IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 December 2022, accepted 10 January 2023, date of publication 16 January 2023, date of current version 1 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3236963

RESEARCH ARTICLE

Robust Diagnosability Analysis Using Basis Reachability Graph

SHIQI LI[1], SIAN ZHOU[2], LI YIN[2], AND RONGTIAN JIANG[3]
[1]School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China
[2]Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau
[3]Hitachi Building Technology (Guangzhou) Company Ltd., Guangzhou 510613, China

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

This work was supported by the Guangzhou Innovation and Entrepreneurship Leading Team Project under Grant 202009020008.

ABSTRACT Fault-freeness is one of the necessary guarantees for healthy and stable operations of discrete event systems. Traditional diagnostic models may fail since sensors may suffer electronic component failures, communication failures, or atmospheric electromagnetic interference. Robust diagnosis problem has attracted more and more attention since it improves reliability of the diagnosis technology. When generating the Petri net reachability graph necessary for constructing a reachability diagnoser, one may face the state explosion problem. As a Petri net gets more complicated, the complexity of its reachability graph also increases exponentially. With the basis reachability graph, a lightweight diagnoser named a basis reachability diagnoser is developed, together with a robust basis reachability diagnoser obtained by dilating the basis reachability graph. Compared with reachability graphs, the advantages of basis reachability graphs in construction complexity are confirmed and the efficiency in robust diagnosability analysis is improved.

INDEX TERMS Discrete event system, diagnosability, basis reachability graph, robust diagnosability.

I. INTRODUCTION

Fault-freeness is necessary for the healthy and stable operation of systems, and is the goal of all system designers and producers. With the rapid development of information technology and industrial automation control, human-made control systems such as automatic manufacturing systems [1], [2], [3], power grids [4], transportation systems [5], and communication networks [6] are becoming more sophisticated. Due to their meticulous structures, a tiny failure in these systems may cause immeasurable losses. Under this circumstance, fault diagnosis of discrete event systems (DESs) [7], [8] as a hot research area has received much attention from researchers and practitioners in recent years [9], [10], [11], [12], [13], [14], [15].

Within the framework of nondeterministic finite-state automata (NFA), a fault diagnosis method is proposed in [16], [17], where faults are assumed to be unobservable events. The authors introduce a diagnoser, which is an individual procedure, to detect and isolate a fault event. The diagnoser can provide diagnostics using on-line observations of the system behavior. The methods presented in [16] are applicable for any DES without being modified. An integrated approach is presented in [18] to control and diagnose the occurrence of failures.

Petri nets are a kind of precise semantic representation of DESs with powerful expression capabilities [19]. These advantages motivate more and more DES researchers to study Petri nets [20], [21], [22], [23], [24], [25]. In [20], the author briefly reviews the history and the application areas of Petri nets, introducing behaviors, structural properties and some subclasses of Petri nets. Among them, labeled Petri nets (LPNs) are extensively employed to deal with fault diagnosis [22], [26], [27], deadlock prevention [28], [29], opacity verification [24], [30], [31], [32], etc., in DES owing to their outstanding modeling ability, particularly their categorization in different transitions.

The work in [26] provides an LPN-based DES fault detection method, which assumes that part of transitions are unobservable in a mesh system, including those that represent faults. Basis markings and justifications are introduced to characterize markings that consistent with given

The associate editor coordinating the review of this manuscript and approving it for publication was Laura Celentano.

observations. The authors also implement their work off-line if the net system is bounded, which optimizes the complexity. Later in [22], the authors develop a MATLAB diagnostics toolbox and verify the new approach's effectiveness in [26].

In [33], the diagnosability problem is presented in the form of an example from the automotive industry. Since exhaust gas does not burn, recirculation reduces the peak combustion temperatures. Thus, the valve may fail. A series of control sequences is designed to diagnose the valve malfunction, which can be abstracted as an off-line diagnosis procedure, i.e., a diagnosability analysis procedure. In [34], a problem from the automotive industry of a vehicle braking system equipped with an antilock brake system is solved by diagnosability analysis using Petri nets.

To improve the efficiency of diagnosability analysis using LPNs [35], [36], the authors introduce a compact way to represent the reachability graph using basis markings, i.e., a subset of a system's reachable markings [37]. It can be confirmed that employing basis markings in diagnosability analysis rather than reachable markings offers a sizable complexity advantage since the number of basis markings is typically smaller than that of reachable markings.

Sensors are used to provide data in conventional diagnosis. However, in practice, the sensors themselves may be faulty, which will prevent the system from obtaining valid observations. There are also many other external issues that can invalidate the diagnosis model. Therefore, the notion of robust diagnosis [38], [39], [40], [41] is introduced to deal with this complexity.

In [42], supervisory control in the DES framework with sensor failures as partial observations is discussed. All sensors are initialized to observe occurrences of events and transmit them to the controller. However, if a sensor fails, it will not be able to transmit its information any longer. Under this circumstance, observability is redefined, which tolerates the existence of faulty sensors. Note that the author assumes that the faulty sensor never recovers from a malfunction in this work.

Robust co-diagnosability is introduced in [38]. It is assumed that all partial observers are subject to failures. The way to solve the diagnosability problem with these broken observers is discussed. The authors also present two different tests, one utilizing verifier automata and the other with diagnoser automata.

The authors in [40] and [41] respectively consider the robust diagnosability problem with intermittent sensor failures and intermittent observations loss. The former presupposes that the sensors could malfunction at any time while gathering data, but that they could also recover from the malfunction at any time. The latter takes into account the more complicated scenario in which the sensors are operational but communication between them and the controller can break down at any point. Similarly, the communication channel may restore. This problem gives rise to the definition of robust diagnosability. Meanwhile, the authors extend the conclusions to robust co-diagnosability.

This work is dedicated to the study of robust diagnosability problem within the framework of LPNs. The rest of this paper is organized as follows. Section II reviews the primary notions and definitions of automata and Petri nets. Section III recalls the definitions of language diagnosability, indeterminate cycles, reachability diagnosers, and language robust diagnosability. Section IV focuses on the computational complexity of constructing a reachability graph. We try to reconstruct our diagnosis model in a more compact way. We give the generation algorithm of a (robust) basis reachability diagnoser combined with the proposed dilation operation. We propose a novel and more refined necessary and sufficient condition for language robust diagnosability in combination with robust basis reachability diagnosers. Section V provides a numerical experiment in which we compare the efficiency of two diagnosis models. Section VI summarizes the theme, method, and result of this work. The possible direction in which this work may continue in the future is also proposed.

II. PRELIMINARIES

A. AUTOMATA

A deterministic finite-state automaton (DFA), denoted by G, is a five-tuple

$$G = (Q, \Sigma, f, q_0, Q_m), \tag{1}$$

where Q denotes a set of states, $\Sigma = \{a, b, c, \cdots\}$ indicates a set of events, $f : Q \times \Sigma \rightarrow Q$ is called partial transition function, q_0 is the initial state and $Q_m \subseteq Q$ is a set of marked states. Set Σ^* is the Kleene-closure of Σ. A language defined over Σ is a set of finite-length strings formed from events in Σ, i.e., a subset of Σ^*. The transition function f can be extended to $Q \times \Sigma^* \rightarrow Q$ by defining $f(q, se) = f(f(q, s), e)$, where $s \in \Sigma^*$ and $e \in \Sigma$. Generally, we write $f(q, e)!$ if $f(q, e)$ is defined. A transition function is also recorded as (q_1, e, q_2) in some scenarios, which is equivalent to $f(q_1, e) = q_2$.

