Diagnosability Enforcement in Labeled Petri Nets Using Supervisory Control

Yihui Hu, Ziyue Ma, Zhiwu Li*, Alessandro Giua

July 7, 2021

Abstract

In this article, we deal with the active diagnosis problem in labeled Petri nets by developing a supervisor for a plant such that the closed-loop system is diagnosable. Since control actions may introduce deadlocks even if an original plant is deadlock-free, we first generalize the classical notion of diagnosability in labeled Petri nets to the nets that may contain potential deadlocks. To avoid enumerating all reachable markings of a plant, we develop a structure called *quiescent basis reachability graph*, and accordingly propose a structure named *Q-diagnoser* to verify the diagnosability of a net. We prove that a plant is diagnosable if and only if there does not exist any indeterminate cycle in its Q-diagnoser. Finally, for an undiagnosable plant, we introduce a *diagnosability enforcing supervisor* to enforce the diagnosability by trimming a Q-diagnoser. Moreover, our approach guarantees that the closed-loop system cannot reach a dead marking unless a fault transition has fired.

Published as:

[Y. Hu, Z. Ma, Z. Li, A. Giua, "Diagnosability Enforcement in Labeled Petri Nets Using Supervisory Control", *Automatica*, **2021**, 131, Article 109776.]
DOI: 10.1016/j.automatica.2021.109776

*Corresponding Author

Yihui Hu is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China, and also with Department of Electrical and Electronic Engineering, University of Cagliari, 09124 Cagliari, Italy (e-mail: huyihui@stu.xidian.edu.cn).

Ziyue Ma is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China (e-mail: maziyue@xidian.edu.cn).

Zhiwu Li is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China, and also with Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau, China (e-mail: zhwli@xidian.edu.cn).

Alessandro Giua is with Department of Electrical and Electronic Engineering, University of Cagliari, 09124 Cagliari, Italy (e-mail: giua@unica.it)

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61703321, 61873342, the National Key R&D Program of China under Grant No. 2018YFB1700104, the National Natural Science Foundation of Shaanxi Province under Grant Nos. 2019JQ-022, 2019JM-049, the Fundamental Research Funds for the Central Universities under Grant Nos. JB210413, JB190407 and the Science and Technology Development Fund, MSAR, under Grant No. 0012/2019/A1.

1 Introduction

Fault diagnosis for discrete event systems (DESs) [11] has been extensively studied in recent years. The aim of *diagnosis* [8, 14, 25, 39, 44] is to infer if some faults have occurred in a plant by observing the events it generates. A plant is said to be *diagnosable* [41] if the occurrence of faults in it can be detected within a finite number of steps. For the sake of safety and reliability, a plant should necessarily be diagnosable to ensure that any fault can be detected and repaired in time. For a diagnosable plant, a *diagnostic agent* can be designed to perform online diagnosis. On the other hand, when a plant is not diagnosable, some corrective action should be taken: this is usually called *diagnosability enforcement*.

In the literature, there are many results on the verification of diagnosability in the framework of automata. Sampath *et al.* [41] verify diagnosability using a structure called *diagnoser*, and show that a plant is diagnosable if and only if there does not exist any *indeterminate cycle* in its diagnoser. Since the complexity of a diagnoser is exponential with respect to the number of states in the corresponding plant, some researchers develop a new structure called *verifier* [23, 31, 45, 48] to test diagnosability in polynomial time. On the other hand, Petri nets are a mathematical tool in which structural analysis and abstraction techniques can be used to reduce the computation complexity of analysis and control [8, 37, 38]. As a result, many researchers focus on the diagnosability verification by using Petri nets [1, 6, 9, 24, 34, 33, 47]. Basile *et al.* [1] present a necessary and sufficient condition for diagnosability by solving an *integer linear programming problem* (ILPP). In [9], Cabasino *et al.* introduce *modified basis reachability graph* and *basis reachability diagnoser* to verify diagnosability of a net without enumerating all reachable markings. Similarly, Jiroveanu *et al.* [24] propose an automaton called *ROF-automaton*, which provides a compact representation of the state space. Cabasino *et al.* [6] show that the diagnosability of an unbounded Petri net can be determined by analyzing the coverability graph of the so-called verifier net. Recently, Ran *et al.* [36] also explore diagnosability verification in decentralized Petri net models. A different model is considered by Ramírez-Treviño *et al.* in [34, 33], where the diagnosability problem in *interpreted Petri nets* (IPNs) is studied. In their framework, the considered plant net has an output function that associates an output vector to each marking. In [34], the authors introduce a notion called *input-output diagnosability* and provide sufficient structural conditions to verify it on the premise that any T-semiflow must contain all risky transitions. This work is further extended in [33] where a concept called *relative distance* is used to present a new characterization providing sufficient conditions for diagnosability, and polynomial algorithms are proposed to determine the diagnosability.

