This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3132642, IEEE Access

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Robust Diagnosability Analysis of Discrete Event Systems Using Labeled Petri Nets

SHIQI LI[1], MURAT UZAM[2], LI YIN[3], ZHAOFENG ZHONG[4], LI ZHENG[5], and NAIQI WU[3]

[1]School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China
[2]Department of Electrical and Electronics Engineering, Yozgat Bozok University, Yozgat 66100, Turkey
[3]Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau
[4]Hitachi Building Technology (Guangzhou) Co., Ltd, Guangzhou 510613, China
[4]ChangJi College, ChangJi 831100, China

Corresponding author: Li Yin (liyin@must.edu.mo).

This paper is supported by Guangzhou Innovation and Entrepreneurship Leading Team Project Funding under Grant No. 202009020008.

ABSTRACT Over the past decades, fault diagnosis of discrete event systems has many applications and attracts much attention from researchers and practitioners. With the increasingly high requirements on the reliability of cyber-physical systems such as automated manufacturing systems, fault detection technology has been unprecedentedly developed. Traditional approaches to diagnosability analysis of discrete event systems assume that all the communications between sensors and diagnosers work normally and correctly. However, communication failures may occur anytime, which may cause the loss of observations. This observation loss makes the traditional diagnoser fail or report incorrect information. The problem of fault diagnosis against intermittent loss of observations is addressed, i.e., robust diagnosability. In this paper, an approach to robust diagnosability analysis based on labeled Petri nets is presented. A necessary and sufficient condition for robust diagnosability is proposed. We also introduce a verification procedure of robust diagnosability using robust reachability diagnosers.

INDEX TERMS fault diagnosis, discrete event system, robust diagnosability, labeled Petri net

I. INTRODUCTION

Discrete event systems (DESs) [1] serve as a technical alias of a class of man-made systems, where computers are extensively integrated. Their modeling, analysis and control have received much attention from researchers and practitioners. Many problems are addressed by using Petri nets and finite state automata [2]–[6].

Fault detection is a necessary guarantee for the healthy and stable operation of a system, which motivates extensive studies in recent years [7]–[9]. For instance, in a manufacturing system, a fault may cause incorrect statistics of the remaining amount of raw materials. Besides, in a warehouse management system, wrong records of storage information will make the product be shipped out unexpectedly, which is also a sort of fault.

In the field of fault detection, a primary focus is on the problem of fault diagnosis (on-line diagnosis) [10], [11] and diagnosability analysis (off-line diagnosis) [12]. Fault diagnosis aims at telling whether a fault event has happened by analyzing and processing an observed string generated by a system. As for diagnosability analysis, one should recognize if the occurrence of a fault is able to be detected within a finite time delay. In this work, we focus on diagnosability analysis.

In the framework of automata, a method of fault diagnosis [13]–[15] is first proposed, where systems are modeled with nondeterministic finite-state automata, and failures are assumed as unobservable events. In these studies [13]–[15], a diagnoser is introduced to detect fault events, which provides diagnostics based on on-line observations.

As a graphical description of a DES, Petri nets have exact semantics and powerful expression capabilities [16], [17]. Due to their excellent modeling capability, especially their classification on different transitions [2], [18], labeled Petri nets (LPNs) are widely used to solve the problems such as unreliable observation [19]–[22], deadlock prevention [23]–[25], opacity verification [26], [27], etc.

Traditional diagnosis approaches rely on sensors to obtain information. Sensors may malfunction in practice, which

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3132642, IEEE Access

cannot provide correct and accurate information for a diagnoser. Therefore, the notion of robust diagnosis [28]–[30] is proposed to cope with this situation.

To overcome the deficiency of potential sensor failures, a robust diagnoser that provides redundancy in the diagnosis bases is designed in [28], which ensures the fault diagnosis with the occurrence of permanent sensor failures, i.e., failed sensors never recover anymore. In [31], a polynomial time construction procedure of a diagnoser verifier automaton is given to test the existence of controllers under permanent sensor failures.

In [29] and [30], the authors successively define robust diagnosability subject to intermittent sensor failures and intermittent loss of observations. In [29], sensors are supposed to fail at any time when collecting information, but they may recover at any time as well. To deal with the problem of fault diagnosis in the presence of intermittent sensor failures, the authors define a language operation, namely language dilation. This newly proposed operation takes the influence caused by intermittent sensor failures into account. As an extension work, the authors consider a more complex situation in [30], i.e., not only sensors but also the communication channels between sensors and the controller may fail at any time.

The authors in [32] solve the problem of finding sets of observabe events that ensure the language diagnosability of DESs in the framework of automata. An approach is proposed to obtain some certain event bases by inspecting the structure of a diagnoser automaton.

Based on LPNs, the authors of [33]–[35] present fault diagnosis approaches for DESs. According to [33], some transitions and failures are unobservable in a net system. Notions of basis markings and justifications are proposed to describe the markings that are consistent with observations [34], which improve the efficiency of fault diagnosability analysis.

Different from the monolithic architecture, the robust diagnosability of decentralized DESs is called robust co-diagnosability [36]–[38], which is first proposed in [39]. Due to the distributed property of some specific systems, their facilities may be scattered in different regions. Under this circumstance, centralized robust diagnosers cannot always be deployed. To solve this problem, a coordinator is designed. Each local diagnoser site is only allowed to communicate via the coordinator, which processes the information according to some set rules. Meanwhile, the authors in [30] and [46] also extend their conclusions to robust co-diagnosability.

In practice, traditional diagnosis models relying on sensors may fail due to the occurrences of an electronic component failure, communication failure, or atmospheric electromagnetic interference, etc. As a result, a system may experience intermittent loss of observations. Compared with [31] and [28], we investigate not only the permanent but also intermittent failure situations in this work, which expands the scope of the robustness. The work in [30] is based on the systems modeled with automata. In this paper, we also use LPNs, which highlights their advantages in exact semantics and powerful expression capabilities. Compared with the verifier net approach in [12], we construct a reachability diagnoser with through simple computation process with less computational cost.

This work is devoted to diagnosability analysis, especially the robust diagnosability problem in the framework of LPNs. To verify the language diagnosability of a specific bounded LPN, a diagnosis model namely a reachability diagnoser is designed based on reachability graphs. An algorithm is given to generate the reachability diagnoser of an LPN. Due to some external interference or internal malfunctions, observation information once transmitted normally may not reach the diagnoser. This observation loss may occur or disappear at any moment. Thus, traditional diagnosers may fail intermittently. In this case, we may never obtain accurate diagnosis information due to the lack of robustness of a diagnoser. Therefore, a new robust diagnosis model of LPN-modeled systems namely a robust reachability diagnoser is designed, which takes into account the intermittent loss of observations. We also derive a necessary and sufficient condition of language robust diagnosability by defining implicit indeterminate cycles in a robust reachability diagnoser. It can be summarized from the performance of a specific example in different situations. The proof is provided.

This paper is structured as follows. In Section II, we present the background and some notations of automata and Petri nets. Section III introduces the problem of diagnosability of DESs and the robust diagnosability problem. Section IV formulates the problem of language diagnosability in the framework of LPNs. The definition and condition of language diagnosability are followed. Section V illustrates the robust diagnosability analysis based on LPNs. An example under several situations is used to explore the necessary and sufficient condition for language robust diagnosability. Besides, the proofs of relevant theorems are given. Finally, conclusions are drawn in Section VI.

II. BACKGROUND ON AUTOMATA AND PETRI NETS

A. AUTOMATA

Let $G = (Q, \Sigma, f, q_0, Q_m)$ denote an automaton model of a DES, where Q is the set of states, Σ is the finite alphabet of symbols denoting event labels, $f : Q \times \Sigma \to Q$ is the transition function, q_0 is the initial state, and Q_m is the set of marked states. Assume that the event set Σ can be partitioned as $\Sigma_o \dot{\cup} \Sigma_{uo}$, where Σ_o and Σ_{uo} denote the sets of observable and unobservable events, respectively. The language generated by G, denoted by $\mathcal{L}(G)$ or simply by L, is defined as $L = \{s \in \Sigma^* \mid (\exists q \in Q) f(q_0, s) = q\}$. Let $L/s = \{u \in \Sigma^* \mid su \in L\}$ denote the post-language of L after string $s \in L$, where Σ^* denotes the Kleene-closure of Σ. Let ε be the empty string with its length being zero and $\varepsilon \in \sigma^*$.