We define projection $P_o : \Sigma^* \rightarrow \Sigma_o^*$, and it satisfies:

$$\begin{aligned}
&P_o(\varepsilon) = \varepsilon,\\
&P_o(e) = \begin{cases} e, \text{if } e \in \Sigma_o,\\ \varepsilon, \text{if } e \in \Sigma_{uo},\end{cases}\\
&P_o(se) = P_o(s)P_o(e), \text{for } s \in \Sigma^*, e \in \Sigma,
\end{aligned} \tag{2}$$

where the alphabet Σ is partitioned into the set of observable events Σ_o and the set of unobservable events Σ_{uo}, i.e., $\Sigma_o \cup \Sigma_{uo} = \Sigma$ and $\Sigma_o \cap \Sigma_{uo} = \emptyset$. The inverse projection operation P_o^{-1} is defined as

$$P_o^{-1}(u) = \{s \in \Sigma^* : P_o(s) = u\}. \tag{3}$$

Let $G_1 = (Q_1, \Sigma_1, f_1, q_{0,1}, Q_{m,1})$ and $G_2 = (Q_2, \Sigma_2, f_2, q_{0,2}, Q_{m,2})$. The synchronous (or parallel) composition is defined as:

$$\begin{aligned}
G_{sync} =&G_1 \parallel G_2\\
=&Ac(Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2,\\
&f_{1\parallel 2}, (q_{0,1}, q_{0,2}), Q_{m,1} \times Q_{m,2}),
\end{aligned} \tag{4}$$

where $Ac(\cdot)$ is the accessible part of an automaton. Any state or transition that cannot be reached from the initial state will be trimmed by $Ac(\cdot)$ operation. Let $\Gamma : Q \rightarrow 2^{\Sigma}$ denote the feasible event set in a state, i.e., $\Gamma(q) = \{e \in \Sigma | f(q, e)!\}$. The transition function denoted as $f_{1\|2}$ satisfies:

$$f_{1\|2}((q_1, q_2), e) = \begin{cases} (f_1(q_1, e), f_2(q_2, e)), & \\ \quad \text{if } e \in \Gamma_1(q_1) \cap \Gamma_2(q_2), & \\ (f_1(q_1, e), q_2), & \\ \quad \text{if } e \in \Gamma_1(q_1) \setminus \Sigma_2, & \\ (q_1, f_2(q_2, e)), & \\ \quad \text{if } e \in \Gamma_2(q_2) \setminus \Sigma_1. & \end{cases} \tag{5}$$

Let $\Sigma_f \subseteq \Sigma_{uo}$ denote the set of fault events. For ease of understanding, we only consider one fault event in this paper, i.e., $\Sigma_f = \{e_f\}$. Let s_f represent the ending event of s. Let $\bar{s}$ denote the prefix-closure of s, i.e., the set of all sequences that are prefixes of s. Similarly, $\bar{L} = \bigcup_{s \in L} \bar{s}$ defines the prefix-closure of language L. The length of s is $\| \; s \; \|$. Let $\Psi(\Sigma_f)$ denote the set of strings in a language L that end with a fault event, i.e., $\Psi(\Sigma_f) = \{s \in L | s_f \in \Sigma_f\}$. Let $\Sigma_f \in s$ denote $\bar{s} \cap \Psi(\Sigma_f) \neq \emptyset$ by a slight abuse of notation, i.e., s includes at least one fault event from Σ_f. Set $L/s = \{u \in \Sigma^* : su \in L\}$ is the *post*-language of L after trace $s \in L$.

B. PETRI NETS

A Petri net, denoted by N, is a four-tuple

$$N = (P, T, Pre, Post), \tag{6}$$

where $P = \{p_1, p_2, \ldots, p_m\}$ denotes the set of m places depicted with circles and $T = \{t_1, t_2, \ldots, t_n\}$ is the set of n transitions depicted with bars, respectively. $Pre : P \times T \rightarrow \mathbb{N}$ and $Post : P \times T \rightarrow \mathbb{N}$ are the *pre*- and *post*-incidence functions. Let $C = Post - Pre$ be the incidence matrix.

Given a transition $t \in T$, the preset of t is defined as

$$^{\bullet}t = \{p \in P | Pre(p, t) > 0\}. \tag{7}$$

A marking is a vector $M : P \rightarrow \mathbb{N}$ that assigns a non-negative integer number to each place. This number represents the amount of tokens, which is generally depicted as black dots (or recorded directly) in a place. We use an m-dimensional vector indexed by P to denote a marking M with $M = \Sigma_{p \in P} M(p)$. An initial marking describes the very first status of a net, i.e., $\langle N, M_0 \rangle$ means a Petri net initialized with M_0.

A transition $t_j \in T$ is enabled at M if for all $p \in {}^{\bullet}t_j$, $M(p) \geq Pre(p, t_j)$, denoted by $M[t_j\rangle$. An enabled transition t_j can fire at marking M and yields a new marking M', denoted by $M[t_j\rangle M'$. And we have

$$M' = M + C(\cdot, t_j) = M + C \cdot \vec{t_j}, \tag{8}$$

where $\vec{t_j}$ is the n-dimensional canonical basis vector whose j-th entry is one.

A transition sequence $\sigma = t_1 t_2 \ldots t_k$ is feasible at marking M if there exist a marking sequence $M_1, M_2, \ldots, M_k$ such that $M[t_1\rangle M_1[t_2\rangle \ldots [t_k\rangle M_k$ holds, simply denoted as $M[\sigma\rangle M_k$ or $M[\sigma\rangle$.

A reachable marking M is a marking that satisfies $M_0[\sigma\rangle M$, where σ is an arbitrary transition sequence feasible at M_0. The reachability set of (N, M_0) contains all reachable markings from M_0. The relation within the collection $R(N, M_0)$ can be represented graphically by a reachability graph. It is a directed graph whose nodes are markings in $R(N, M_0)$ and arcs are labeled with transitions of N.

A Petri net $\langle N, M_0 \rangle$ is "safe" if for all markings $M \in R(N, M_0)$ and all places $p \in P$ such that $M(p) \leq 1$ is true. It is "bounded" if there exists $k \in \mathbb{N}^+$, for all markings $M \in R(N, M_0)$ and all places $p \in P$ such that $M(p) \leq k$, otherwise it is "unbounded". In other words, a net N is said to be structurally bounded if it is bounded for any initial marking.

A labeled Petri net (LPN), denoted by $\mathcal{N}$, is a four-tuple

$$\mathcal{N} = (N, M_0, E, \ell), \tag{9}$$

where $\langle N, M_0 \rangle$ is a Petri net system, E is an alphabet (a set of labels), and $\ell : T \rightarrow E \cup \{\varepsilon\}$ 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 $T = T_o \dot{\cup} T_u$, where T_o is the set of observable transitions and T_u denote the set of unobservable transitions. The labeling function can be extended to firing sequences, i.e., $\ell(\sigma t) = \ell(\sigma)\ell(t)$, where $\sigma \in T^*$ and $t \in T$.

Given a transition sequence $\sigma \in T^*$ and a labeling function ℓ, $w = \ell(\sigma)$ is an "observed word" (or "observation"). All observed words from M_0 is denoted by:

$$\mathcal{L}(N, M_0) = \{w \in E^* | \exists \sigma \in T^*, M_0[\sigma\rangle, \ell(\sigma) = w\}, \tag{10}$$

where $\mathcal{L}(N, M_0)$ is also called the language generated from M_0. In this work, we also use $\mathcal{L}(N, M_0)$ to equivalently denote $L_{\mathcal{N}}$.

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

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

to indicate the set of firing sequences who are consistent with $w \in E^*$. Then,

$$\mathcal{C}(w) = \{M \in R(N, M_0) | (\exists \sigma : \ell(\sigma) = w), M_0[\sigma\rangle M\} \tag{11}$$

is the set of reachable markings consistent with $w \in E^*$. In plain words, given an observation w, $\mathcal{S}(w)$ is the set of sequences that may have fired, while $\mathcal{C}(w)$ is the set of markings indicating where the system may have been.

III. ROBUST DIAGNOSABILITY

A. DIAGNOSABILITY ANALYSIS

We use T_f, T_u, and T_o to denote faulty transition set, unobservable transition set, and observable transition set, respectively. The set $\Omega(T_f)$ contains all finite firing sequences that ending with the transition in T_f. Language diagnosability in LPNs is defined as follows:

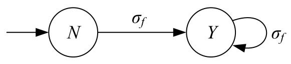

FIGURE 1. Label automaton G_l.

Definition 1 (Language Diagnosability Using LPNs [43]): Let $\mathcal{N} = (N, M_o, E, \ell)$ be a Petri net system $\langle N, M_0\rangle$ 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

$$(\forall n \in \mathbb{N})(\forall s \in \Omega(T_f))(\forall u \in T^*/s)$$
$$(\ell(su) := w_f)(\| u \| \geq n \Rightarrow C_{LPND}), \quad (12)$$

where the diagnosability condition C_{LPND} is

$$(\nexists w \in L_{\mathcal{N}})[(w_f = w) \wedge (\overline{\mathcal{S}(w)} \cap \Omega(T_f) = \emptyset)]. \quad (13)$$

Language diagnosability of a system can be simply verified using diagnosers. Let $G^{\mathcal{N}}$ denote the reachability graph of $\mathcal{N}$. The reachability diagnoser $G_d^{\mathcal{N}}$ can be calculated as $G_d^{\mathcal{N}} = Obs(G^{\mathcal{N}} || G_l, \Sigma_o)$, where $Obs(\cdot)$ and G_l denote the observer automaton and the label automaton, respectively. The label automaton G_l is shown in Fig. 1.