For a plant that is not diagnosable, there are mainly two types of approaches in the literature to enforce diagnosability. The first type of approaches assumes that it may be possible to change the observation structure of a plant. In the framework of automata, Cassez *et al.* [12] present a dynamic strategy for sensor activation to guarantee diagnosability. Moreover, Wang *et al.* [43] propose a sensor activation policy to enforce diagnosability with a minimal cost. For Petri nets, Basile *et al.* [2] develop an integer linear programming to find a minimal set of sensors that makes a net system k-diagnosable. Cabasino *et al.* [10] introduce a new labeling function making a system diagnosable using *verifier net*. To circumvent the state explosion problem that the method in [10] may potentially encounter, Ran *et al.* [35] develop a new structure called the *unfolded verifier*, and determine a new labeling function to enforce the diagnosability with a minimal cost.

Nevertheless, enforcing diagnosability by modifying the observation structure usually requires to implement additional sensors into a plant. This may not always be possible, since the new sensors may be too expensive or technically unfeasible. Therefore, in such a case the second type of approaches, called *active diagnosis*, is preferable [5, 13, 15, 20, 22, 40, 46]. In an active diagnosis scheme, a supervisor is designed to forbid all undiagnosable evolutions of a plant, thus ensuring that the closed-loop system remains diagnosable. In the literature, active diagnosis has been widely studied in automata where faults are defined on events [5, 13, 22, 40, 42, 46] and states [15]. However, there are few works that deal with active diagnosis in Petri nets as far as we know. Based on a notion called a *regulation circuit controller*, an approach is presented in [19] to enforce diagnosability in *interpreted Petri nets*. Moreover, the nets considered in [19] are *live, binary* and *event-detectable*. Differently from the framework in [19], this paper studies the active diagnosis problem in labeled Petri nets (LPNs).

The active diagnosis problem in LPNs can be solved by computing the reachability graph of a plant and using the graph to design a supervisor by means of the automaton-based algorithms, e.g., [40]. However, such a method is rather inefficient since it requires a full enumeration of the reachability space of a net. An alternative approach consists in adopting state abstraction techniques such as the *basis reachability graph* (BRG) that has been successfully applied to fault diagnosis [8, 29, 36], prognosis [28], and marking estimation [30]. Since the supervisor designed for active diagnosis may induce deadlocks in a plant and the corresponding BRG does not explicitly characterize this phenomenon, the active diagnosis problem in LPNs cannot be solved by simply applying the automaton-based method in [40] to the BRG of a plant net.

In this paper, we propose a new BRG-based structure that contains the information required to analyze the presence of deadlocks in a plant. Differently from the method in [40] where all dead states are enumerated, we do not explicitly enumerate all dead markings of the plant. Instead, for each basis marking, at most one virtual basis marking is introduced to represent all dead markings. Moreover, the supervisor designed in this work guarantees that the closed-loop system is deadlock-free when no fault occurs, while the supervisor designed in [40] is not. The main contributions of this paper are summarized as follows.

- First, we generalize the notion of diagnosability to LPNs that are not necessarily deadlock-free. This is necessary because control actions enforcing active diagnosis may induce deadlocks even if an original net is deadlock-free. Moreover, we assume that a deadlock occurring in a plant can be indirectly "observed" by a modeling primitive called *quiescence* in [40] (and *time-out* in [18]). In plain words, if no firing of transitions is observed for a sufficient long time, then one can infer that a plant is blocked (due to either a deadlock or a control-induced deadlock). This inference can be modeled by a particular *quiescent event* in a logical framework.
- The conventional BRG developed in [8] does not contain sufficient information to characterize deadlocks and quiescent behavior of a plant. Thus, we develop a new BRG-like structure called the *quiescent basis reachability graph* (QBRG), in which the quiescent behavior is encoded. Analogously to a BRG, a QBRG models the quiescent behavior of an LPN without explicitly listing all reachable markings. To compute a QBRG, an integer linear programming technique is proposed to characterize the quiescent behavior of a plant net. Then a structure called a *Q-diagnoser* is developed based on a QBRG to verify the diagnosability of a plant.
- Finally, based on the notion of Q-diagnoser, we propose an algorithm to design a *diagnosability enforcing supervisor* for a given plant. The supervisor is obtained by recursively removing all *indeterminate cycles* in the Q-diagnoser, which circumvents the need of a complete marking enumeration. Moreover, the supervisor designed by our approach guarantees that the closed-loop system is deadlock-free if no

fault occurs.

