i was taken to be $D_i = M_i(G \| \overline{R})$. Analogously, we can define an extended local diagnoser for site i to be $\mathcal{D}_i^k := \mathcal{M}_i(\mathcal{G}^k \| \overline{\mathcal{R}}^k) = \mathcal{M}_i(G \| \overline{\mathcal{R}}^k)$ for the diagnosis of a joint$_k^{iop}$-diagnosable system. As is the case with the verification, certain complexity reduction is also possible for the synthesis of local diagnosers. We define a "reduced" diagnoser at site i to be $\widetilde{\mathcal{D}}_i^k := \mathcal{M}_i(G \| \overline{\mathcal{R}}_i^k)$.

The *reachability set* $Reach_{\widetilde{\mathcal{D}}_i^k}(\cdot)$ (resp., $Reach_{\mathcal{D}_i^k}(\cdot)$) denotes the set of possible states of $\widetilde{\mathcal{D}}_i^k$ (resp., $\mathcal{D}_i^k$) reached by an execution of a trace in $L(\widetilde{\mathcal{D}}_i^k)$ (resp., $L(\mathcal{D}_i^k)$). Let $x^0_{\widetilde{\mathcal{D}}_i^k}$ denote the initial state of diagnoser $\widetilde{\mathcal{D}}_i^k$, and $\delta_{\widetilde{\mathcal{D}}_i^k}$ denote its transition function. The reachability set $Reach_{\widetilde{\mathcal{D}}_i^k}(\cdot)$ is computed recursively upon each incoming information as follows. $Reach_{\widetilde{\mathcal{D}}_i^k}(\varepsilon) = \varepsilon^*_{\widetilde{\mathcal{D}}_i^k}(x^0_{\widetilde{\mathcal{D}}_i^k})$; $\forall s \in L(\widetilde{\mathcal{D}}_i^k), \sigma \in \Lambda_1 \cup \Lambda_2$: $Reach_{\widetilde{\mathcal{D}}_i^k}(s\sigma) = \varepsilon^*_{\widetilde{\mathcal{D}}_i^k}(\delta_{\widetilde{\mathcal{D}}_i^k}(Reach_{\widetilde{\mathcal{D}}_i^k}(s), \sigma))$. The reachability set $Reach_{\mathcal{D}_i^k}(\cdot)$ is computed similarly. Let $(\Omega_F)_{\widetilde{\mathcal{D}}_i^k}$ denote the set of "failure states" of $\widetilde{\mathcal{D}}_i^k$, i.e., $(\Omega_F)_{\widetilde{\mathcal{D}}_i^k} = X \times \{F\} \times Z_{ji}^k$. Similarly, $(\Omega_F)_{\mathcal{D}_i^k} = X \times \{F\} \times Z_{12}^k \times Z_{21}^k$ denotes the failure states of diagnoser $\mathcal{D}_i^k$.

Theorem 2: Given a plant G and a specification model R with $L(R) \subseteq L(G)$, define $\mathcal{D}_i^k := \mathcal{M}_i(G \| \overline{\mathcal{R}}^k)$ and $\widetilde{\mathcal{D}}_i^k := \mathcal{M}_i(G \| \overline{\mathcal{R}}_i^k)$. Then, $Reach_{\mathcal{D}_i^k}(s) \subseteq (\Omega_F)_{\mathcal{D}_i^k}$ if and only if $Reach_{\widetilde{\mathcal{D}}_i^k}(s) \subseteq (\Omega_F)_{\widetilde{\mathcal{D}}_i^k}$.