The projection operation P_o is defined as $P_o : \Sigma^* \to \Sigma_o^*$,

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3132642, IEEE Access

which satisfies:

$$P_o(\varepsilon) = \varepsilon,$$
$$P_o(\sigma) = \begin{cases} \sigma, \text{if } \sigma \in \Sigma_o, \\ \varepsilon, \text{if } \sigma \in \Sigma_{uo}, \end{cases}$$
$$P_o(s\sigma) = P_o(s)P_o(\sigma), \text{for } s \in \Sigma^*, \sigma \in \Sigma.$$

The inverse projection operation P_o^{-1} , for $u \in \sigma_o$, is defined as

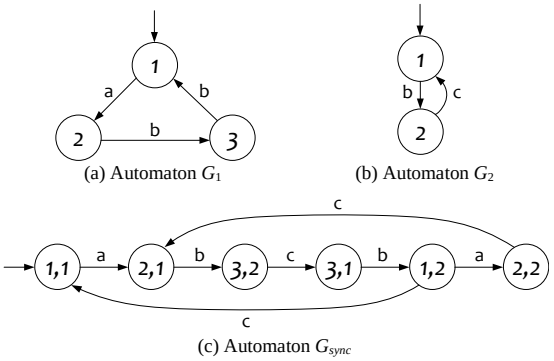
$$P_o^{-1}(u) = \{s \in \Sigma^* : P_o(s) = u\}.$$

Let $G_1 = (Q_1, \Sigma_1, f_1, q_{0,1}, Q_{m,1})$ and $G_2 = (Q_2, \Sigma_2, f_2, q_{0,2}, Q_{m,2})$ be two automata. The parallel (or synchronous) composition is defined as $G_{sync} = G_1 \parallel G_2 = Ac(Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, f_{1\|2}, (q_{0,1}, q_{0,2}), Q_{m,1} \times Q_{m,2})$, where $Ac(\cdot)$ denotes the accessible part of an automaton, which means all the states and transitions that are not reachable from the initial state are trimmed from the original automaton, and $f_{1\|2}$ is the transition function of G_{sync}.

Let $\Gamma : Q \to 2^Q$ denote the active event set at a state. The transition function $f_{1\|2}$ satisfies:

$$f_{1\|2}((q_1, q_2), \sigma)$$
$$= \begin{cases} (f_1(q_1, \sigma), f_2(q_2, \sigma)), \text{if } \sigma \in \Gamma_1(q_1) \cap \Gamma_2(q_2), \\ (f_1(q_1, \sigma), q_2), \text{if } \sigma \in \Gamma_1(q_1) \setminus \Sigma_2, \\ (q_1, f_2(q_2, \sigma)), \text{if } \sigma \in \Gamma_2(q_2) \setminus \Sigma_1. \end{cases}$$

Example 1: G_1 and G_2 in Figs. 1(a) and 1(b) are two automata with $\Sigma_1 = \{a, b\}$ and $\Sigma_2 = \{b, c\}$. Let G_{sync} be the parallel composition of G_1 and G_2, as shown in Fig. 1(c).



(a) Automaton G_1 (b) Automaton G_2

(c) Automaton G_{sync}

FIGURE 1. (a) G_1, (b) G_2, and (c) parallel composition G_{sync}.

A language $L \subseteq \Sigma^*$ is said to be live if for all $s \in L$, there exists an event $\sigma \in \Sigma$ such that $s\sigma \in L$. Let $\Sigma_f \subseteq \Sigma_{uo}$. For the sake of simplicity, we assume that there is only one fault event throughout this work, i.e., $\Sigma_f = \{\sigma_f\}$. Let $\Psi(\Sigma_f)$ denote the set of all finite strings ending with the event in Σ_f, $\| s \|$ denote the length of string s, and $\bar{s} = \{u \in \Sigma^* \mid (\exists v \in \Sigma^*)uv = s\}$ denote the prefix-closure of string s.

B. PETRI NETS

A Petri net is a four-tuple $N = (P, T, Pre, Post)$, where P is a set of m places, T is a set of n transitions, depicted by circles and bars (or boxes), respectively. $Pre : P \times T \to \mathbb{N}$ and $Post : P \times T \to \mathbb{N}$ are the pre- and post-incidence functions that specify the arcs from places to transitions and transitions to places, respectively, where $\mathbb{N}$ is the set of non-negative integers. Pre and $Post$ can be represented by $m \times n$ integer matrices indexed by P and T. The incidence matrix of Petri net N is defined as $C = Post - Pre$.

Let $x \in P \cup T$ be a node in a Petri net. The *preset* of x is defined as ${}^\bullet x = \{y \in P \cup T | Pre(y, x) > 0\}$. The *postset* of x is defined as $x^\bullet = \{y \in P \cup T | Post(y, x) > 0\}$. A net is said to be *pure* (or *self-loop free*) if it contains no self-loop; it is said to be *acyclic* if it has no circuit.

A marking is a mapping $M : P \to \mathbb{N}$ assigns a non-negative integer number of tokens (depicted by black dots) to a place. A marking can be tabulated and represented by an m-dimensional vector indexed by P. The marking of place p at M is denoted by $M(p)$, which indicates the number of tokens in place p at marking M. For economy of space, a marking M can be compactly written as a multi-set, i.e., $M = \Sigma_{p \in P} M(p)$.

A transition $t_j \in T$ is said to be enabled at marking M if for all $p \in {}^\bullet t_j$, $M(p) \geq Pre(p, t_j)$, which is denoted by $M[t_j\rangle$. An enabled transition t_j at M can fire, yielding a new marking M', which is denoted by $M[t_j\rangle M'$. According to the definition of incidence matrix C, M' can be calculated as

$$M' = M + C(\cdot, t_j) = M + C \cdot \vec{t_j},$$

where $\vec{t_j}$ is an n-dimensional canonical basic vector with its j-th entry being one.

A Petri net system $\langle N, M_0 \rangle$ is a structure with an initial marking M_0. A transition sequence $\sigma = t_1 t_2 \dots t_k \in T^*$, or called a *firing sequence*, is feasible (enabled) at marking M if there exist markings $M_1, M_2, \dots, M_k$ such that $M[t_1\rangle M_1[t_2\rangle \dots [t_k\rangle M_k$ holds, which is denoted as $M[\sigma\rangle M_k$, or simply $M[\sigma\rangle$ if M_k is of no interest. A marking M is said to be *reachable* if there exists a firing sequence σ such that $M_0[\sigma\rangle M$. The set of all the markings reachable from M_0 defines the *reachability set* of a Petri net, denoted as $R(N, M_0)$. A firing sequence $\sigma \in T^*$ is said to be faulty (or normal) if there has (or has not) at least a faulty transition $t_f \in T_f$ in σ.

A labeled Petri net (LPN) is a four-tuple $\mathcal{N} = (N, M_0, E, \ell)$, where $\langle N, M_0 \rangle$ is a Petri net system, E is an alphabet (a set of labels), and $\ell : T \to E \cup \{\varepsilon\}$ is the labeling function that assigns to each transition $t \in T$ a symbol from E or the empty word ε. The transition set can be partitioned into two disjoint sets, i.e., $T = T_o \cup T_u$, where T_o and T_u denote the set of observable transitions and that of unobservable transitions, respectively. The reachability set of an LPN is denoted by $R_{\mathcal{N}}$.

The unobservable subnet $\mathcal{N}_{us} = (P, T_{us}, Pre_{us}, Post_{us})$ is the net generated by removing all transitions $t \in T_o$ from $\mathcal{N}$, where Pre_{us} and $Post_{us}$ are the restrictions of Pre

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3132642, IEEE Access

and $Post$ to T_u, respectively. The incidence matrix of the unobservable subnet is denoted as $C_u = Post_{us} - Pre_{us}$.

For a transition sequence $\sigma \in T^*$ and the labeling function ℓ, $w = \ell(\sigma)$ is called an "observed word" or simply an "observation". The set of observed words (observations) from a marking M is denoted as:

$$\mathcal{L}(N, M) = \{w \in E^* | \exists \sigma \in T^*, M[\sigma\rangle, w = \ell(\sigma)\},$$

which is also called the *language* generated from $M \in R_{\mathcal{N}}$. $\mathcal{L}(N, M)$ is equivalently denoted as $L_{\mathcal{N}}$ is this paper.

Given an observation w of an LPN $\mathcal{N} = (N, M_0, E, \ell)$, we define

$$\mathcal{S}(w) = \{\sigma \in \mathcal{L}(N, M_0) | \ell(\sigma) = w\}$$

as the set of firing sequences consistent with $w \in E^*$ and

$$\mathcal{C}(w) = \{M \in R_{\mathcal{N}} | (\exists \sigma : \ell(\sigma) = w), M_0[\sigma\rangle M\}$$

as the set of reachable markings consistent with $w \in L_{\mathcal{N}}$.

An LPN is "bounded" if there exists $k \in \mathbb{N}^+ = \{1, 2, \ldots\}$, for all $M \in R(N, M_0)$, for all $p \in P$, $M(p) \leq k$ holds. It is "unbounded" if it is not bounded. An LPN $\mathcal{N}$ is said to be structurally bounded if it is bounded for any initial marking. Note that all the contribution in the framework of LPNs in this work is based on bounded LPNs.

III. DIAGNOSABILITY ANALYSIS USING AUTOMATA

A. DIAGNOSABILITY

Simply speaking, a language L is diagnosable if a failure of any type can be detected by observing an existing event record within a finite delay. The formal definition of language diagnosability can be stated as follows [13].

Definition 1 (Language diagnosability using automata): A live language L_G generated by an automaton $G = (Q, \Sigma, f, q_0, Q_m)$ is diagnosable with respect to projection $P_o : \Sigma^* \to \Sigma_o^*$ and $\Sigma_f \subseteq \Sigma$ if

$$(\exists n \in \mathbb{N})(\forall s \in \Psi(\Sigma_f))(\forall u \in L/s)(\| u \| \geq n \Rightarrow C_{\mathrm{D}}),$$

where the diagnosability condition C_{D} can be stated as:

$$(\nexists w \in L)[(P_o(su) = P_o(w)) \wedge (\bar{w} \cap \Psi(\Sigma_f) = \emptyset)].$$

In plain words, a language is said to be diagnosable with respect to P_o and Σ_f if and only if for any string s ending with a fault event and an arbitrarily long bounded string u, there is no fault-free string w such that $P_o(su) = P_o(w)$. One can easily verify the diagnosability of a specific language with the help of the diagnoser automaton. A diagnoser automaton G_d is defined as $G_d = (Q_d, \Sigma_o, f_d, q_{0,d}, Q_{m,d}) = Obs(G \parallel G_l, \Sigma_o)$, where $Obs(G, \Sigma_o)$ denotes the observer automaton of G with respect to Σ_o. The label automaton denoted as G_l is shown in Fig. 2.