Technically, the reachability graph of $\mathcal{N}$ is necessary for building a reachability diagnoser. Then, we can construct an automaton G who has the same space scale as the reachability graph. These two models are obviously isomorphic, which ensures the efficiency of conversions. After that, we generate an observer on $G \parallel G_l$ with regard to Σ_o to obtain $G_d^{\mathcal{N}}$.

Definition 2 (Indeterminate Cycles Using LPNs [43]): 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 $G_d^{\mathcal{N}}$ 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 $(M_j^k, Y), (\tilde{M}_j^r, N) \in q_{d_l}$, 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 $\{M_j^k | j = 1, 2, \ldots, n, k = 1, 2, \ldots, m\}$ and $\{\tilde{M}_j^r | j = 1, 2, \ldots, n, r = 1, 2, \ldots, m'\}$ form their corresponding cycles in the generator of $G_{\mathcal{N}}$, where

$$\begin{aligned} &M_j^k[t_j\rangle M_{j+1}^k,\ j = 1, 2, \ldots, n-1,\\ &\quad k = 1, 2, \ldots, m,\\ &M_n^k[t_n\rangle M_1^{k+1},\ k = 1, 2, \ldots, m-1,\\ &\quad M_n^m[t_n\rangle M_1^1, \end{aligned} \quad (14)$$

and

$$\begin{aligned} &\tilde{M}_j^r[t_j\rangle \tilde{M}_{j+1}^r,\ j = 1, 2, \ldots, n-1,\\ &\quad r = 1, 2, \ldots, m',\\ &\tilde{M}_n^r[t_n\rangle \tilde{M}_1^{r+1},\ r = 1, 2, \ldots, m'-1,\\ &\quad \tilde{M}_n^{m'}[t_n\rangle \tilde{M}_1^1. \end{aligned} \quad (15)$$

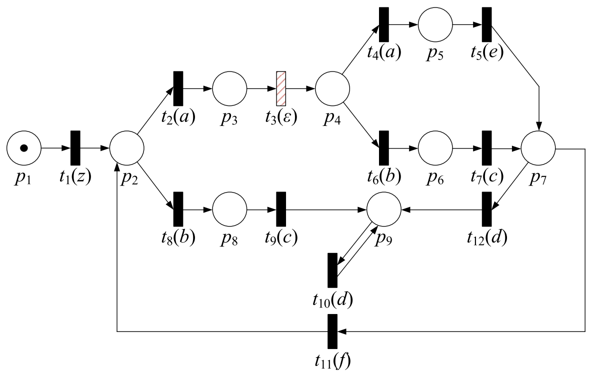

FIGURE 2. An LPN $\mathcal{N}$.

With the definition of indeterminate cycles of $G_d^{\mathcal{N}}$, the necessary and sufficient condition for language diagnosability in a LPN can be obtained.

Lemma 1 [43]: 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 $T_f = \{t_f | \ell(t_f) = \varepsilon\}$ if and only if its reachability diagnoser $G_d^{\mathcal{N}}$ has no indeterminate cycle.

Lemma 1 is a different version illustrated with LPNs, who is actually consistent with that in [16]. Thus, we omit the proof here.

B. ROBUST DIAGNOSABILITY ANALYSIS

Although in most cases, all sensors and communications can work normally such that the occurrence of any event in a system can be captured and the corresponding information can be transmitted to the diagnoser automaton, disturbances to the system do occur occasionally. Sensor malfunctions, weak circuit soldering, even extreme weather conditions or communication failure can cause loss of observations, making some events temporarily unobservable.

This loss of observations may be temporary or permanent, depending on whether the failure can be recovered to normal at some time by itself. When loss of observations occurs, traditional diagnosers will get stuck or report information incorrectly. This problem and its influence are illustrated by an example in both temporary and permanent situations. For sake of convenience, the presentation "intermittent loss of x" is equivalent to "intermittent loss of observations of transitions labeled with x".

Example 1: Consider the LPN in Fig. 2 and the firing sequence $\sigma_1 = t_1t_2t_3t_4t_5\ t_{12}t_{10}^n$, where $w_{f_1} = \ell(\sigma_1) = za\varepsilon aed^n$.

Consider the possibility that, at a particular time, the transitions labeled with a cannot be recorded as usual, i.e., the transitions t_2 and t_4 are impacted by an intermittent loss of observations. Assume that the diagnoser cannot obtain the firing information of t_2 and communication restores before t_4 fires.

For $G_d^{\mathcal{N}}$, event z, generated by t_1, is first identified, which makes $G_d^{\mathcal{N}}$ evolves to (M_1N). Since a generated by t_2 lost and transition t_3 is unobservable, a, generated by

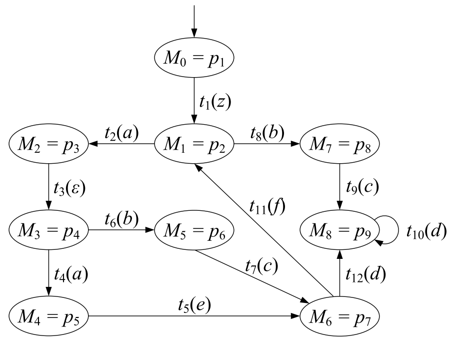

FIGURE 3. Reachability graph $G_{\mathcal{N}}$ of $\mathcal{N}$ in Fig. 2.

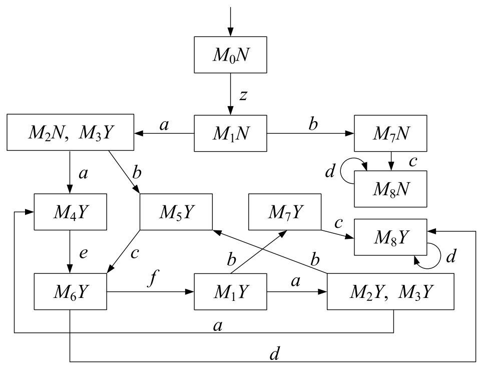

FIGURE 4. Reachability diagnoser $G_d^{\mathcal{N}}$ of $\mathcal{N}$ in Fig. 2.

t_4, is the following letter received by $G_d^{\mathcal{N}}$. It makes $G_d^{\mathcal{N}}$ evolves to (M_2N, M_3Y). Note that the next letter of $G_d^{\mathcal{N}}$ should be processed is e; however, e is not a legal successor to (M_2N, M_3Y) in $G_d^{\mathcal{N}}$. That is, $G_d^{\mathcal{N}}$ is stuck by the intermittent loss of letter a.

Now consider another sequence $\sigma_2 = t_1t_2t_3t_6t_7t_{12}t_{10}^n$, where $w_{f_2} = \ell(\sigma_2) = za\varepsilon bcd^n$. Similarly, assume the transition labeled with a is fired but not recorded for any reason. After identifying z, $G_d^{\mathcal{N}}$ enters (M_1N). Then, bcd^n should be the next sequence that $G_d^{\mathcal{N}}$ got. In this case, $G_d^{\mathcal{N}}$ properly recognizes each letter individually, and it ultimately remains in (M_8N), proving that the fault transition is never fired, and $G_d^{\mathcal{N}}$ presents utterly false information regarding the occurrence of fault transitions.

A robust diagnosis system needs to be able to withstand various unknown risks. Apparently, the traditional diagnoser automata cannot cope with the random loss of observations. The model of diagnoser automata needs to be modified by taking the loss into consideration. Fortunately, an operation proposed in [41] named dilation helps.

Definition 3 (Language Dilation): Let $\Sigma_{ilo} \cup \Sigma_{nilo} \cup \Sigma_{uo}$ be partitions of Σ, where Σ_{ilo} is the set of transitions influenced by intermittent loss of observations, Σ_{nilo} is the set of other observable transitions not influenced by intermittent loss of observations and Σ_{uo} is the set unobservable transitions. Let $\Sigma'_{ilo} = \{\sigma' : \sigma \in \Sigma_{ilo}\}$ and $\Sigma_{dil} = \Sigma \cup \Sigma'_{ilo}$. The dilation operation is defined as:

$$D : \Sigma^* \to 2^{(\Sigma_{dil})^*}, \tag{16}$$

where

$$D(\varepsilon) = \{\varepsilon\},$$
$$D(\sigma) = \begin{cases} \{\sigma\}, & if\ \sigma \in \Sigma \setminus \Sigma_{ilo}, \\ \{\sigma, \sigma'\}, & if\ \sigma \in \Sigma_{ilo}, \end{cases}$$
$$D(s\sigma) = D(s)D(\sigma),\ for\ s \in \Sigma^*, \sigma \in \Sigma. \tag{17}$$

The dilation operation D can be also executed in languages, i.e.,

$$D(L) = \bigcup_{s \in L} D(s). \tag{18}$$

Definition 4 (Dilated Automata): A dilated automaton G_{dil} is a five-tuple

$$G_{dil} = (Q, \Sigma_{dil}, f_{dil}, q_0, Q_m) \tag{19}$$

obtained from G, where $\Sigma_{dil} = \Sigma \cup \Sigma'_{ilo}$ and $f_{dil}(q, \sigma_{dil}) = f(q, \sigma)$.