Some preliminary results related to this approach have been presented in a conference paper [21], in which the observable transitions are assumed to be free-labeled. In this paper, the active diagnosis problem is studied in the more general framework of labeled Petri nets. In [21], we did not address the deadlock issue. In comparison, a supervisor designed in this paper does not incur faultless deadlocks: a deadlock may only occur when one or more fault transitions have fired. Such a property is also useful in practice: once the plant reaches a deadlock after the occurrence of faults, the operator may examine the plant and initiate a recovery process if needed. Furthermore, proofs of the mains results that were just sketched in [21] are fully developed in this paper.

The remainder of this paper is organized as follows. In Section 2, we recall the basics of labeled Petri nets and notions of diagnosability. In Section 3, the active diagnosis problem is formulated, and the notion of diagnosability is generalized to LPNs that are not necessarily deadlock-free. In Section 4, the notion of QBRG is introduced and a structure called a Q-diagnoser is developed to verify the diagnosability of an LPN. In Section 5, an algorithm is developed to compute a supervisor for active diagnosis. Finally, conclusions are drawn in Section 6.

2 Preliminary

2.1 Petri net

A Petri net is a four-tuple $PN = (P, T, Pre, Post)$, where P is a set of m *places* represented by circles and T is a set of n *transitions* represented by bars; $Pre : P \times T \to \mathbb{N}$ and $Post : P \times T \to \mathbb{N}$ are the *pre-* and *post- incidence matrices* which specify the arcs from places to transitions and from transitions to places, respectively. Here, $\mathbb{N}$ is the set of non-negative integers. $C = Post - Pre \in \mathbb{N}^{m\times n}$ is the *incidence matrix* of the net. For a transition $t \in T$, the *preset* of t is defined as ${}^\bullet t = \{p \in P \mid Pre(p, t) > 0\}$, while the *postset* of t is defined as $t^\bullet = \{p \in P \mid Post(p, t) > 0\}$.

A *marking* of a Petri net is a function M: $P \to \mathbb{N}$, which assigns to each place a non-negative integer number of *tokens*, represented by black dots. A Petri net with an initial marking M_0 is called a *marked net* and is denoted by $\langle PN, M_0\rangle$.

A transition t is *enabled* at a marking M if $M \geq Pre(\cdot, t)$ holds, which is denoted by $M[t\rangle$. At a marking M, an enabled transition t may fire reaching a new marking $M' = M + C(\cdot, t)$, which is denoted by $M[t\rangle M'$. We use $t \in \sigma$ to denote that transition t appears at least once in a sequence $\sigma \in T^*$. We write $M[\sigma\rangle M'$ with $\sigma = t_1 \cdots t_k$ to denote that at marking M transitions $t_1, \cdots, t_k$ can fire sequentially, which eventually yields marking M'. We say that marking M' is *reachable* from marking M if there exists a firing sequence $\sigma \in T^*$ such that $M[\sigma\rangle M'$. We use $L(PN, M_0)$ to represent the set of all sequences that are enabled at the initial marking M_0, i.e., $L(PN, M_0) = \{\sigma \in T^* \mid M_0[\sigma\rangle\}$.

The set of all markings that are reachable from M_0 is the *reachability set*, denoted by $R(PN, M_0)$. A marked net $\langle PN, M_0\rangle$ is *bounded* if there exists a number $k \in \mathbb{N}$ such that for all $M \in R(PN, M_0)$, and all $p \in P, M(p) \leq k$ holds.

A marking M is said to be *dead* if no transition is enabled at M. A sequence $\sigma \in L(PN, M_0)$ is *terminal* if the firing of σ reaches a dead marking M. A marked net $\langle PN, M_0\rangle$ is said to be *deadlock-free* if for all

$M \in R(PN, M_0)$, M is not dead. The following result immediately follows from the definition of dead marking.