Meanwhile, the definition of indeterminate observed cycles and a theorem about language diagnosability are also given in [13], which can be stated as follows.

Definition 2 (Indeterminate cycles): A set of uncertain states $q_{d_1}, q_{d_2}, \ldots, q_{d_n} \in Q_d$ is said to form an indeterminate cycle if:

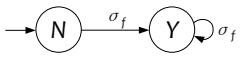


FIGURE 2. Label automaton G_l.

1) states $q_{d_1}, q_{d_2}, \ldots, q_{d_n}$ form a cycle in G_d with $f_d(q_{d_j}, \sigma_j) = q_{d_{j+1}}$, $j = 1, 2, \ldots, n-1$, $f_d(q_{d_n}, \sigma_n) = q_{d_1}$, where for all $j \in \{1, 2, \ldots, n\}$, $\sigma_j \in \Sigma_o$;
2) there exist $(q_j^k, Y), (\tilde{q}_j^r, N) \in q_{d_l}$, q_j^k is not necessarily distinct from $\tilde{q}_j^r$ for $j = 1, 2, \ldots, n$, $k = 1, 2, \ldots, m$ and $r = 1, 2, \ldots, m'$ in such a way that the sequence of states $\{q_j^k \mid j = 1, 2, \ldots, n, k = 1, 2, \ldots, m\}$ and $\{\tilde{q}_j^r \mid j = 1, 2, \ldots, n, r = 1, 2, \ldots, m'\}$ form their corresponding cycles in G', where

$$\begin{aligned} &f_{G'}(q_j^k, \sigma_j) = q_{j+1}^k, j = 1, 2, \ldots, n-1, \\ &\quad k = 1, 2, \ldots, m, \\ &f_{G'}(q_n^k, \sigma_n) = q_1^{k+1}, k = 1, 2, \ldots, m-1, \\ &f_{G'}(q_n^m, \sigma_n) = q_1^1, \end{aligned}$$

and

$$\begin{aligned} &f_{G'}(\tilde{q}_j^r, \sigma_j) = \tilde{q}_{j+1}^r, j = 1, 2, \ldots, n-1, \\ &\quad r = 1, 2, \ldots, m', \\ &f_{G'}(\tilde{q}_n^r, \sigma_n) = \tilde{q}_1^{r+1}, r = 1, 2, \ldots, m'-1, \\ &f_{G'}(\tilde{q}_n^{m'}, \sigma_n) = \tilde{q}_1^1. \end{aligned}$$

A generator automaton G' can be simply understood as an observable version of G, where $f_{G'}(q, \sigma) = q'$ if $f(q, s) = q'$ for some $s \in L_\sigma(G)$, i.e., $L(G') = P_o[L(G)]$.

Theorem 1: Assume that L is live, i.e., $\Gamma(q_i) \neq \emptyset$ for all $q_i \in Q$, and there is no cycle of unobservable events in G. The language L generated by automaton G is diagnosable with respect to P_o and Σ_f if and only if its diagnoser automaton G_d has no indeterminate cycle.

The proof of Theorem 1 can be found in [13], therefore omitted here. With Theorem 1, one can efficiently determine the diagnosability of a specific language.

B. ROBUST DIAGNOSABILITY

A robust diagnosability problem has been proposed in [30], where it assumes that some sensors or communication channels between sensors and diagnosers may fail intermittently. Under this circumstance, Σ_o is partitioned as $\Sigma_o = \Sigma_{ilo} \dot{\cup} \Sigma_{nilo}$, where Σ_{ilo} denotes the set of events subject to intermittent loss of observations and Σ_{nilo} denotes the set of events not subject to intermittent loss of observations. Observable events in Σ_{ilo} may convert into unobservable ones with respect to the occurrence of observation losses, which makes the original diagnoser stall or provide wrong diagnosis information. To describe the intermittent loss of observations mathematically, $\Sigma'_{ilo} = \{\sigma' : \sigma \in \Sigma_{ilo}\}$ is defined and σ' representing that the unobservable fault event is assigned to each $\sigma \in \Sigma_{ilo}$.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3132642, IEEE Access

Let $\Sigma = \Sigma_{ilo} \dot{\cup} \Sigma_{nilo} \dot{\cup} \Sigma_{uo}$. Note that Σ_o is now equivalent with $\Sigma_{ilo} \dot{\cup} \Sigma_{nilo}$. We have $\Sigma'_{ilo} = \{\sigma' : \sigma \in \Sigma_{ilo}\}$ and $\Sigma_{dil} = \Sigma \cup \Sigma'_{ilo}$. The dilation is the mapping $D : \Sigma^* \to 2^{\Sigma^*_{dil}}$ where

$$D(\varepsilon) = \varepsilon,$$
$$D(\sigma) = \begin{cases} \{\sigma\}, & \text{if } \sigma \in \Sigma \setminus \Sigma_{ilo}, \\ \{\sigma, \sigma'\}, & \text{if } \sigma \in \Sigma_{ilo}, \end{cases}$$
$$D(s\sigma) = D(s)D(\sigma), s \in \Sigma^*, \sigma \in \Sigma.$$

The dilation operation D can be executed to not only strings but also languages, i.e.,

$$D(L) = \bigcup_{s \in L} D(s).$$

Meanwhile, the modified automaton model, namely a dilated automaton $G_{dil} = (Q, \Sigma_{dil}, f_{dil}, q_0, Q_m)$ and the projection operation $P_{dil,o} : \Sigma^*_{dil} \to \Sigma^*_o$ can be defined in [30].

Definition 3 (Language robust diagnosability): A live language L generated by G is robustly diagnosable with respect to projection $P_{dil,o} : \Sigma^*_{dil} \to \Sigma^*_o$ and $\Sigma_f \subseteq \Sigma$ if

$$(\exists n \in \mathbb{N})(\forall s \in \Psi(\Sigma_f))(\forall u \in L/s)(\| u \| \geq n \Rightarrow C_{\mathrm{RD}}),$$

where the robust diagnosability condition C_{RD} is there does not exist $w \in L$ such that

$$(P_{dil,o}(D(su)) = P_{dil,o}(D(w))) \wedge (\bar{w} \cap \Psi(\Sigma_f) = \emptyset).$$

According to Definition 3, language $L(G)$ is robustly diagnosable with respect to D, $P_{dil,o}$, and Σ_f if and only if for any string s ending with a fault event and an arbitrarily long bounded string u, there is no fault-free string w such that $P_{dil,o}(D(su)) = P_{dil,o}(D(w))$. In [30], two ways including robust diagnosers and verifiers are given to test the robust diagnosability.

IV. DIAGNOSABILITY ANALYSIS IN THE FRAMEWORK OF LPNS

Let T_o, T_u, and T_f denote the sets of observable transitions, unobservable transitions, and faulty transitions, respectively. Let $\Omega(T_f)$ denote the set of all finite firing sequences ending with a transition in T_f. The definition of language diagnosability in the framework of LPNs can be stated as follows.

Definition 4 (Language diagnosability in the framework of LPNs): Let $\mathcal{N} = (N, M_o, E, \ell)$ be an LPN system with labeling function $\ell : T \to E \cup \{\varepsilon\}$. Transition set T can be partitioned into $T = T_o \cup T_u$ with $T_o \cap T_u = \emptyset$. Let $T_f \subseteq T_u$ be a set of fault events. The prefix-closed language $L_{\mathcal{N}}$ is diagnosable if

$$(\forall n \in \mathbb{N})(\forall s \in \Omega(T_f))(\forall u \in T^*/s)$$
$$(\ell(su) := w_f)(\| u \| \geq n \Rightarrow C_{\mathrm{LPND}}),$$

where the diagnosability condition C_{LPND} is

$$(\nexists w \in L_{\mathcal{N}})[(w_f = w) \wedge (\overline{\mathcal{S}(w)} \cap \Omega(T_f) = \emptyset)].$$

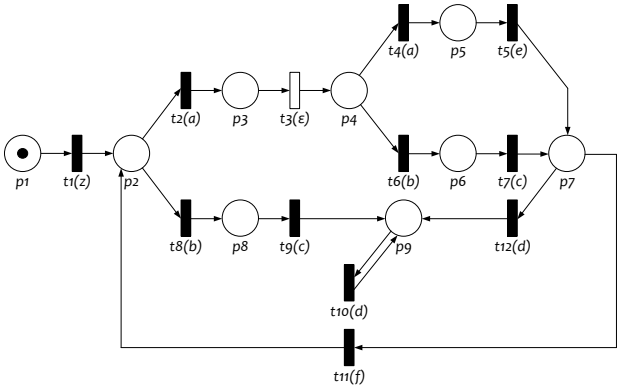


FIGURE 3. An LPN model $\mathcal{N}$.