Remark 1: Let G_{dil} be the dilated automaton obtained from G. We have

$$D(L) = L_{dil} = \mathcal{L}(G_{dil}). \tag{20}$$

Similaryly, we give the definition of a dilated LPN as follows.

Definition 5 (Dilated LPNs): A dilated LPN is a four-tuple

$$\mathcal{N}_{dil} = (N_{dil}, M_0, E_{dil}, \ell_{dil}), \tag{21}$$

where $\langle N_{dil}, M_0\rangle$ is a dilated Petri net, $E_{dil} = E \cup \ell(T'_{ilo})$, and ℓ_{dil} is a mapping: $T_{dil} \to E_{dil} \cup \{\varepsilon\}$.

A dilated LPN has basically the same structure as the original LPN. Unobservable transitions are added into the set of transitions, which represent the transitions subject to intermittent loss of observations. The alphabet set is expanded with letters on those unobservable transitions and so is the labeling function. In a word, a dilation operation on an LPN makes pairs of transitions appear, thus taking into account intermittent observation losses.

With Definition 5, the language robust diagnosability in LPNs can be defined as follows:

Definition 6 (Language Robust Diagnosability Using LPNs): Given an LPN $\mathcal{N} = (N, M_o, E, \ell)$, $\langle N, M_0\rangle$ 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

$$(\forall n \in \mathbb{N})(\forall s \in \Omega(T_f))(\forall u \in T^*/s)$$
$$\times(w_f := \ell(su))(\| u \| \geq n \Rightarrow C_{RD}), \tag{22}$$

where the robust diagnosability condition C_{RD} is

$$(\nexists w \in L_{\mathcal{N}})[(\ell(\mathcal{S}(D(w_f))) = \ell(\mathcal{S}(D(w))))$$
$$\wedge(\overline{\mathcal{S}(w)} \cap \Omega(T_f) = \emptyset)]. \tag{23}$$

Remark 2: If $T_{ilo} = \emptyset$, then $D(w_f) = \{w_f\}$ and $D(w) = \{w\}$. In this case, Definition 6 becomes the same as that of the traditional language diagnosability in Definition 1.

The identical diagnoser known as a robust reachability diagnoser, represented as $G_{dil,d}^{\mathcal{N}}$, can likewise be used to verify linguistic robust diagnosability using LPNs. We start by outlining what a comprehensive diagnoser is.

Definition 7 (Robust Diagnoser Automata): A robust diagnoser automaton is a DFA

$$G_{dil,d} = Obs(G_{dil} \parallel G_l, \Sigma_o). \tag{24}$$

If a system is modeled with an LPN, we can easily obtain its robust reachability diagnoser if we dilate its reachability graph and apply (24). The construction process of robust reachability diagnoser $G^{\mathcal{N}}_{dil,d}$ is almost the same as that of traditional reachability diagnoser $G^{\mathcal{N}}_d$ as we mentioned previously. The only difference is that the input of the former is a dilated LPN $\mathcal{N}_{dil}$.

Due to the dilation operation, unobservable transitions may be added to an LPN and its reachability graph, which may make the observable indeterminate cycle in a reachability diagnoser become unobservable. The indeterminate cycle formed with unobservable events between uncertain states are called implicit indeterminate cycle. Note that implicit cycles are not necessary to be indeterminate, e.g., cycles on certain states cannot be counted as they do not give any ambiguity to robust diagnosability.

Furthermore, indeterminate cycles that satisfy only Definition 2 are called explicit indeterminate cycles, whose existence makes the language not diagnosable. The term ''explicit'' emphasizes the observable nature of indeterminate cycles.

With the definitions of robust reachability diagnosers and implicit indeterminate cycles, similarly to Lemma 1, we now present a method to verify language robust diagnosability, which is also a necessary and sufficient condition for language robust diagnosability.

Lemma 2 [43]: 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 $G^{\mathcal{N}}_{dil,d}$ has no explicit or implicit indeterminate cycle.

The proof of Lemma 2 is similar to the proof of Theorem 2 in [43]. Thus, it is omitted here.

IV. ROBUST DIAGNOSABILITY ANALYSIS USING BRG

A. BASIS REACHABILITY GRAPHS

We have introduced the robust diagnosability analysis of LPN models by their reachability graphs. However, man-made systems are more sophiscated in practice. As the complexity of the system model increases, the spatial scale of the reachability graph tends to increase exponentially. Enumerating all markings to perform language diagnosability analysis consumes a lot of time and computing power. We now try to use a more compact reachability graph to provide a more efficent method for language robust diagnosability analysis.

Definition 8 (Explanations and e-vectors): Given a marking M and an observable transition $t \in T_o$, the set of explanations of t at M is defined as

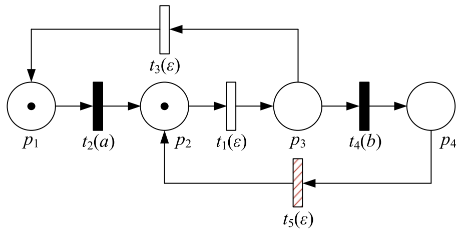

FIGURE 5. An LPN $\mathcal{N}$.

$$\Sigma(M, t) = \{\sigma \in T_u^* | M[\sigma\rangle M', M' \geq Pre(\cdot, t)\}, \tag{25}$$

and the set of e-vectors (or explanation vectors) is defined as

$$Y(M, t) = \pi(\Sigma(M, t)). \tag{26}$$

Elements in $Y(M, t)$ are the firing vectors that are associated with the explanations.

In plain words, $\Sigma(M, t)$ is the set of transition sequences who are unobservable and enable t at M. To construct a BRG, we only concern about the transition sequences that have the minimal firing vector. Firing vectors of these sequences are called minimal e-vectors.

Definition 9 (Minimal Explanations and Minimal e-Vectors): Given a marking M and an observable transition $t \in T_o$, the set of minimal explanations of t at M is defined as

$$\Sigma_{min}(M, t) = \{\sigma \in \Sigma(M, t) | \nexists \sigma' \in \Sigma(M, t) : \pi(\sigma') \lneqq \pi(\sigma)\}, \tag{27}$$

and the set of minimal e-vectors (or explanation vectors) is defined as

$$Y_{min}(M, t) = \pi(\Sigma_{min}(M, t)). \tag{28}$$

With the help of the notions of minimal explanations and minimal e-vectors, we can now define the basis marking.

Definition 10 (Basis Markings): Given an LPN $\mathcal{N} = (N, M_o, E, \ell)$. The set of basis markings $\mathcal{M}_B$ is a subset of $R(N, M_0)$ satisfying

1) *$M_0 \in \mathcal{M}_B$,*
2) *$\forall M \in \mathcal{M}_B, \forall t \in T_o, \forall y_u \in Y_{min}(M, t)$, it holds $M' \in \mathcal{M}_B$, where*

$$M' = M + C_u \cdot y_u + C(\cdot, t). \tag{29}$$

In simple words, the set of basis markings have two kinds of elements: initial marking M_0 and markings reachable from M_0 by firing each observable transition together with its minimal explanation. All the basis markings can be iteratively computed from M_0. In the intermediate process, markings obtained by firing unobservable transitions are no longer considered.

Example 2: Consider the LPN $\mathcal{N}$ in Fig. 5, where $M_0 = p_1 + p_2$. Transitions t_2 and t_4 are observable, labeled with a and b, respectively. The faulty transition set is $T_f = \{t_5\}$. The reachability graph $G_{\mathcal{N}}$ is shown in Fig. 6. By Definitions 8 and 9, we have $\Sigma(M_0, t_2) = \{\varepsilon, t_1, t_1t_3\}$, $\Sigma_{min}(M_0, t_2) = \{\varepsilon\}$, and $Y_{min} = \{\vec{0}\}$. Let $M = p_2 + p_4$. We have $\Sigma(M, t_2) =$

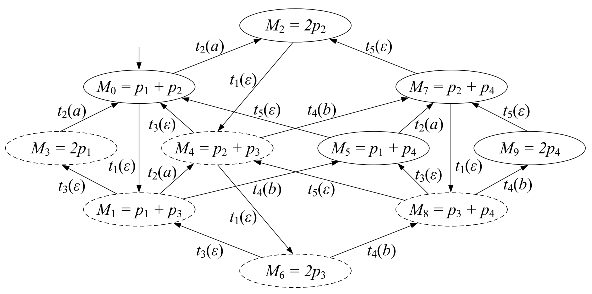

FIGURE 6. Reachability graph of $\mathcal{N}$ in Fig. 5.