Fact 1 *Given a Petri net $PN = (P, T, Pre, Post)$, a marking $M \in R(PN, M_0)$ is dead if and only if the following constraint set, denoted by $\rho(M)$, is feasible:*

$$\rho(M) : \bigwedge_{t \in T} (\bigvee_{p \in {}^\bullet t} M(p) \leq Pre(p, t) - 1) \tag{1}$$

Fact 1 shows that the set of dead markings can be described by a set of linear equalities that characterize the enabling conditions of transitions. The logical *OR* condition in the constraint set $\rho(M)$ in Eq. (1) can be converted to its equivalent conjunctive normal form by the method in [7].

For a sequence $\sigma \in T^*$, we use $\sigma_{\uparrow T'}$ with $T' \subseteq T$ to denote the projection of sequence σ onto the transition set T', and we write $\mathbf{y}_\sigma$ to denote the firing vector of σ, i.e., $\mathbf{y}_\sigma(t) = k$ if transition t appears k times in σ.

A Petri net is *acyclic* if it does not contain any cycle. For an acyclic net, the following result holds.

Proposition 1 *[32] Given an acyclic Petri net, a marking M is reachable from M_0 if and only if there exists a vector $\mathbf{y} \in \mathbb{N}^n$ satisfying the* state equation $M = M_0 + C \cdot \mathbf{y}$. □

2.2 Labeled Petri net

A *labeled Petri net* (LPN) is a structure $G = (PN, M_0, E, \ell)$, where PN is a Petri net, M_0 is the initial marking, E is the set of observable events and $\ell : T \to E \cup \{\varepsilon\}$ is the labeling function that assigns to each transition $t \in T$ either a symbol from the given event set E or the *empty string* ε. The set of transitions T is partitioned into two disjoint sets as follows: $T = T_o \cup T_{uo}$, where $T_o = \{t \in T \mid \ell(t) \in E\}$ is *the observable transition set* and $T_{uo} = \{t \in T \mid \ell(t) = \varepsilon\}$ is *the unobservable transition set*. The labeling function is also naturally extended to firing sequences $\ell : T^* \to E^*$.

In an LPN, the observation of a sequence $\sigma \in T^*$ is denoted as $w = \ell(\sigma) \in E^*$. The *language* of an LPN G is defined as $\mathcal{L}(G) = \{w \in E^* \mid \exists \sigma \in L(PN, M_0) : \ell(\sigma) = w\}$. Given an observation $w \in E^*$, we define $\ell^{-1}(w) = \{\sigma \in L(PN, M_0) \mid \ell(\sigma) = w\}$ as the set of firing sequences consistent with w.

2.3 Basis reachability graph

The study in [8, 27] develops a semi-structural approach to represent the reachability set of a bounded Petri net.

Definition 1 *[27] Given a Petri net $PN = (P, T, Pre, Post)$, a pair $\pi = (T_E, T_I)$ is called a* basis partition[1] *of T if (1) $T_I \subseteq T$, $T_E = T \setminus T_I$; and (2) the T_I-induced subnet is acyclic. The sets T_E and T_I are called the set of* explicit transitions *and the set of* implicit transitions*, respectively.* □

Definition 2 *[27] Given a Petri net $PN = (P, T, Pre, Post)$, a basis partition $\pi = (T_E, T_I)$, a marking M, and a transition $t \in T_E$, we define:*

[1] In general, there may exist multiple valid basis partitions for a given plant net. The selection of explicit and implicit transitions is not necessarily associated with physical meanings. However, for certain problems some particular basis partitions are useful.

- $\Sigma(M,t) = \{\sigma \in T_I^* \mid M[\sigma\rangle M', M' \geq Pre(\cdot,t)\}$ *as the set of* explanations *of transition t at marking M;*
- $Y(M,t) = \{\mathbf{y}_\sigma \in \mathbb{N}^{|T|} \mid \sigma \in \Sigma(M,t)\}$ *as the set of* explanation vectors *of transition t at marking M;*
- $Y_{min}(M,t)$ *denotes the set of all minimal elements of $Y(M,t)$, i.e., the* minimal explanation vectors. □

The set of *basis markings* $\mathcal{M}$ is recursively defined as follows:

- $M_0 \in \mathcal{M}$;
- If $M \in \mathcal{M}$, then for all $t \in T_E$, for all $\mathbf{y} \in Y_{min}(M,t)$,

$$(M' = M + C_I \cdot \mathbf{y} + C(\cdot,t)) \Rightarrow (M' \in \mathcal{M}).$$