Fig. 3 shows an LPN $\mathcal{N}$ with a Petri net system $\langle N, M_0 \rangle$ and labeling function $\ell : T \to E \cup \{\varepsilon\}$, $P = \{p_1 - p_9\}$, $T = \{t_1 - t_{12}\}$, $M_0 = p_1$, and $E = \{a, b, c, d, e, f, z\}$. Note that t_3 labeled with ε is a fault unobservable transition and the other 11 transitions in T are observable, i.e., $T_f = \{t_3\}$. The language $L_{\mathcal{N}}$ is $z(a\varepsilon\{ae \cup bc\}f)^*(\{a\varepsilon\{ae \cup bc\}d^*\} \cup \{bcd^*\})$. The reachability graph of $\mathcal{N}$ is shown in Fig. 4.

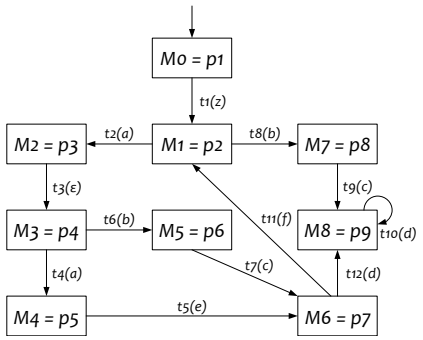


FIGURE 4. Reachability graph of $\mathcal{N}$.

Example 2: Consider a firing sequence $\sigma_1 = t_1t_2t_3t_4t_5t_{12}t_{10}^n$. We have $w_{f_1} := \ell(\sigma_1) = za\varepsilon aed^n$ and $\mathcal{C}(w_{f_1}) = \{M_8\}$. Obviously, σ_1 is the catenation of a faulty sequence $s_1 = t_1t_2t_3$ and a bounded arbitrarily long normal sequence $u_1 = t_4t_5t_{12}t_{10}^n$ whose corresponding words are $za\varepsilon$ and aed^n, respectively. It is not difficult to find that the system cannot reach the same marking with anyone in $\mathcal{C}(w_{f_1})$ by firing any sequence that has no faulty transition in it. The firing of a sequence including faulty transitions such as σ_1 can be strictly distinguished from the firing of other normal transition sequences. By observing the word generated by $\mathcal{N}$, we can always infer whether a faulty transition has been fired. Thus the language generated by the LPN system is diagnosable.

Using diagnosers can readily verify the language diagnosability of a system. Similar to the method using automata, we here propose a new concept, i.e., the reachability diagnoser

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3132642, IEEE Access

of an LPN. Let $\mathcal{N}_d$ denote the reachability diagnoser of $\mathcal{N}$, which can be obtained from $\mathcal{N}$ with Algorithm 1 as follows.

Algorithm 1 Construction of $\mathcal{N}_d$

Input: An LPN $\mathcal{N}$ with a net system $\langle N, M_0\rangle$.
Output: A reachability diagnoser $\mathcal{N}_d$.

1: the root node is M_0 that has initially no label;
2: **while** there are nodes with no label, **do**
3: consider a node M with no label;
4: (a) for each transition t enabled at M:
5: let $M' = M + C(\cdot, t)$;
6: **if** there does not exist a node M' in the graph, **then**
7: add it;
8: add an arc t from M to M';
9: **end if**
10: (b) label the node M "old";
11: **end while**
12: remove all labels from nodes;
13: create automaton $G = (Q, \Sigma, f, q_0, Q_m)$ where $Q = \emptyset$, $\Sigma = \emptyset$, $q_0 = M_0$, and $Q_m = \emptyset$;
14: **while** M_k exists, **do**
15: let $q_k = M_k$, $Q = Q \cup \{q_k\}$;
16: **if** there does not exist the label of t in Σ with $M_k = M' + C(\cdot, t)$, **then**
17: let $\sigma = t$, $\Sigma = \Sigma \cup \{\sigma_{k_j}\}$;
18: **end if**
19: **end while**
20: create automaton $G_{sync} = G \parallel G_l$;
21: let $\mathcal{N}_d = Obs(G_{sync}, \Sigma_o)$, where Σ_o is the observable events set in G;
22: **return** $\mathcal{N}_d$.

In plain words, to construct a reachability diagnoser, we first need to construct the reachability graph of $\mathcal{N}$. Then we construct an automaton G having the same scale of space with the reachability graph of $\mathcal{N}$. Markings and transitions in the reachability graph should be converted into states and transitions in the automaton. Obviously, these two models are isomorphic, which guarantees that the conversion can be finished successfully. Finally, $\mathcal{N}_d$ can be obtained by generating the observer over $G \parallel G_l$ with respect to Σ_o. It is not difficult to find that each node in either the reachability graph or the automaton is traversed once. For the consistent size of the two models, Algorithm 1 has an $\mathcal{O}(n)$ time complexity for the conversion process, where n represents the number of markings in the reachability graph of $\mathcal{N}$.

Fig. 5 shows the reachability diagnoser of $\mathcal{N}$. By inspecting Fig. 5, all cycles in $\mathcal{N}_d$ are observable and none of them satisfies the definition of indeterminate cycles, which is stated as follows.

Definition 5 (Indeterminate Cycles of $\mathcal{N}_d$.): A set of uncertain states $q_{d_1}, q_{d_2}, \ldots, q_{d_n} \in Q_d$ is said to form an indeterminate cycle if:

1) states $q_{d_1}, q_{d_2}, \ldots, q_{d_n}$ form a cycle in $\mathcal{N}_d$ with $f_d(q_{d_j}, \sigma_j) = q_{d_{j+1}}$, $j = 1, 2, \ldots, n-1$, $f_d(q_{d_n}, \sigma_n) = q_{d_1}$, where for all $j \in \{1, 2, \ldots, n\}$, $\sigma_j \in \Sigma_o$;

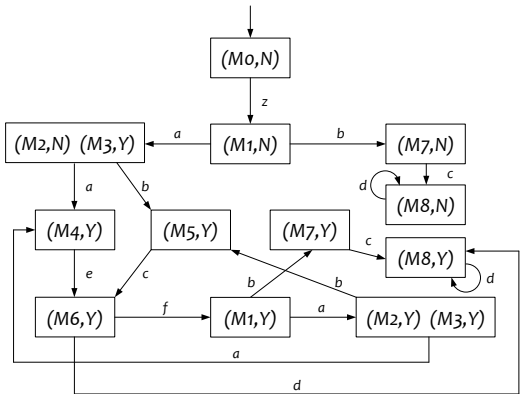


FIGURE 5. Reachability diagnoser of $\mathcal{N}$.

2) there exist $(M_j^k, Y), (\tilde{M}_j^r, N) \in q_{d_l}$, M_j^k is not necessarily distinct from $\tilde{M}_j^r$ for $j = 1, 2, \ldots, n$, $k = 1, 2, \ldots, m$, and $r = 1, 2, \ldots, m'$ in such a way that the sequence of states $\{M_j^k \mid j = 1, 2, \ldots, n, k = 1, 2, \ldots, m\}$ and $\{\tilde{M}_j^r | j = 1, 2, \ldots, n, r = 1, 2, \ldots, m'\}$ form their corresponding cycles in the generator of the reachability graph of $\mathcal{N}$, where

$$M_j^k[t_j\rangle M_{j+1}^k,\ j = 1, 2, \ldots, n-1,\ k = 1, 2, \ldots, m,$$
$$M_n^k[t_n\rangle M_1^{k+1},\ k = 1, 2, \ldots, m-1,$$
$$M_n^m[t_n\rangle M_1^1,$$

and

$$\tilde{M}_j^r[t_j\rangle \tilde{M}_{j+1}^r,\ j = 1, 2, \ldots, n-1,\ r = 1, 2, \ldots, m',$$
$$\tilde{M}_n^r[t_n\rangle \tilde{M}_1^{r+1},\ r = 1, 2, \ldots, m'-1,$$
$$\tilde{M}_n^{m'}[t_n\rangle \tilde{M}_1^1.$$

With the definition of indeterminate cycles of $\mathcal{N}_d$, a necessary and sufficient condition for LPN language diagnosability can be obtained, as presented below.

Theorem 2: A live language $L_{\mathcal{N}}$ generated by an LPN $\mathcal{N} = (N, M_o, E, \ell)$ whose unobservable subnet is acyclic is diagnosable with respect to $\ell : T \to E \cup \{\varepsilon\}$ and $T_f = \{t_f : \ell(t_f) = \varepsilon\}$ if and only if its reachability diagnoser $\mathcal{N}_d$ has no observable indeterminate cycles.

Theorem 2 as a different version illustrated with LPNs is actually consistent with Theorem 1 since the reachability graph of $\mathcal{N}$ is essentially an automaton. Thus, the proof of Theorem 2 is similar to Theorem 1.