Example 3: Let us revisit the system presented in Example 1. For the unit-delay case, we know from Example 2 that the system is joint$_1^{iop}$-diagnosable. Figure 8 shows the two local diagnosers $\widetilde{\mathcal{D}}_1^1$ and $\widetilde{\mathcal{D}}_2^1$. Assume that the plant executes a trace $s = ba$. Diagnoser 1 observes event a' followed by a communicated observation b' from diagnoser

2 with unit-delay. Diagnoser 2 observes event b' followed by a communicated observation a' from diagnoser 1 with 0 or 1 delay. The reachability sets $Reach_{\widetilde{\mathcal{D}}_1^1}$ and $Reach_{\widetilde{\mathcal{D}}_2^1}$ are computed as follows:

- $Reach_{\widetilde{\mathcal{D}}_1^1}(\epsilon) = \{(0,0,\epsilon),(4,F,b)\}$,
  $Reach_{\widetilde{\mathcal{D}}_1^1}(a') = \{(1,1,a),(1,1,\epsilon),(5,F,ba)\}$,
  $Reach_{\widetilde{\mathcal{D}}_1^1}(a'b') = \{(5,F,a),(5,F,\epsilon)\}$;
- $Reach_{\widetilde{\mathcal{D}}_2^1}(\epsilon) = \{(0,0,\epsilon),(1,1,a),(2,2,aa)\}$,
  $Reach_{\widetilde{\mathcal{D}}_2^1}(b') = \{(4,F,b),(5,F,ba),(4,F,\epsilon),(5,F,a),(6,F,aa)\}$,
  $Reach_{\widetilde{\mathcal{D}}_2^1}(b'a') = \{(5,F,\epsilon),(6,F,a),(6,F,ac)\}$.

When diagnoser 1 observes event a', it cannot determine whether the plant is at a "normal" state 1 or the "failure" state 5, and thus is ambiguous about whether a failure has occurred or not. After receiving the communicated event b' from diagnoser 2, diagnoser 1 is sure that the plant is at state 5, and a failure has occurred. On the other hand, since all elements in $Reach_{\widetilde{\mathcal{D}}_2^1}(b')$ have their second coordinate labeled "F", diagnoser 2 is certain about the occurrence of a failure after observing event b'.

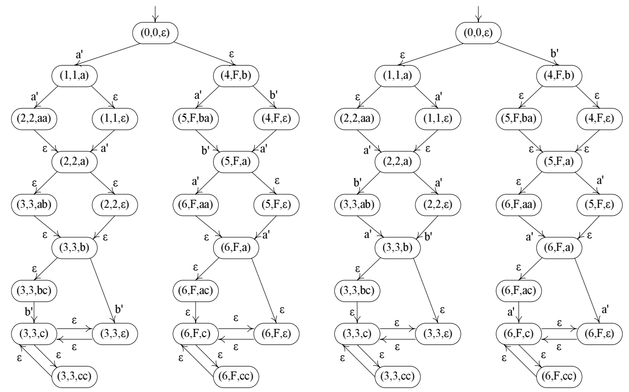


Fig. 8. Local diagnoser $\widetilde{\mathcal{D}}_1^1$ (left) and $\widetilde{\mathcal{D}}_2^1$ (right)

VII. Conclusions

This paper has studied the distributed failure diagnosis problem under the k-bounded communication delay. A similar setting has been used for distributed *control* in the work of Tripakis [20]. A notion of *joint$_k^{iop}$-diagnosability* is introduced so that any failure can be diagnosed within a bounded delay of its occurrence by one of the local sites using its own observations and the k-bounded delayed observations received from other local sites, which communicate among each other using the immediate observation passing protocol. We have shown that such problem of distributed diagnosis can be reduced to an instance of decentralized diagnosis. Results from [16] for decentralized diagnosis are applied for verifying joint$_k^{iop}$-diagnosability, and synthesizing local diagnosers. Further simplifications are presented to improve computational complexity.

References

[1] A. Beneveniste, E. Fabre, S. Haar, and C. Jard. Diagnosis of asynchronous discrete-event systems: A net unfolding approach. *IEEE Transactions on Automatic Control*, 48(5):714–727, 2003.

[2] A. Bouloutas, G. W. Hart, and M. Schwartz. Simple finite-state fault detectors for communication networks. *IEEE Trans. on Communications*, 40(3):477–479, March 1992.