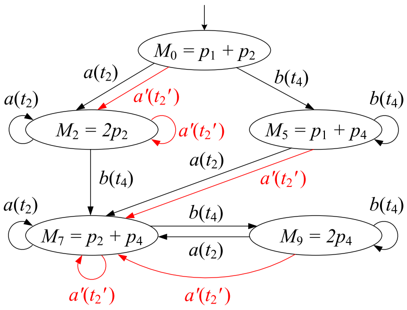

FIGURE 7. BRG of $\mathcal{N}$ in Fig. 5.

$\{t_1t_3, t_5t_1t_3\}$, $\Sigma_{min}(M, t_2) = \{t_1t_3\}$, *and* $Y_{min} = \{[1\ 1\ 0]^T\}$. *The basis markings of* $\mathcal{N}$ *are* M_0, M_2, M_5, M_7, *and* M_9. *For sake of a clear presentation, we depict the non-basis markings in circles with dashed lines.*

Based on Definition 10, an iterative algorithm is designed to compute basis markings and construct a BRG, witch can be found in [24].

Example 3: Fig. 7 shows the BRG of LPN $\mathcal{N}$ *in Example 2. Apparently, the number of states has been reduced from 10 to 5. Note that the transitions appended after the event labels are not generated by the construction of the BRG. They appear here only to show the source of each event label clearly, which does not affect the deterministic property of the BRG.*

Definition 11 (Unobservable Reach of M): Given an LPN $\mathcal{N} = (N, M_o, E, \ell)$ *and a marking* $M \in R(N, M_0)$, *we define the unobservable reach of M as*

$$\mathcal{U}(M) = \{M' \in \mathbb{N}^m | \exists \sigma_u \in T_u^* : M[\sigma_u\rangle M'\}. \tag{30}$$

Corollary 1: Let $\mathcal{N} = (N, M_o, E, \ell)$ *be an LPN whose unobservable subnet is acyclic. For all* $w \in \mathcal{L}(N, M_0)$, *we have*

$$\begin{aligned}\mathcal{C}(w) &= \bigcup_{M_b \in \mathcal{M}_b(w)} \mathcal{U}(M_b) \\ &= \bigcup_{M_b \in \mathcal{M}_b(w)} \{M \in \mathbb{N}^m | \exists y_u \in \mathbb{N}^{n_u} : \\ &\quad M = M_b + C_u \cdot y_u\}. \end{aligned} \tag{31}$$

In plain words, given a marking M and a basis marking M_b consistent with w, marking M is also consistent with w if and only if M belongs to the unobservable reach of M_b. Meanwhile, marking M belonging to $\mathcal{U}(M)$ means that $M = M_b + C_u \cdot y_u$ has a non-negative integer solution y_u.

Example 4: Consider the LPN in Fig. 5. Unobservable reaches of basis markings in Fig. 7 are shown in TABLE 1 below.

TABLE 1. Unobservable reaches of the basis markings in Fig. 7.

Basis Markings	$\mathcal{U}(M)$
M_0	$\{M_0, M_1, M_3\}$
M_2	$\{M_0, M_1, M_2, M_3, M_4, M_6\}$
M_5	$\{M_0, M_1, M_3, M_5\}$
M_7	$\{M_0, M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8\}$
M_9	$\{M_0, M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9\}$

B. BASIS REACHABILITY DIAGNOSERS

It has been verified that a BRG can significantly reduce the scale of a reachability graph. We introduce a new diagnosis model, i.e., basis reachability diagnosers that only consider basis markings and the notion of unobservable reach. Based on the intuitive definition of language diagnosis, a reachability diagnoser can be constructed following the steps below:

STEP 1. Start from the initial marking M_0 and sign it as "N".

STEP 2. Traverse the whole BRG. Find an unmarked marking and sign it with the help of a **signing procedure** (which will be stated later). If there already exists the same signed marking, go to STEP 3.

STEP 3. Draw a directed arc from the precursor node to the current node and label it with a corresponding letter in the BRG.

The subject of these steps is to check whether a fault event has occurred while traversing each transition in the BRG. Corresponding marks are made and a new graph with diagnosis information is drawn. Due to the existence of the BRG, the complexity of traversing the entire state space is much lower than the case of using the reachability graph. To present the diagnosis information, we need a signing procedure as follows:

STEP i. Sign the current node as "Y" if its precursor node is signed as "Y". Terminate the procedure.

STEP ii. Sign the current node as "U" if its precursor node is signed as "U" or there exist some specific markings in $\mathcal{U}(M)$ that can be reached by firing any faulty transition sequence, i.e., $M[\sigma_f\rangle$ holds.

STEP iii. If one of the necessary conditions for reaching the current marking from its direct precursor node is the firing of any fault transition, sign the current marking as "Y" regardless of whether or how the current marking has been signed. Terminate the procedure.

STEP iv. If the current marking is still unsigned, sign it as "N".

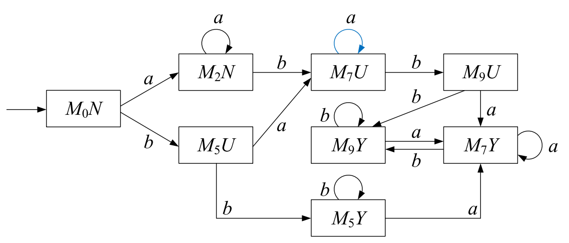

FIGURE 8. Basis reachability diagnoser of $\mathcal{N}$ in Fig. 5.

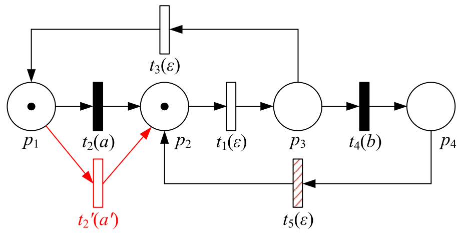

FIGURE 9. Dilated LPN $\mathcal{N}_{dil}$ of $\mathcal{N}$ in Fig. 5.

There are three kinds of signs in total: "Y" stands for "YES" representing the occurrence of faulty transitions, "N" stands for "NO" representing that none of faulty transitions has been fired, and "U" stands for "UNKNOWN" representing that it is not clear yet.

Example 5: Fig. 8 shows the basis reachability diagnoser of the BRG in Fig. 7.

Consider a transition sequence $\sigma_1 = t_2t_1t_3t_1t_2$ with $w_1 = \ell(\sigma_1) = a\varepsilon\varepsilon\varepsilon a$. We observe string aa in the basis reachability diagnoser and the system is at marking M_2. No marking is reachable from M_2 by firing any unobservable transition sequence containing faulty transition t_5, which indicates that no faulty transition has happened currently.

Now consider $\sigma_2 = t_1t_4t_2t_1t_3t_2t_1t_4$ with $w_2 = \ell(\sigma_2) = \varepsilon ba\varepsilon\varepsilon a\varepsilon b$. The system will be at marking M_9. Since there exist markings such as $M_7, M_8, \cdots \in \mathcal{U}(M_9)$ reachable from M_9 by firing unobservable transition sequences containing fault transition t_5, we cannot infer whether t_5 has fired by observing baab only.

Consider another transition sequence $\sigma_3 = t_1t_4t_5t_1t_4$ with $w_3 = \ell(\sigma_3) = \varepsilon b\varepsilon\varepsilon b$. Marking (M_5Y) reached by observing bb in the basis reachability diagnoser means that the fault transition t_5 has necessarily fired.