Proof: (Necessity) Consider the reachability diagnoser $\mathcal{N}_d = (Q_d, \Sigma_o, f_d, q_{0,d}, Q_{m,d})$ and let G denote the reachability graph of the LPN $\mathcal{N}$. By contrapositive, assume that there exist states $q_{d_1}, q_{d_2}, \ldots, q_{d_n} \in Q_d$ forming an indeterminate cycle. Let $f_d(q_{d_i}, \sigma_i) = q_{d_{i+1 \bmod n}}$, indicating that these n states are connected with n transitions in order, from end to end. Let $(q_l^k, Y), (\tilde{q}_l^r, N) \in q_{d_l}$, $l = 1, 2, \ldots, n$,

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3132642, IEEE Access

$k = 1, 2, \ldots, m$, and $r = 1, 2, \ldots, m'$ form corresponding cycles in the generator automaton of G. We have:

$$f(q_l^k, s_l^k\sigma_l) = q_{l+1}^k,\ l = 1, 2, \ldots, n-1,\ k = 1, 2, \ldots, m,$$
$$f(q_n^k, s_n^k\sigma_n) = q_1^{k+1},\ k = 1, 2, \ldots, m-1,$$
$$f(q_n^m, s_n^m\sigma_n) = q_1^1,$$

and

$$f(\tilde{q}_l^r, \tilde{s}_l^r\sigma_l) = \tilde{q}_{l+1}^r,\ l = 1, 2, \ldots, n-1,\ r = 1, 2, \ldots, m',$$
$$f(\tilde{q}_n^r, \tilde{s}_n^r\sigma_n) = \tilde{q}_1^{r+1},\ r = 1, 2, \ldots, m'-1,$$
$$f(\tilde{q}_n^{m'}, \tilde{s}_n^{m'}\sigma_n) = \tilde{q}_1^1,$$

where $m, m' \in \mathbb{N}$, $s_l^k, \tilde{s}_l^r \in L(G)$, and $s_l^k, \tilde{s}_l^r \in \Sigma_{uo}^*$.

Thanks to (q_1^1, Y), $(\tilde{q}_1^1, N) \in q_1$, there exist s_0, $\tilde{s}_0 \in L$ such that $f(q_0, s_0) = (q_1^1, Y)$, $f(\tilde{q}_0, \tilde{s}_0) = (\tilde{q}_1^1, N)$ and $P_o(s_0) = P_o(\tilde{s}_0)$. That is to say, $\Sigma_f \in s_0$, $\Sigma_f \notin \tilde{s}_0$, and $\Sigma_f \notin \tilde{s}_l^r$ hold for all l and r.

Consider two transition sequences:

$$\begin{aligned} w =& s_0(s_1^1\sigma_1 s_2^1\sigma_2 \cdots s_n^1\sigma_n s_1^2\sigma_1 s_2^2\sigma_2 \cdots \\ & s_n^2\sigma_n s_1^m\sigma_1 s_2^m\sigma_2 \cdots s_n^m\sigma_n)^{km'} \\ \tilde{w} =& \tilde{s}_0(\tilde{s}_1^1\sigma_1 \tilde{s}_2^1\sigma_2 \cdots \tilde{s}_n^1\sigma_n \tilde{s}_1^2\sigma_1 \tilde{s}_2^2\sigma_2 \cdots \\ & \tilde{s}_n^2\sigma_n \tilde{s}_1^{m'}\sigma_1 \tilde{s}_2^{m'}\sigma_2 \cdots \tilde{s}_n^{m'}\sigma_n)^{km} \end{aligned}$$

with an arbitrary length k. We have $w, \tilde{w} \in L$, $P_o(w) = P_o(\tilde{w}) = P_o(s_0)(\sigma_1\sigma_2\cdots\sigma_n)^{kmm'}$. Let $s \in \bar{s_0}$ ending with σ_f, i.e., $s \in \Psi(\Sigma_f)$ and $u \in L/s$ such that $w = su$. To satisfy the condition set in Definition 4, we can always find a constant k such that $\| u \| > n$ for any $n \in \mathbb{N}$. Thus, we have $\tilde{w} \in P_o^{-1}[P_o(su)]$ while $\Sigma_f \notin \tilde{w}$. Under this circumstance, the choice of s violates the rule in Definition 4, i.e., L is not diagnosable.

(Sufficiency) Assume that the reachability diagnoser $\mathcal{N}_d$ has no indeterminate cycle. Select any $s \in L$ such that $s \in \Psi(\Sigma_f)$ and let $f(q_0, s) = q$. Select any $u_1 \in P_o(L)$. Let $f(q_o, su_1) = q_1$, meanwhile $f_d[q_{d_0}, P_o(su_1)] = q_{d_1}$ in G_d. By $\Sigma_f \in su_1$, we have (q_{d_1}, Y). There are now two cases to be discussed:

1) State q_{d_1} is certain. Since su_1 is faulty, for each $w \in P_o^{-1}[P_o(su_1)]$ as a longer sequence containing all the symbols in su_1, there must be a faulty event σ_f in it. In addition, the assumptions in Theorem 2 require no unobservable cycles in G. Thus, w without any faulty event does not exist and L is diagnosable.
2) State q_{d_1} is uncertain. Recall Definition 4 and two subcases arise here:
   a) There are no cycles of uncertain states in G_d. Having no cycles of uncertain states implies that one can always acquire a definite Y or N about whether any faulty event happens, which intuitively satisfies the requirement of language diagnosability.
   b) There exists a cycle of uncertain states q_{d_1}, q_{d_2}, …, $q_{d_n} \in Q_d$, but it is not an indeterminate one. Suppose that the system cannot loop in the cycle for an arbitrarily long time after the occurrence of a faulty event such that the fault can be detected eventually. In that case, it can be determined that the language is diagnosable.
   If there is a cycle in G' formed with N-states from G_d, there cannot simultaneously exist any cycle formed with Y-states in G'. Thus, G_d will reach a Y-state once a fault happens. On the contrary, the existence of a cycle in G' formed with Y-states from G_d will make it impossible for any cycle formed with N-states to exist. Select any N-state from $q_{d_l} \in Q_d$. It is obvious that it cannot be a successor of any Y-state along the corresponding sequence in G'. Hence, G_d will stay in the cycle forever if a fault has happened, or leave the cycle eventually if everything works normally.

In summary, we have proved that the live language generated by an LPN whose unobservable subnet is acyclic is diagnosable if and only if its reachability diagnoser has no indeterminate cycle. The condition and conclusion of the result are mutually necessary and sufficient. □

V. ROBUST DIAGNOSABILITY ANALYSIS IN THE FRAMEWORK OF LPNS

In the previous part of this paper, we have introduced robust diagnosability. We will explain how to analyse robust diagnosability in the framework of LPNs by a specific example.

Example 3: Consider the sequence $\sigma_1 = t_1t_2t_3t_4t_5t_{12}t_{10}^n$ again. Assume that at some specific moment, the transitions labeled with a cannot be recorded as usual, i.e., transitions t_2, t_4 are subject to intermittent loss of observations. Suppose that the firing information of t_2 cannot reach the diagnoser for a while and, somehow, the communication restores before the firing of t_4. For $\mathcal{N}_d$, the first event to be recognized is z produced by t_1. The reachability diagnoser $\mathcal{N}_d$ move into (M_1, N). Since a produced by t_2 has been lost and the fault transition t_3 is unobservable, a produced by t_4 is the next letter $\mathcal{N}_d$ receives, which makes $\mathcal{N}_d$ move into $(M_2, N)(M_3, Y)$. Notice that the next letter $\mathcal{N}_d$ should have dealt with is e, however, e is not a legal sequel event of $(M_2, N)(M_3, Y)$ in $\mathcal{N}_d$. That is to say, $\mathcal{N}_d$ gets stuck due to the intermittent loss of a.

Now consider another sequence $\sigma_2 = t_1t_2t_3t_6t_7t_{12}t_{10}^n$, $w_{f_2} = \ell(\sigma_2) = za\varepsilon bcd^n$. Similarly, suppose that the transition labeled with a has been fired but failed to be recorded for some reason. After recognizing z, $\mathcal{N}_d$ moves into (M_1, N) and the following sequence $\mathcal{N}_d$ should have received is bcd^n. $\mathcal{N}_d$ individually recognizes all the letters successfully, and it stays in (M_8, N) finally, which indicates that the fault transition has never been fired. Under this circumstance, $\mathcal{N}_d$ reports incorrect information about the occurrence of fault transitions.

It is not difficult to find that the traditional diagnoser model has certain limitations when dealing with a system containing intermittent loss of observations. With the help

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3132642, IEEE Access

of dilation operation, language robust diagnosability using LPNs is defined as follows.

Definition 6 (Language robust diagnosability in the framework of LPNs): Given an LPN $\mathcal{N} = (N, M_o, E, \ell)$ with $\langle N, M_0\rangle$ satisfying $T = T_o \cup T_u$ and $T_f \subseteq T_u$, the prefix-closed language $L_{\mathcal{N}}$ generated by $\mathcal{N}$ is robustly diagnosable if

$$(\forall n \in \mathbb{N})(\forall s \in \Omega(T_f))(\forall u \in T^*/s)$$

$$(w_f := \ell(su))(\| u \| \geq n \Rightarrow C_{\text{LPNRD}}),$$

where the robust diagnosability condition C_{LPNRD} is

$$(\nexists w \in L_{\mathcal{N}})$$

$$[(\ell(\mathcal{S}(D(w_f))) = \ell(\mathcal{S}(D(w)))) \wedge (\overline{\mathcal{S}(w)} \cap \Omega(T_f) = \emptyset)].$$

Let us check the LPN in Fig. 3 again. According to the analysis in the previous section, $L_{\mathcal{N}}$ is diagnosable with respect to Σ_f and P_o. Suppose again that the transitions labeled with a are subject to intermittent loss of observations. With the help of dilation operation, $\mathcal{N}$ can be dilated as $\mathcal{N}'_{dil}$, as shown in Fig. 6. Two unobservable transitions t'_2 and t'_4 labeled with a' are added and assigned to t_2 and t_4. Meanwhile, the reachability graph of $\mathcal{N}'_{dil}$ is shown in Fig. 7.