Given a partition $\pi = (T_E, T_I)$, the corresponding *basis reachability graph (BRG)* is a deterministic finite state automaton (DFA) defined in [27]. In short, a BRG $\mathcal{B}$ is a quadruple $(\mathcal{M}, Tr, \Delta, M_0)$, where:

- the state set $\mathcal{M}$ is the set of basis markings;
- the event set Tr is the set of pairs $(t, \mathbf{y}) \in T_E \times \mathbb{N}^{|T_I|}$;
- the transition function Δ is:

$$\begin{aligned}\Delta = \{(M_1, (t,\mathbf{y}), M_2) \mid t \in T_E, \mathbf{y} \in Y_{min}(M_1,t),\\ M_2 = M_1 + C_I \cdot \mathbf{y} + C(\cdot,t)\}\end{aligned}$$

- the initial state is the initial marking M_0.

Definition 3 *[27] Given a net $PN = (P,T,Pre,Post)$, a basis partition $\pi = (T_E,T_I)$, and a basis marking M_b, the* implicit reach *of M_b is defined as $R_I(M_b) = \{M \in \mathbb{N}^m \mid (\exists \sigma \in T_I^*) M_b[\sigma\rangle M\}$.* □

Theorem 1 *[27] Given a Petri net $PN = (P,T,Pre,Post)$, a basis partition $\pi = (T_E,T_I)$, and a marking M', the following condition holds:*

$$\begin{aligned}(\exists \sigma \in T^*) \sigma_{\uparrow T_E} = \sigma_E \wedge M_0[\sigma\rangle M'\\ \Leftrightarrow (\exists M_b \in \mathcal{M})(M_0, \sigma_E, M_b) \in \Delta^* \wedge M' \in R_I(M_b)\end{aligned}$$

where $(M_0, \sigma_E, M_b) \in \Delta^*$ denotes that, in the BRG, there exists a path from M_0 to M_b labeled by $(t_1, \mathbf{y_1}), \cdots, (t_2, \mathbf{y_2})$ such that $t_1 \cdots t_k = \sigma_E$. □

3 Active diagnosis problem formulation in labeled Petri nets

Given an LPN $G = (PN, M_0, E, \ell)$, its unobservable transition set T_{uo} is partitioned into the *set of regular unobservable transitions T_{reg}* and the *set of fault transitions T_f*. The latter can be further partitioned into different fault classes $T_f^i (i = 1, 2, ..., r)$ that model different types of faults affecting the plant, i.e., $T_f = \bigcup_{i=1}^{r} T_f^i$. For the sake of simplicity, in this work we consider an LPN with a single fault class, i.e., $T_f = T_f^1$. However, our approach can be extended to the active diagnosis of nets with multiple fault classes with a slight modification of the methodology proposed in [41] for this purpose.

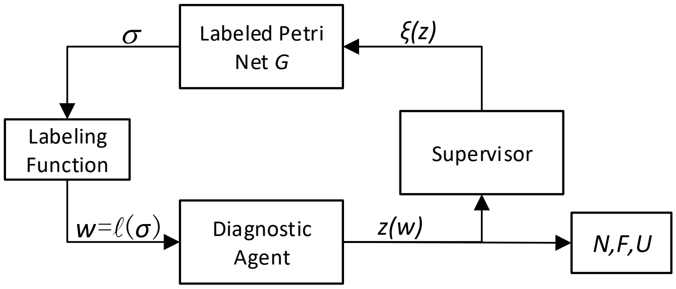


Figure 1: The active diagnosis scheme.

In [9], the notion of *diagnosability* for deadlock-free nets is presented. In simple words, given a deadlock-free LPN $G = (PN, M_0, E, \ell)$, let $\sigma \in L(PN, M_0)$ be a sequence that ends with a fault transition and let σ' be a sufficiently long continuation of σ. Plant G is diagnosable if any firing sequence having the same observation as $\sigma\sigma'$ contains at least one fault transition. According to its definition, diagnosability is a behavioral property, i.e., it depends on the language generated by a plant. Assume that we are given an undiagnosable plant represented by a labeled Petri net. We consider here a problem of active diagnosis: it consists in modifying the plants behavior by supervisory control, thus ensuring that the closed-loop system is diagnosable.