Now we try to re-examine the three transition sequences from the perspective of the signing procedure to verify the proposed conclusions. The detailed steps are as follows:

- $M_0[\sigma_1\rangle M_2$, where $\sigma_1 = t_2t_1t_3t_1t_2$:
  i. The precursor node M_0 is not signed as "Y".
  ii. The precursor node M_0 is not signed as "U". We have $\mathcal{U}(M_2) = \{M_0 - M_4, M_6\}$ and none of them is reachable by firing any faulty transition sequence.
  iii. M_2 is not reachable by firing any fault transition.
  iv. Still unsigned. Sign it with "N".
- $M_0[\sigma_2\rangle M_9$, where $\sigma_2 = t_1t_4t_2t_1t_3t_2t_1t_4$:
  i. The precursor node M_7 is not signed as "Y".
  ii. The precursor node M_7 is signed as "U". Sign it as "U".
  iii. M_9 is not reachable by firing any fault transition.
  iv. Keep the sign "U" unchanged.
- $M_0[\sigma_3\rangle M_5$, where $\sigma_3 = t_1t_4t_5t_1t_4$:
  i. The precursor node M_5 is not signed as "Y".
  ii. The precursor node M_5 is signed as "U". Sign it as "U".
  iii. M_5 is reachable from M_5 via t_5 definitely. Re-sign it with "Y" and terminate the procedure.

The sign obtained with the help of the signing procedure is exactly the same with that of our speculation previously. Therefore, the validity of the signing procedure is confirmed.

Different from the reachability diagnoser, the states in a basis reachability diagnoser seem to be quite simple, i.e., no states are merged during the construction of the basis reachability diagnoser. However, situations are not always like this, depending on two conditions. First is whether there exist unobservable transitions in the graph structure used to construct the diagnosis automaton. Apparently, unobservable transitions can be found in a reachability graph rather than in a BRG. That is the exact reason why compound states exist in a reachability diagnoser. Secondly, the deterministic property of the graph structure used to construct the diagnosis automaton also matters. For instance, consider the BRG in Fig. 7 and assume that all the transitions are labeled with a. Correspondingly, the first state after (M_0N) should be a compound state made up of (M_2N) and (M_5U) via a. However, regardless of whether the BRG is deterministic or not, the deterministic property will be imparted during the construction process of a basis reachability diagnoser, since the construction of an observer itself is a standard determinization procedure to convert an NFA into a DFA [8].

Note that there exists a cycle on (M_7U) in Fig. 7. According to the previous experience, this kind of cycles should be noticed, since it may cause indeterminacy and make the language lose its diagnosability. We will discuss it together with robust diagnosability in the next section.

C. ROBUST BASIS REACHABILITY DIAGNOSERS

We have successfully established a brand-new diagnosis model called basis reachability diagnosers. Usually, we have to reconsidere the robustness of this new model if intermittent loss of observations occurs.

For instance, there exists a transition sequence $\sigma = t_2t_1t_4t_4$ such that $w = \ell(\sigma) = a\varepsilon bb$. Suppose that the firing of transition t_2 cannot be observed successfully. In the basis reachability diagnoser of $\mathcal{N}$, by observing bb we infer that the system is now in (M_5Y), which means that the fault transition has been fired. However, t_5 is not fired, or at least it cannot be made certain by observing abb only. We now give the solution by the dilation operation.

Example 6: Assume that intermittent observation loss influences all transitions labeled with a. By dilation operation, $\mathcal{N}$ in Fig. 5 can be dilated and depicted in Fig. 9.

TABLE 2. Numbers of markings and basis markings.

k_1	$\lvert R(N, M'_0)\rvert$	$\lvert\mathcal{M}'_B\rvert$	k_2	$\lvert R(N, M''_0)\rvert$	$\lvert\mathcal{M}''_B\rvert$
5	84	27	5	84	13
10	364	77	10	364	23
15	969	152	15	969	33
20	2024	252	20	2024	43
25	3654	377	25	3654	53
30	5984	527	30	5984	63
35	9139	702	35	9139	73
40	13244	902	40	13244	83

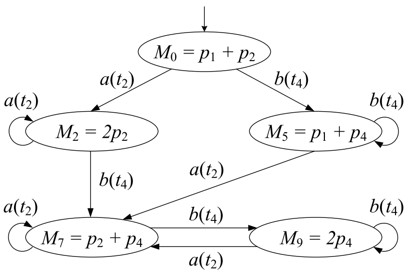

FIGURE 10. Dilated BRG $B^{\mathcal{N}}_{dil}$ of $\mathcal{N}$ in Fig. 5.

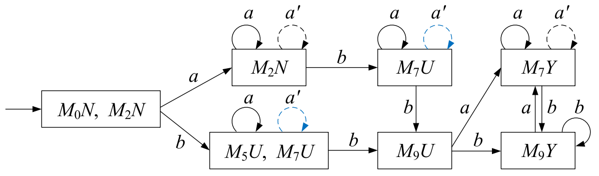

FIGURE 11. Robust basis reachability diagnoser $B^{\mathcal{N}}_{dil,d}$ of $\mathcal{N}$ in Fig. 5.

Different from a reachability graph, the BRG generated from $\mathcal{N}_{dil}$ is exactly the same as that from the original LPN, since the unobservable transitions from the dilation do not change the minimal explanation and e-vector of each observable transition. Consider $\Sigma(M_0, t_2)$ and $\Sigma(M_7, t_2)$ in Example 2 again. We have $\Sigma_{min}(M_0, t_2) = \{\varepsilon\}$, $Y_{min} = \{\vec{0}\}$, $\Sigma_{min}(M_7, t_2) = \{t_1t_3\}$, and $Y_{min} = \{[1\ 1\ 0]^T\}$. Meanwhile, no unobservable transitions are allowed to show up in a BRG. Under this circumstance, to construct a robust basis reachability diagnoser from a BRG means that the BRG needs to be modified slightly to accommodate the intermittent loss of observations.

Example 7: By dilation operation, a dilated BRG denoted as $B^{\mathcal{N}}_{dil}$ is shown in Fig. 10. Note that there should not have unobservable transitions in a BRG. We here name the $B^{\mathcal{N}}_{dil}$ a "dilated BRG" with a slight abuse of notion only for the sake of simplicity. Fig. 11 shows the diagnoser constructed with $B^{\mathcal{N}}_{dil}$ called a robust basis reachability diagnoser, denoted as $B^{\mathcal{N}}_{dil,d}$.

It is not difficult to find that unobservable transitions occur in $B^{\mathcal{N}}_{dil,d}$ and some markings are merged due to the existence of unobservable transitions in $B^{\mathcal{N}}_{dil}$, which confirms our point of view earlier. We now give the construction algorithm of a robust basis reachability diagnoser together with that of a basis reachability diagnoser as mentioned previously. The output of the algorithm depends entirely on the type of the BRG input.

Algorithm 1 Construction of a (Robust) Basis Reachability Diagnoser $B^{\mathcal{N}}_d$ (or $B^{\mathcal{N}}_{dil,d}$)

```
Input: A (dilated) BRG $B_{\mathcal{N}}$ (or $B^{\mathcal{N}}_{dil}$).
Output: A (robust) basis reachability diagnoser $B^{\mathcal{N}}_d$ (or $B^{\mathcal{N}}_{dil,d}$).
1:  define a DFA $G = (Q, \Sigma, f, q_0, Q_m)$ and initialize $Q := \emptyset$, $\Sigma := \emptyset$, $f := \emptyset$, and $Q_m := \emptyset$;
    /* $G$ is essentially $B^{\mathcal{N}}_d$ or $B^{\mathcal{N}}_{dil,d}$, depending on which diagnosis model is desired */
2:  define a one-way marking queue $MQ$ shaped like $M_i \leftarrow M_j \leftarrow \cdots \leftarrow M_n$;
    /* new markings can only be entered from the end of the queue */
3:  initialize $MQ := M_0$;
4:  sign $M_0$ with $N$;
5:  while $MQ$ is not empty do
6:      pop the head marking from $MQ$ and define it as $M_{temp}$;
7:      for all transitions $t_i$ in $B_{\mathcal{N}}$ (or $B^{\mathcal{N}}_{dil}$) such that $M_{temp}[t_i\rangle$, do
8:          push $M_{k_i}$ such that $M_{temp}[t_i\rangle M_{k_i}$ into $MQ$;
9:          execute the signing procedure on each $M_{k_i}$ and obtain $(M_{k_i}, S_{k_i})$, where $S_{k_i}$ is the diagnostic sign of $M_{k_i}$;
10:         if $M_{k_i} \in \mathcal{U}(M_{temp})$, then
11:             merge $(M_{temp}, S_{temp})$ and $(M_{k_i}, S_{k_i})$;
12:         end if
13:         if $M_{k_i} \notin \mathcal{U}(M_{temp})$, then
14:             draw an arc from $(M_{temp}, S_{temp})$ to $(M_{k_i}, S_{k_i})$ and label it with $\ell(t_i)$;
15:             if the arc already exists then
16:                 delete $M_{k_i}$ from $MQ$;
17:             end if
18:             let $Q = Q \cup (M_{temp}, S_{temp})$;
19:             let $\Sigma = \Sigma \cup \ell(t_i)$;
20:         end if
21:         let $f = f \cup \{[(M_{temp}, S_{temp}), \ell(t_i), (M_{k_i}, S_{k_i})]\}$;
22:     end for
23: end while
24: let the state containing "$M_0N$" be $q_0$;
25: return $B^{\mathcal{N}}_d$ (or $B^{\mathcal{N}}_{dil,d}$).
```

We now explain how Algorithm 1 works. To construct a basis reachability diagnoser (or a robust basis reachability diagnoser), we need to prepare a BRG (or dilated BRG) as the input. Initialize a plain DFA as the expected output. Define a queue composed of markings. The queue needs to meet the following conditions: 1) elements can be popped from the head of the queue; 2) elements at any position can be deleted by a certain index without affecting the relative order of the remaining elements; 3) new elements can only be inserted from the end of the queue. Initialize the queue with the initial

TABLE 3. Comparison of four diagnosis models.