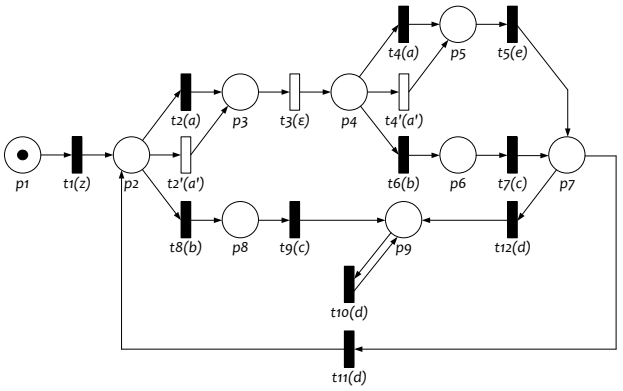


FIGURE 6. $\mathcal{N}'_{dil}$ with a subject to ILO.

Faulty sequence $s_1u_2 := \sigma_2$ is from $\Omega(T_f)$, for $\ell(\sigma_2) = w_{f_2} = za\varepsilon bcd^n$. We have $D(w_{f_2}) = \{za\varepsilon bcd^n, za'\varepsilon bcd^n\}$. Notice that there exists a normal sequence $\sigma_{2,n} = t_1t_8t_9t_{10}^n$, for $\ell(\sigma_{2,n}) = w_{2,n} = zbcd^n$ such that $\mathcal{C}(w_{2,n}) = \{M_8\}$. In plain words, the system $\mathcal{N}'_{dil}$ can reach M_8 by firing either a faulty sequence σ_2 or a normal sequence $\sigma_{2,n}$. Therefore, the diagnoser cannot accurately distinguish whether a fault transition has fired. That is to say, the language generated by $\mathcal{N}$ is not robustly diagnosable when the transitions labeled with a are subject to intermittent loss of observations.

Similar to diagnosability, language robust diagnosability can also be verified using corresponding diagnosers named robust reachability diagnosers. Fig. 8 shows $\mathcal{N}'_{dil,d}$, the robust reachability diagnoser of $\mathcal{N}'_{dil}$. By inspecting Fig. 8, the existence of the observed cycle labeled with d on $(M_8, Y)(M_8, N)$ makes $L_{\mathcal{N}}$ lose its robust diagnosability.

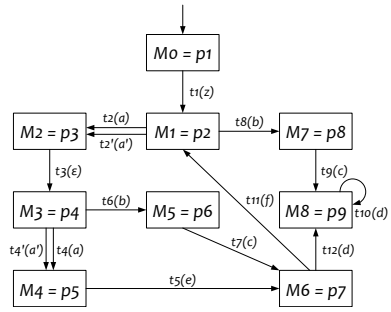


FIGURE 7. Reachability graph of $\mathcal{N}'_{dil}$.

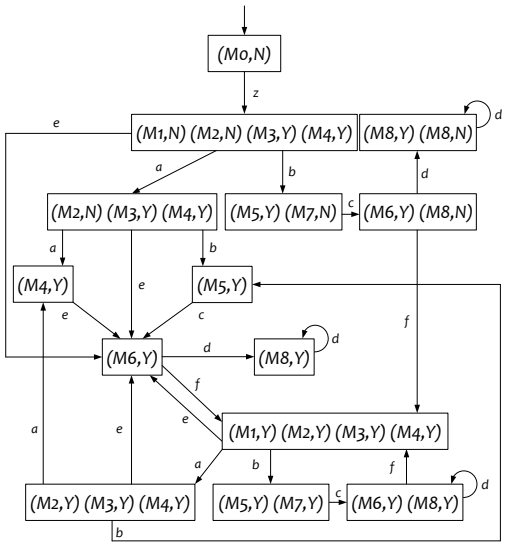


FIGURE 8. Robust reachability diagnoser $\mathcal{N}'_{dil,d}$.

Example 4: Now suppose that transitions t_{10} and t_{12} labeled with d are unsuccessfully observed intermittently.

In Fig. 9, $\mathcal{N}''_{dil}$ shows the dilated net structure. The reachability graph and $\mathcal{N}''_{dil,d}$ are shown in Figs. 10 and 11, respectively.

Cycles depicted with dashed lines appear in $\mathcal{N}''_{dil,d}$ since t_{10} forms a self-loop in the original net structure. When the firing of t_{10} fails to be recognized by the reachability diagnoser, unobservable t'_{10} fires, which is denoted by dashed lines. It is not difficult to find that not only observable indeterminate cycles labeled with d but also unobservable cycles labeled with d' can make $L_{\mathcal{N}}$ no longer robustly diagnosable.

Consider (M_6Y, M_8N, M_8Y) in Fig. 11. It seems that $\mathcal{N}''_{dil,d}$ will never know whether the faulty transition has happened if the transitions labeled with d never recover from malfunctions, which makes t_{10} and t_{12} keep firing silently. Now, it seems that not only observed indeterminate cycles but also unobservable cycles on uncertain states can influence language robust diagnosability. In order to improve relevant results, a subclass of indeterminate cycles has to be defined.

Definition 7 (Implicit Cycles and Indeterminate Implicit Cy-

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3132642, IEEE Access

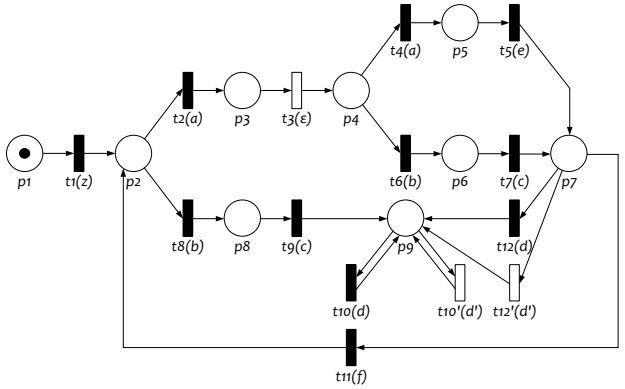


FIGURE 9. $\mathcal{N}''_{dil}$ with d subject to ILO.

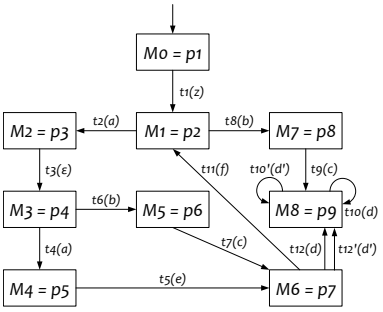


FIGURE 10. Reachability graph of $\mathcal{N}''_{dil}$.

cles of $\mathcal{N}_{dil,d}$): A cycle formed with states q_{dil}, d_1, q_{dil}, d_2, …, q_{dil}, $d_n \in \mathcal{N}_{dil,d}$ is defined as an implicit cycle if the following conditions hold:

- q_{dil,d_1}, q_{dil,d_2}, …, q_{dil,d_n} form a cycle in $\mathcal{N}_{dil,d}$ with $f_d(q_{dil})$, d_j, $\sigma_j) = q_{dil}$, d_{j+1}, $j = 1, 2, \ldots, n-1$, $f_d(q_{dil,d_n}), \sigma_n) = q_{dil,d_1}$, where for all $j \in \{1, 2, \ldots, n\}$, $\sigma_j \in \Sigma_{uo}$.

The implicit cycle is also indeterminate if it additionally satisfies the second condition:

- $q_{dil,d_1}, q_{dil,d_2}, \ldots, q_{dil,d_n}$ form an indeterminate cycle in $\mathcal{N}_{dil,d}$.

In plain words, an implicit indeterminate cycle is an indeterminate cycle formed with unobservable events between uncertain states. Note that not all implicit cycles are indeterminate, i.e., the cycle on certain states cannot be counted, which brings no ambiguity to robust diagnosability. For instance, the cycle labeled with d' on $(M_8 Y)$ is unobservable; however, it is not an indeterminate one. Meanwhile, indeterminate cycles satisfying only Definition 5 will be called explicit indeterminate cycles literally and their existence has been proven to make language not diagnosable. The word "explicit" is meant to contrast with "implicit", which emphasizes the observable property of an indeterminate cycle. An explicit indeterminate cycle here is exactly the indeterminate cycle defined in Definition 5.

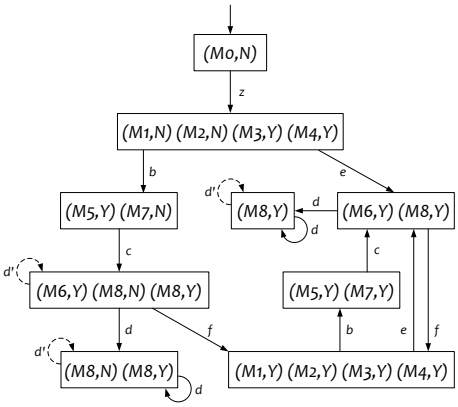


FIGURE 11. Robust reachability diagnoser $\mathcal{N}''_{dil,d}$.