The set-up of supervisory control in LPNs is proposed in [16]. Given an LPN $G = (PN, M_0, E, \ell)$, the set of events E is partitioned into the set of controllable events E_c and the set of uncontrollable events E_{uc}, i.e., $E = E_c \cup E_{uc}$. In this paper, we aim to design a control policy based on the current *diagnostic state*. The scheme of active diagnosis is depicted in Figure 1. Specifically, when a plant net generates a sequence σ, through the labeling function, a diagnostic agent observes $w = \ell(\sigma)$ and computes the corresponding diagnostic state $z(w)$ that contains information of both the current set of markings and possible fault occurrences.

Definition 4 *A diagnostic state is a set of pairs $z(w) = \{(M_1, \gamma_1), (M_2, \gamma_1), \ldots, (M_n, \gamma_n)\}$, where each pair (M_i, γ_i) denotes that the plant may be currently at marking M_i having previously fired a fault transition ($\gamma_i = F$) or not ($\gamma_i = N$). A diagnostic state $z(w)$ can be classified into three cases:*

- Normal *(no fault transition has fired so far): if for all $(M_i, \gamma_i) \in z(w), \gamma_i = N$;*
- Faulty *(a fault transition has fired): if for all $(M_i, \gamma_i) \in z(w), \gamma_i = F$;*
- Uncertain *(a fault transition may have fired): if there exist $(M_i, \gamma_i), (M_j, \gamma_j) \in z(w)$, such that $\gamma_i = N$, $\gamma_j = F$.* □

When receiving a current diagnostic state $z(w)$, a *supervisor* makes a control decision that specifies a subset of controllable events $\xi(z) \subseteq E_c$ to execute, while all other controllable events in $E_c \backslash \xi(z)$ are disabled. Note that: (i) a supervisor cannot disable any transitions with uncontrollable labels, and (ii) if a supervisor disables an event $e \in E_c$, all transitions labeled e are disabled. Here, we use (G, ξ) to denote the closed-loop system. The problem investigated in this paper is formulated as follows.

Problem 1 (Active diagnosis) *Given an undiagnosable plant modeled by an LPN G, we want to determine a control policy ξ for G such that the closed-loop system (G, ξ) is diagnosable.* □

Since in this paper we will apply the BRG approach to represent the reachability space of a net, the LPN G considered satisfies the following assumptions:

A1) G is bounded;

A2) The T_{uo}-induced subnet is acyclic.

Assumption A1 guarantees that the BRG of a plant net is always finite. Assumption A2 allows to use the state equation to characterize the implicit reach of a basis marking.

Remark 1 *Note that in the literature, an ILPP technique [1, 3, 4] is used to characterize the reachability set of a bounded net system, which does not rely on Assumption A2. However, in this paper the proposed supervisor is computed based on the BRG approach. Assumption A2 ensures that a BRG contains a correct abstract representation of a net reachability set.* □

3.1 Diagnosability of LPNs with deadlocks

In the literature, it is commonly assumed that a plant net to be diagnosed is deadlock-free [6, 9, 24, 36]. However, the action of a supervisor for active diagnosis may create deadlocks in the closed-loop system. This happens when the closed-loop system reaches a marking while receiving a control decision that disables all plant-enabled transitions: this situation is called a *control-induced deadlock*. Therefore, to perform active diagnosis, the notion of diagnosability in LPNs needs to be generalized. In the sequel we use $\psi(T_f)$ to denote the set of all sequences in $L(PN, M_0)$ that ends with a fault transition in T_f, i.e., $\psi(T_f) = \{\sigma t_f \in L(PN, M_0) : t_f \in T_f\}$.

Definition 5 *An LPN $G = (PN, M_0, E, \ell)$ is* diagnosable *with respect to a set of fault transitions T_f if for all $\sigma' \in \psi(T_f)$, there exists a non-negative integer $k \in \mathbb{N}$ such that for all $\sigma'\sigma'' \in L(PN, M_0)$, the following two conditions hold:*