	RD	RRD	BRD	RBRD
constructed from	reachability graph	dilated reachability graph	BRG	dilated BRG
$\mathcal{O}(n)$ worst	exponential	exponential	exponential	exponential
$\mathcal{O}(n)$ best	polynomial	polynomial	linear	linear
on-line diagnosis	✓	✓	✓	✓
off-line diagnosis	✓	✓	✓	✓
robustness	×	✓	×	✓

RD: reachability diagnoser; RRD: robust reachability diagnoser;
BRD: basis reachability diagnoser; RBRD: robust basis reachability diagnoser.

marking in the input graph structure. Next, we traverse the entire BRG by pushing the unchecked markings into the queue and popping the checked markings out of the queue.

As shown in lines 5 to 23 of Algorithm 1, a marking is popped up and set as M_{temp}. We first push each marking M_{k_i} reachable from M_{temp} via a single transition t_i into the queue. By executing the signing procedure on M_{k_i}, we can obtain (M_{k_i}, S_{k_i}). Note that t_i can be unobservable if the input graph is a dilated BRG. Thus, check if M_{k_i} belongs to the unobservable reach of M_{temp}. If so, merge the two states into a new one. Otherwise, establish a transition arc from (M_{temp}, S_{temp}) to (M_{k_i}, S_{k_i}) if it does not exist yet. Label it with the letter corresponding to t_i from the input graph. If the transition already exists, delete M_{k_i} from the queue just in case that the transition is depicted repeatedly, which will make the algorithm fall into an infinite loop. Meanwhile, add the signed state and transition letter to their corresponding sets of the output DFA. Repeat the entire process until the queue is emptied, and we eventually obtain a reachability diagnoser (or robust basis reachability diagnoser).

Remember we have observed an explicit cycle on an unknown state in the basis reachability diagnoser in Fig. 8. As shown in Fig. 11, not only explicit cycles but also implicit cycles appear. For instance, there exist cycles labeled with a' on (M_5U, M_7U) and (M_7U).

Theorem 1: Let $L_{\mathcal{N}}$ denote a live language that is generated by an LPN $\mathcal{N}$. For given dilation function $D : \Sigma^* \rightarrow 2^{(\Sigma_{dil})^*}$ and fault enent $t_f \in T_u$, $L_{\mathcal{N}}$ is robustly diagnosable if and only if there is no explicit or implicit cycles on markings signed as "U" in the robust basis reachability diagnoser $B^{\mathcal{N}}_{dil,d}$.

Proof (Necessity): Assume by contrapositive that there exists an explicit or implicit cycle on a "U" marking in a robust basis reachability diagnoser $B^{\mathcal{N}}_{dil,d}$. According to Definition 6, the language is not robustly diagnosable as long as there exists one normal word w such that the set of markings consistent with $P_o(D(w))$ is the same with that of $P_o(D(w_f))$, where $\mathcal{S}(w_f)$ contains a fault transition. Suppose that there exists an explicit or implicit cycle in the BRG. It is possible for both the normal sequence and faulty sequence to loop endlessly in the cycle eventually.

From the perspective of $B^{\mathcal{N}}_{dil,d}$, two words lead to the same marking signed with "U" either explicitly or implicitly. That is to say, we have found at least a normal word w, whose firing sequence satisfies the previous condition, i.e., C_{RD} in Definition 6. Hence, $L_{\mathcal{N}}$ is not robustly diagnosable.

(Sufficiency) Assume that, there exists no explicit or implicit cycle on "U" markings in a robust basis reachability diagnoser $B^{\mathcal{N}}_{dil,d}$. In other words, it is not possible anymore to satisfy the arbitrary length requirement in C_{RD} due to the existence of the cycle in $B^{\mathcal{N}}_{dil,d}$. Both the normal sequence and faulty sequence must provide letters continuously. Meanwhile, a dilated BRG is always live and bounded. To avoid forming cycles on "U" markings and satisfy the requirement of C_{RD}, two transitions will eventually lead the system to two markings, i.e., "N" and "Y" via two paths. In this approach, the firing of a fault transition can be always inferred, which means that $L_{\mathcal{N}}$ is robustly diagnosable.

By Algorithm 1, $B^{\mathcal{N}}_{d}$ or $B^{\mathcal{N}}_{dil,d}$ can be easily constructed. Compared with a traditional diagnosis model generated from a reachability graph, basis reachability diagnosers and robust reachability basis diagnosers have states no more than the former. In general, since only basis markings are considered, the scale of diagnosers generated from a BRG is much smaller than that of traditional diagnosers. In other words, using a BRG instead of a reachability graph in practice can significantly reduce the scale and complexity of diagnosis models.

V. A NUMERICAL EXPERIMENT

Example 8: Consider again the LPN $\mathcal{N}$ in Fig. 5. Suppose that $M_0' = k_1p_1 + p_2$ and $M_0'' = p_1 + k_2p_2$, i.e., we parameterize the number of tokens in p_1 and p_2. The numbers of markings and basis markings can be computed with the MATLAB toolbox provided in [44] as TABLE 2 shows.

Fig. 12 shows the growth of $|R(N, M_0)|$ and $|\mathcal{M}_B|$ along with the changes of k. Obviously, using a BRG can tremendously reduce the number of markings compared with a reachability graph, at least an order of magnitude.

So far, to analyze language robust diagnosability, we have introduced four diagnosis models — reachability diagnosers, robust reachability diagnosers, basis reachability diagnosers, and robust basis reachability diagnosers. Four diagnosis models have similar functions but different characteristics, which are summarized in TABLE 3.

It is worth noting that $\mathcal{O}(n)$ in TABLE 3 represents the complexity of generating the graph necessary for constructing a diagnosis model, which often depends on some parameters in a system that determine the scale of its state space. For instance, we must first generate the reachability graph of an LPN to build a reachability diagnoser. In the worst case, the

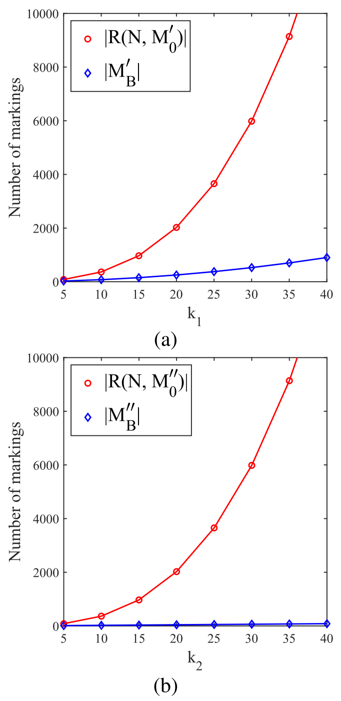

FIGURE 12. The growth of $|R(N, M_0)|$ and $|\mathcal{M}_B|$ with (a) k_1 and (b) k_2.

system size is exponentially related to the number of states in the reachability graph.

VI. CONCLUSION

We look into the problem of diagnosability analysis of DESs in this paper. A solution is provided for the problem of robust diagnosability against intermittent loss of observations.

First we review the definition of language diagnosability and robust diagnosability using LPNs. To verify language diagnosability using LPNs, we establish (robust) reachability diagnosers with the help of reachability graphs (and the dilation operation). We explain the notion of (implicit) indeterminate cycles and introduce a necessary and sufficient condition for language (robust) diagnosability.

The construction complexity of (robust) reachability diagnosers to verify the diagnosability of LPNs is not computationally competitive. A BRG, which is a condensed manner to express reachable markings and their relationship, is presented to increase the effectiveness of diagnosability analysis. We introduce the principle and construction algorithm of a BRG. A few steps are given, through which a basis reachability diagnoser or a robust basis reachability diagnoser can be established. The condition of language robust diagnosability is modified with the help of a robust basis reachability diagnoser. Finally, we give a numerical example to demonstrate the advantage of robust diagnosability analysis using a BRG instead of a reachability graph.