Example 5: Consider $\mathcal{N}$ if transitions labeled with b, i.e., t_6, t_8 are subject to intermittent loss of observations. By dilating t_6 and t_8, $\mathcal{N}'''_{dil}$ can be depicted as Fig. 12. The reachability graph and robust reachability diagnoser are shown in Figs. 13 and 14. There is neither (explicit) indeterminate cycle nor implicit indeterminate cycle in $\mathcal{N}'''_{dil}$. Therefore, $\mathcal{N}$ is robustly diagnosable if transitions labeled with b are subject to intermittent loss of observations.

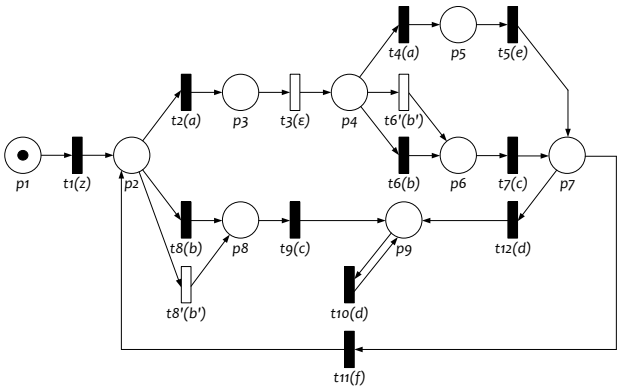


FIGURE 12. $\mathcal{N}''''_{dil}$ with b subject to ILO.

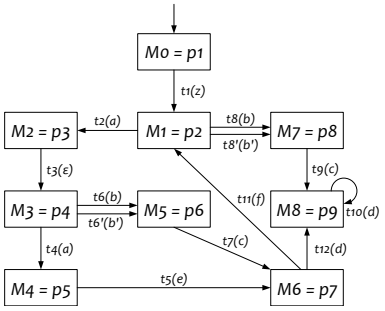


FIGURE 13. Reachability graph of $\mathcal{N}''''_{dil}$.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3132642, IEEE Access

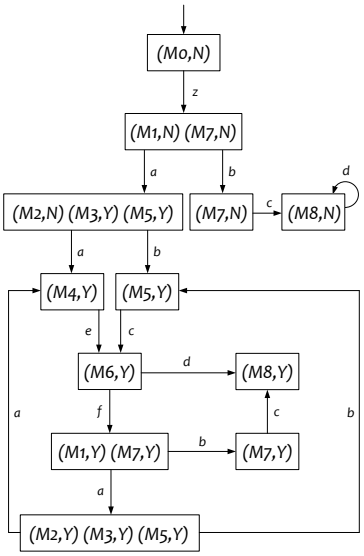


FIGURE 14. Robust reachability diagnoser $\mathcal{N}'''_{dil,d}$.

Theorem 3: A live language $L_{\mathcal{N}}$ generated by an LPN $\mathcal{N} = (N, M_0, E, \ell)$ with respect to $D : T^* \to 2^{(T_{dil})^*}$ and $T_f \subseteq T_u$ is robustly diagnosable if and only if its robust reachability diagnoser $\mathcal{N}_{dil,d}$ has no explicit or implicit indeterminate cycle.

Proof: (Necessity) Consider the circumstance that there exists an implicit indeterminate cycle in $\mathcal{N}_{dil,d}$. Obviously, there is a state $q_{dil,d}$ that is formed with n states $M_{i_1}, M_{i_2}, \ldots, M_{i_n}$ from the reachability graph of $\mathcal{N}_{dil}$ attached with a diagnosis label.

It is not difficult to find that there exists a faulty sequence $suv \in L_{\mathcal{N}\,dil}$, for $\ell(suv) = w_f$. Sequence suv satisfies:

1) s is faulty, i.e., $\overline{\mathcal{S}(s)} \cap \Omega(T_f) \neq \emptyset$ such that $f_{dil,d}(q_{0_{dil,d}}, P_o(s)) = q_{dil,d}$ holds,
2) u is unobservable, i.e., $u \in (T'_{ilo} \cup T_u)^*$ such that $f_{dil,d}(q_{0_{dil,d}}, P_o(su)) = q_{dil,d}$ holds,
3) v is also unobservable with a length of k, i.e., $t \in (T'_{ilo} \cup T_u)^*$ such that $f_{dil,d}(q_{0_{dil,d}}, P_o(suv)) = q_{dil,d}$ and $M_{i_1}[v\rangle M_{i_j}$ hold, where $j = (k \bmod n) + 1$.

Notice that $q_{dil,d}$ is an uncertain state, implying that there also exists $w \in L_{\mathcal{N}\,dil}$ such that $\overline{\mathcal{S}(s)} \cap \Omega(T_f) = \emptyset$ and $P_o(suv) = P_o(w)$. In other words, the existence of an implicit indeterminate cycle indicates that the projection of a faulty sequence and a normal sequence are the same, which violates the condition of language robust diagnosability.

(Sufficiency) Assume that $L_{\mathcal{N}}$ is not robustly diagnosable with respect to D and T_f. According to Definition 6 of language robust diagnosability using LPNs, for some $w_f = \ell(su)$, where $s \in \Omega(T_f)$, $u \in T^*/s$, and $\| u \| \geq n$ for $n \in \mathbb{N}$, there exists some $w \in L_{\mathcal{N}}$ such that $\mathcal{C}(D(w_f)) = \mathcal{C}(D(w))$ and $\overline{\mathcal{S}(w)} \cap \Omega(T_f) = \emptyset$ hold, where w could have bounded or unbounded length.

Since the sets of reachable markings consistent with $D(w_f)$ and $D(w)$ are the same and $\mathcal{N}_{dil,d}$ is deterministic, there exists an uncertain state $q_{dil,d} \in Q_{dil,d}$ satisfying

$$q_{dil,d} = f_{dil,d}(q_{0_{dil,d}}, P_o(D(w_f))) = f_{dil,d}(q_{0_{dil,d}}, P_o(D(w))).$$

Because of the arbitrary length of w stipulated in Definition 6, we assume that the number of states in $Q_{dil,d}$ is x with $\| w \| > x$. A cycle of states formed with corresponding transitions will exist in $\mathcal{N}_{dil,d}$. There are two cases that need to be discussed:

1) There exists at least one observable transition in the cycle. Under this circumstance, the cycle is an explicit indeterminate cycle.
2) All transitions in the cycle are unobservable. It this case, the cycle is an implicit indeterminate cycle.

These two cases show us that the existence of an (explicit or implicit) indeterminate cycle is inevitable as long as a language is not robustly diagnosable. □

Note that in the proof of necessity, we omit the situation of explicit indeterminate cycle since it has been proven in the proof of Theorem 1. In summary, we have explored three different situations:

1) The transitions labeled with a influenced by intermittent loss of observations: Explicit indeterminate cycles appear in $\mathcal{N}'_{dil,d}$ such that $L_{\mathcal{N}}$ is not robustly diagnosable, as we have discussed in the previous section.
2) The transitions labeled with d influenced by intermittent loss of observations: We find that the existence of implicit indeterminate cycles in $\mathcal{N}''_{dil,d}$ also makes a language lose its robust diagnosability. Hence, we give a necessary and sufficient condition for language robust diagnosability and prove it.
3) The transitions labeled with b influenced by intermittent loss of observations: No more (explicit or implicit) indeterminate cycles exist in $\mathcal{N}'''_{dil,d}$ this time. Language $L_{\mathcal{N}}$ is robustly diagnosable.

By analyzing these three situations, we have derived a necessary and sufficient condition for robust diagnosability. The situation in which any other transition is affected by the intermittent loss of observations always falls into one of the three above. Therefore, we do not spell them out anymore.

VI. CONCLUSIONS

In this paper, we address the problem of robust diagnosability against intermittent loss of observations using LPNs. Communications between transitions (sensors) in a system and the diagnoser may sometimes fail, which makes the diagnoser get stuck or report incorrect information. With the help of a dilation operation, possible malfunctions between sensors and the diagnoser can be taken into account. We first summarize the definition of language diagnosability, indeterminate cycles, and robust diagnosability using automata. Then we extend the relevant definitions and theorems to the system modeled with LPNs. We introduce a dilation operation and a construction algorithm of robust reachability diagnoser using it. Two necessary and sufficient conditions C_{LPND} and C_{LPNRD} for language diagnosability and robust

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3132642, IEEE Access

diagnosability generated by the system modeled with LPNs are derived. The proof of the relevant theorem is given. Thus, the robust diagnosability problem against intermittent loss of observations of language generated by LPNs can be solved.

The research is conducted under the framework of bounded LPNs. For unbounded LPNs, the construction complexity of their reachability graphs increases exponentially, which is difficult to be coped with when using reachability diagnosers introduced in this work. In fact, there is a more compact way representing reachable markings and the relation between them of an LPN, namely a basis reachability graph. Its applications to supervisory control and fault diagnosis under other settings [40]– [44] and performance in robust diagnosability analysis are worthy of further exploration [46].

This work is carried out based on a centralized diagnoser. In the future, the problem of robust co-diagnosability analysis with a decentralized architecture deserves to be explored further, i.e., whether or how the conclusion drawn in this work should be modified to apply to distributed systems. Meanwhile, fault events are assumed to be unobservable in this work. Suppose that some malicious attacks can change the behavior of a system, i.e., operations such as insertion, removal, and replacement of transition labels. In this case, deciding how to carry out the robust diagnosability analysis if a wrong symbol or label is observed from the fault event is worthy of consideration. It is of interest to consider the rumor propagation mechanism for fault diagnosis and propagation [45].