[3] S. R. Das and L. E. Holloway. Characterizing a confidence space for discrete event timings for fault monitoring using discrete sensing and actuation signals. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, 30(1):52–66, 2000.

[4] R. Debouk, S. Lafortune, and D. Teneketzis. On the effect of communication delays in failure diagnosis of decentralized discrete event systems. volume 13, pages 263–289.

[5] R. Debouk, S. Lafortune, and D. Teneketzis. Coordinated decentralized protocols for failure diagnosis of discrete event systems. *Discrete Event Dynamical Systems: Theory and Applications*, 10:33–79, 2000.

[6] D. N. Godbole, J. Lygeros, E. Singh, A. Deshpande, and A. E. Lindsey. Communication protocols for a fault-tolerant automated highway system. *IEEE Transactions on Control Systems Technology*, 8(5):787–800, September 2000.

[7] C.N. Hadjicostis and G.C. Verghese. Power system monitoring based on relay and circuit breaker information. In *Proceedings of the 2001 IEEE International Symposium on Circuits and Systems*, volume 2, pages 197–200, May 2001.

[8] S. Jiang and R. Kumar. Diagnosis of repeated failures for discrete event systems with linear-time temporal logic specifications. In *Proceedings of IEEE Conference on Decision and Control*, pages 3221–3226, Maui, Hawaii, 2003.

[9] S. Jiang and R. Kumar. Failure diagnosis of discrete event systems with linear-time temporal logic fault specifications. *IEEE Transactions on Automatic Control*, 49(6):934–945, 2004.

[10] S. Jiang, R. Kumar, and H. E. Garcia. Diagnosis of repeated/intermittent failures in discrete event systems. *IEEE Transactions on Automatic Control*, 19(2):310–323, 2003.

[11] F. Lin. Diagnosability of discrete event systems and its applications. *Discrete Event Dynamic Systems: Theory and Applications*, 4(1):197–212, 1994.

[12] J. Lygeros, D. N. Godbole, and M. Broucke. A fault tolerant control architecture for automated highway system. *IEEE Transactions on Control Systems Technology*, 8(2):205–219, March 2000.

[13] R. E. Miller and A. K. Arisha. Fault identification in networks by passive testing. In *Proceedings of the IEEE Annual Simulation Symposium*, pages 277–284, 2001.

[14] D. Pandalai and L. Holloway. Template languages for fault monitoring of timed discrete event processes. *IEEE Transactions on Automatic Control*, 45(5):868–882, May 2000.

[15] A. D. Pouliezos and G. S. Stavrakakis. *Real time fault monitoring of industrial processes*. Kluwer Academic Publishers, Boston, MA, 1994.

[16] W. Qiu and R. Kumar. Decentralized failure diagnosis of discrete event systems. In *Proceedings of 2004 International Workshop on Discrete Event Systems*, Reim, France, September 2004.

[17] M. Sampath and S. Lafortune. Active diagnosis of discrete event systems. *IEEE Transactions on Automatic Control*, 43(7):908–929, 1998.

[18] M. Sampath, R. Sengupta, S. Lafortune, K. Sinaamohideen, and D. Teneketzis. Diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 40(9):1555–1575, September 1995.

[19] M. Sampath, R. Sengupta, S. Lafortune, K. Sinaamohideen, and D. Teneketzis. Failure diagnosis using discrete event models. *IEEE Transactions on Control Systems Technology*, 4(2):105–124, March 1996.

[20] S. Tripakis. Decentralized control of discrete-event systems with bounded or unbounded delay communication. *IEEE Transactions on Automatic Control*, 49(9):1489–1501, 2004.

[21] S. H. Zad, R. H. Kwong, and W. M. Wonham. Fault diagnosis in discrete-event systems: Framework and model reduction. *IEEE Transactions on Automatic Control*, 48(7):1199–1212, 2003.