All of the work is performed on the basis of a centralized diagnoser. Future research will focus on decentralized architecture. Additionally, we assume that the intermittent observation losses of specific kinds of transitions are already known before the analysis in this paper. Whether it is possible to infer in real time that which transition is subject to intermittent loss of observations by observing the behavior of a system is worthy of consideration.

REFERENCES

[1] M. Zhou and F. DiCesare, *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Boston, MA, USA: Springer, 2012, vol. 204.

[2] H. Hu and M. Zhou, "A Petri net-based discrete-event control of automated manufacturing systems with assembly operations," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 2, pp. 513–524, Mar. 2015.

[3] C. Gu, Z. Li, N. Wu, M. Khalgui, T. Qu, and A. Al-Ahmari, "Improved multi-step look-ahead control policies for automated manufacturing systems," *IEEE Access*, vol. 6, pp. 68824–68838, 2018.

[4] J. Zhao, Y.-L. Chen, Z. Chen, F. Lin, C. Wang, and H. Zhang, "Modeling and control of discrete event systems using finite state machines with variables and their applications in power grids," *Syst. Control Lett.*, vol. 61, no. 1, pp. 212–222, Jan. 2012.

[5] L. Cheng and M. A. Duran, "Logistics for world-wide crude oil transportation using discrete event simulation and optimal control," *Comput. Chem. Eng.*, vol. 28, nos. 6–7, pp. 897–911, Jun. 2004.

[6] L. Rozé and M.-O. Cordier, "Diagnosing discrete-event systems: Extending the 'diagnoser approach' to deal with telecommunication networks," *Discrete Event Dyn. Syst.*, vol. 12, no. 1, pp. 43–81, Jan. 2002.

[7] P. J. G. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proc. IEEE*, vol. 77, no. 1, pp. 81–98, Jan. 1989.

[8] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed. New York, NY, USA: Springer, 2009.

[9] R. Ammour, E. Leclercq, E. Sanlaville, and D. Lefebvre, "Fault prognosis of timed stochastic discrete event systems with bounded estimation error," *Automatica*, vol. 82, pp. 35–41, Aug. 2017.

[10] X. Wang, C. Mahulea, and M. Silva, "Diagnosis of time Petri nets using fault diagnosis graph," *IEEE Trans. Autom. Control*, vol. 60, no. 9, pp. 2321–2335, Sep. 2015.

[11] R. Ammour, E. Leclercq, E. Sanlaville, and D. Lefebvre, "Faults prognosis using partially observed stochastic Petri-nets: An incremental approach," *Discrete Event Dyn. Syst.*, vol. 28, no. 2, pp. 247–267, Jun. 2018.

[12] 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., Syst.*, vol. 49, no. 2, pp. 333–345, Feb. 2017.

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

[14] M. P. Cabasino, A. Giua, and C. Seatzu, "Diagnosability of bounded Petri nets," in *Proc. 48th IEEE Conf. Decision Control 28th Chin. Control Conf.*, Shanghai, China, Dec. 2009, pp. 1254–1260.

[15] 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, Dec. 2012.

[16] 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, Sep. 1995.

[17] 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, Mar. 1996.

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

[19] Z. He, Z. W. Li, and A. Giua, "Performance optimization for timed weighted marked graphs under infinite server semantics," *IEEE Trans. Autom. Control*, vol. 63, no. 8, pp. 2573–2580, Aug. 2017.

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

[21] Z. W. Li and M. C. Zhou, "Control of elementary and dependent siphons in Petri nets and their application," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 1, pp. 133–148, Dec. 2008.

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

[23] Z. Ma, Y. Tong, Z. Li, and A. Giua, "Basis marking representation of Petri net reachability spaces and its application to the reachability problem," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1078–1093, Mar. 2016.

[24] 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, Jun. 2017.

[25] L. Li, F. Basile, and Z. Li, "An approach to improve permissiveness of supervisors for GMECs in time Petri net systems," *IEEE Trans. Autom. Control*, vol. 65, no. 1, pp. 237–251, Jan. 2019.

[26] 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, Sep. 2010.

[27] Y. Hu, Z. Ma, and Z. Li, "Design of supervisors for active diagnosis in discrete event systems," *IEEE Trans. Autom. Control*, vol. 65, no. 12, pp. 5159–5172, Dec. 2020.

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

[29] D. Wang, X. Wang, and Z. Li, "Nonblocking supervisory control of state-tree structures with conditional-preemption matrices," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 3744–3756, Jun. 2019.

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

[31] Y. Tong, Z. Li, C. Seatzu, and A. Giua, "Current-state opacity enforcement in discrete event systems under incomparable observations," *Discrete Event Dyn. Syst.*, vol. 28, no. 2, pp. 161–182, Jun. 2018.

[32] X. Y. Cong, M. P. Fanti, A. M. Mangini, and Z. W. Li, "On-line verification of current-state opacity by Petri nets and integer linear programming," *Automatica*, vol. 94, pp. 205–213, 2018.

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

[34] M. P. Cabasino, A. Giua, and C. Seatzu, "Diagnosability analysis of an ABS system modeled using Petri nets," *IFAC Proc. Volumes*, vol. 45, no. 20, pp. 842–847, Jan. 2012.

[35] Z. Ma, Y. Xiang, and Z. Li, "Marking diagnosis in labeled Petri nets using basis diagnosers," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, Jeju Island, Republic of Korea, Dec. 2020, pp. 4479–4484.

[36] M. Cabasino, A. Giua, and C. Seatzu, "Diagnosability of discrete-event systems using labeled Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 1, pp. 144–153, Jan. 2014.

[37] 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, Mar. 2016.

[38] J. C. Basilio and S. Lafortune, "Robust codiagnosability of discrete event systems," in *Proc. Amer. Control Conf.*, St. Louis, MO, USA, 2009, pp. 2202–2209.

[39] S. T. S. Lima, J. C. Basilio, S. Lafortune, and M. V. Moreira, "Robust diagnosis of discrete-event systems subject to permanent sensor failures," *IFAC Proc. Volumes*, vol. 43, no. 12, pp. 90–97, 2010.

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

[41] 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.

[42] K. R. Rohloff, "Sensor failure tolerant supervisory control," in *Proc. 44th IEEE Conf. Decis. Control, Eur. Control Conf.*, Seville, Spain, Dec. 2005, pp. 3493–3498.

[43] S. Li, M. Uzam, L. Yin, Z. Zhong, L. Zheng, and N. Wu, "Robust diagnosability analysis of discrete event systems using labeled Petri nets," *IEEE Access*, vol. 9, pp. 163504–163515, 2021.

[44] S. Liu, Y. Tong, C. Seatzu, and A. Giua, "PetriBaR: A MATLAB toolbox for Petri nets implementing basis reachability approaches," *IFAC-PapersOnLine*, vol. 51, no. 7, pp. 316–322, 2018.

SHIQI LI received the B.S. degree in automation from Xidian University, Xi'an, China, in 2018, where he is currently pursuing the M.E. degree in control engineering. His current research interests include discrete event systems, labeled Petri nets, and diagnosability analysis.

SIAN ZHOU received the B.E. degree in electronic science and technology from the East China University of Technology, Fuzhou, China, in 2012, and the M.E. degree in computer technology from Northwest Normal University, Lanzhou, China, in 2015. He is currently pursuing the Ph.D. degree in computer technology and its applications with the Macau University of Science and Technology, Macau. His current research interests include the quantification of opacity and fault diagnosis in discrete event systems.

LI YIN received the B.E. degree in remote sensing from Wuhan University, Wuhan, China, in 2007, the M.S. degree in GIS from the CNNC Beijing Research Institute of Uranium Geology, Beijing, in 2014, and the Ph.D. degree from the Macau University of Science and Technology, Macau, China. He is currently an Assistant Professor with the Institute of Systems Engineering, Macau University of Science and Technology. His research interests include discrete-event systems and fault-tolerant dynamic systems with applications to manufacturing.

RONGTIAN JIANG is one of the main Architects of Hitachi Elevator E-Cloud big data system, as the Leader of the realization of cloud construction, data collection, AI analysis, and elevator diagnosis. He has more than ten years of experience in the field of big data development. He has rich experience in big data planning, platform construction, and data application projects of the Internet of Things enterprises. His research interests include collection, cleaning, storage, calculation, and display of big data.