REFERENCES

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

[2] S. Wang, M. Zhou, Z. Li, and C. Wang, "A new modified reachability tree approach and its applications to unbounded Petri nets," *IEEE Trans. Syst., Man, Cybern.*, vol. 43, no. 4, pp. 932–940, 2013.

[3] S. Lai, D. Nessi, M. P. Cabasino, A. Giua, and C. Seatzu, "A comparison between two diagnostic tools based on automata and Petri nets," in *Proc. 9th International Workshop on Discrete Event Systems*, 2008, pp. 144–149.

[4] M. Khalgui, O. Mosbahi, Z. Li, and H.-M. Hanisch, "Reconfigurable multiagent embedded control systems: From modeling to implementation," *IEEE Trans. Comput.*, vol. 60, no. 4, pp. 538–551, 2010.

[5] C. Hadjicostis and G. Verghese, "Monitoring discrete event systems using Petri net embeddings," *Proc. International Conference on Application and Theory of Petri Nets*, 1999, pp. 188–207.

[6] M. Khalgui, O. Mosbahi, Z. Li, and H.-M. Hanisch, "Reconfiguration of distributed embedded-control systems," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 4, pp. 684–694, 2010.

[7] D. Lefebvre and C. Delherm, "Diagnosis of DES with Petri net models," *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 1, pp. 114–118, 2007.

[8] J. Zaytoon and S. Lafortune, "Overview of fault diagnosis methods for discrete event systems," *Annual Reviews in Control*, vol. 37, no. 2, pp. 308–320, 2013.

[9] G. Zhu, Z. Li, and N. Wu, "Model-based fault identification of discrete event systems using partially observed Petri nets," *Automatica*, vol. 96, pp. 201–212, 2018.

[10] S. Bavishi and E. K. P. Chong, "Automated fault diagnosis using a discrete event systems framework," in *Proceedings of 9th IEEE International Symposium on Intelligent Control*, 1994, pp. 213–218.

[11] S. Hashtrudi Zad, R. H. Kwong, and W. M. Wonham, "Fault diagnosis in discrete-event systems: framework and model reduction," in *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*, vol. 4, 1998, pp. 3769–3774 vol.4.

[12] M. P. Cabasino, A. Giua, S. Lafortune, and C. Seatzu, "A new approach for diagnosability analysis of Petri nets using verifier nets," *IEEE Trans. Autom. Control*, vol. 57, no. 12, pp. 3104–3117, 2012.

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

[14] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. C. Teneketzis, "Failure diagnosis using discrete-event models," *IEEE Trans. Control Syst. Technol.*, vol. 4, no. 2, pp. 105–124, 1996.

[15] M. Sampath, S. Lafortune, and D. Teneketzis, "Active diagnosis of discrete-event systems," *IEEE Trans on Automatic Control*, vol. 43, no. 7, pp. 908–929, 1998.

[16] T. Murata, "Petri nets: Properties, analysis and applications," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[17] N. Ran, A. Giua, and C. Seatzu, "Enforcement of diagnosability in labeled Petri nets via optimal sensor selection," *IEEE Trans. Autom. Control*, vol. 64, no. 7, pp. 2997–3004, 2019.

[18] N. Wu, Z. Li, and T. Qu, "Energy efficiency optimization in scheduling crude oil operations of refinery based on linear programming," *Journal of Cleaner Production*, vol. 166, pp. 49–57, 2017.

[19] F. G. Cabral, M. V. Moreira, O. Diene, and J. C. Basilio, "A Petri net diagnoser for discrete event systems modeled by finite state automata," *IEEE Trans. Autom. Control*, vol. 60, no. 1, pp. 59–71, 2015.

[20] M. P. Cabasino, A. Giua, and C. Seatzu, "Diagnosis of discrete event systems using labeled Petri nets," *IFAC Proceedings Volumes*, vol. 42, no. 5, pp. 52–57, 2009.

[21] G. Zhu, Z. Li, N. Wu, and A. Al-Ahmari, "Fault identification of discrete event systems modeled by Petri nets with unobservable transitions," *IEEE Trans. Syst., Man, Cybern.*, vol. 49, no. 2, pp. 333–345, 2017.

[22] E. Athanasopoulou, L. Li, and C. N. Hadjicostis, "Maximum likelihood failure diagnosis in finite state machines under unreliable observations," *IEEE Trans. Autom. Control*, vol. 55, no. 3, pp. 579–593, 2010.

[23] Z. Li and N. Wei, "Deadlock control of flexible manufacturing systems via invariant–controlled elementary siphons of Petri nets," *The International Journal of Advanced Manufacturing Technology*, vol. 33, no. 1-2, pp. 24–35, 2007.

[24] Z. Li, S. Zhu, and M. Zhou, "A divide-and-conquer strategy to deadlock prevention in flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. C*, vol. 39, no. 2, pp. 156–169, 2009.

[25] G. Liu, P. Li, Z. Li, and N. Wu, "Robust deadlock control for automated manufacturing systems with unreliable resources based on Petri net reachability graphs," *IEEE Trans. Syst., Man, Cybern.*, vol. 49, no. 7, pp. 1371–1385, 2018.

[26] Y. Tong, Z. Li, C. Seatzu, and A. Giua, "Verification of state-based opacity using Petri nets," *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 2823–2837, 2017.

[27] H. Grichi, O. Mosbahi, M. Khalgui, and Z. Li, "Rwin: New methodology for the development of reconfigurable wsn," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 109–125, 2016.

[28] S. Lima, "Robust diagnosis of discrete-event systems subject to permanent sensor failures," *Proc. Int Workshop on Discrete Event Systems*, 2010, pp. 90–97.

[29] L. K. Carvalho, J. C. Basilio, and M. V. Moreira, "Robust diagnosability of discrete event systems subject to intermittent sensor failures," *IFAC Proceedings Volumes*, vol. 43, no. 12, pp. 84–89, 2010.

[30] L. K. Carvalho, J. C. Basilio, and M. V. Moreira, "Robust diagnosis of discrete event systems against intermittent loss of observations," *Automatica*, vol. 48, no. 9, pp. 2068–2078, 2012.

[31] K. R. Rohloff, "Sensor failure tolerant supervisory control," in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 3493–3498.

[32] J. C. Basilio, S. Lima, S. Lafortune, and M. V. Moreira, "Computation of minimal event bases that ensure diagnosability," *Discrete Event Dynamic Systems*, vol. 22, no. 3, pp. 249–292, 2012.

[33] M. P. Cabasino, A. Giua, and C. Seatzu, "Fault detection for discrete event systems using Petri nets with unobservable transitions," *Automatica*, vol. 46, no. 9, pp. 1531–1539, 2010.

[34] M. P. Cabasino, A. Giua, M. Pocci, and C. Seatzu, "Discrete event diagnosis using labeled Petri nets. an application to manufacturing systems," *Control Engineering Practice*, vol. 19, no. 9, pp. 989–1001, 2011.

[35] Z. Ma, Z. Li, and A. Giua, "Characterization of admissible marking sets in Petri nets with conflicts and synchronizations," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1329–1341, 2016.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3132642, IEEE Access

[36] W. B. Qiu and R. Kumar, "Decentralized failure diagnosis of discrete event systems," *IEEE Trans. Syst., Man, Cybern. A*, vol. 36, no. 2, pp. 384–395, 2006.

[37] M. Dotoli, M. P. Fanti, and A. M. Mangini, "Fault detection of discrete event systems using Petri nets and integer linear programming," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 6554–6559, 2008.

[38] R. K. Boel and J. H. van Schuppen, "Decentralized failure diagnosis for discrete-event systems with costly communication between diagnosers," in *Proc. Sixth International Workshop on Discrete Event Systems*, 2002, pp. 175–181.

[39] A. Paoli and S. Lafortune, "Robustcodiagnosability of discrete event systems," in *Proc. 42nd IEEE Conference on Decision and Control*, 2009.

[40] G. Zhu, Z. Li, and N. Wu, "Model-based fault identification of discrete event systems using partially observed Petri nets," *Automatica*, vol. 96, pp. 201–212, 2018.

[41] Y. Tong, Z. Li, C. Seatzu, and A. Giua, "Decidability of opacity verification problems in labeled Petri net systems," *Automatica*, vol. 80, pp. 48–53, 2017.

[42] H. Zhang, L. Feng, N. Wu, and Z. Li, "Integration of learning-based testing and supervisory control for requirements conformance of black-box reactive systems," *IEEE Transactions on Automation Science and Engineering*, vo. 15, no. 1, pp. 2–15, 2017.

[43] Y. F. Chen, Z. W. Li, A. Al-Ahmari, "Nonpure Petri net supervisors for optimal deadlock control of flexible manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 2, pp. 252–265, 2012.

[44] Y. Tong, Z. Li, C. Seatzu, and A. Giua, "Decidability of opacity verification problems in labeled Petri net systems," *Automatica*, vol. 80, pp. 48–53, 2017.

[45] L. Yang, Z. Li, and A. Giua, "Containment of rumor spread in complex social network," *Information Sciences*, vol. 506, pp. 113–130, 2020.

[46] N. Ran, H. Su, A. Giua, and C. Seatzu, "Codiagnosability analysis of bounded Petri nets," *IEEE Trans. Autom. Control*, vol. 63, no. 4, pp. 1192–1199, 2018.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/