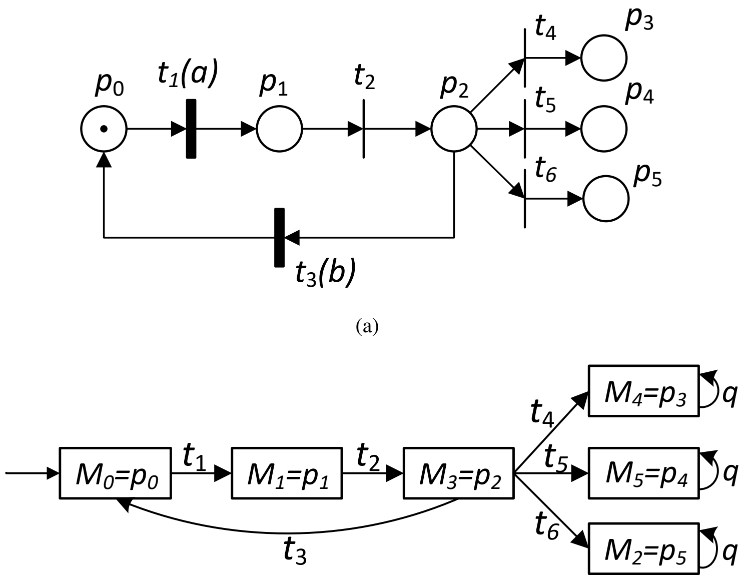
- *if $|\sigma''| \leq k$ and sequence $\sigma'\sigma''$ is terminal, then:*
$$\begin{aligned}&\sigma \in \ell^{-1}(\ell(\sigma'\sigma'')) \wedge (\nexists t \in T)\sigma t \in L(PN, M_0)\\ &\Rightarrow (\exists t_f \in T_f) t_f \in \sigma;\end{aligned}$$
- *if $|\sigma''| \geq k$, then:*
$$(\forall \sigma \in \ell^{-1}(\ell(\sigma'\sigma'')))(\exists t_f \in T_f) t_f \in \sigma. \quad \square$$

The second condition in Definition 5 is the classical notion of diagnosability for deadlock-free LPNs. On the other hand, the first condition means that if a plant reaches a dead marking after a fault transition has fired, then all consistent sequences that have the same observation and yield dead markings contain a fault. By generalizing the definition of diagnosability, we do not require the assumption that the original net is deadlock-free, i.e., our approach can be applied to LPNs containing deadlocks.

3.2 Quiescent behavior and quiescent event

When a plant reaches a marking M that is either a dead marking or a control-induced deadlock, the system halts and no observation is produced in the future. Since in many practical cases the time needed to fire a transition has an upper bound that can be assumed to be known, if the plant does not produce any observation for a sufficiently long time, one can infer that the plant must be deadlocked at some marking[2]. To this end,

[2] Note that a *divergent* plant [17] may similarly renounce to engage in any further communication with the environment even if not deadlocked. However, the models that we consider satisfy Assumptions A1 and A2, and thus they are necessarily divergence-free.



(a)

(b)

Figure 2: (a) The LPN used in Example 1, and (b) its reachability graph after adding the quiescent event q.

deadlocks can be indirectly "observed". Such a time-out behavior is called the *quiescent behavior*, which can be encoded into the model by the following mechanism:

- Once the plant is deadlocked, it will repetitively generate a particular event q called the *quiescent event*. Event q is observable and uncontrollable;
- Event q is generated only when the plant is deadlocked, i.e., the plant does not generate event q if any transition is enabled.

Note that event q is not an event associated to a sensor signal: it is a logical event that represents the condition that a plant does not produce any observation for a sufficient long time. We use the following example to illustrate this.

Example 1 *Consider the LPN in Figure 2(a) in which the set of observable transitions is $T_o = \{t_1, t_3\}$. By firing sequence $\sigma_1 = t_1t_2t_4$, the plant reaches a dead marking M_4 at which the plant generates the quiescent event q. Besides, the plant may also reach dead markings M_2, M_5 and then generate event q. This mechanism can be illustrated by adding a self-loop labeled by q at dead markings in the reachability graph shown in Figure 2(b).* □

4 Basis reachability graph with quiescence and Q-diagnosers

4.1 Diagnosability of deadlock-free LPNs and basis diagnosers

Given a bounded LPN, one can construct its reachability graph (RG) and use the automata approaches presented in [40] for active diagnosis. Nevertheless, the construction of the RG needs to explicitly enumerate all reachable markings of a net. In this paper, we use the notion of basis reachability graph that is a compact representation of the reachability space of a net.

In [9], the authors use a basis reachability graph (BRG) with respect to a particular partition to study the diagnosability verification problem in LPNs. Such a BRG is called a *diagnostic BRG*.

Definition 6 *A diagnostic BRG is a basis reachability graph with respect to a partition where $T_E = T_o \cup T_f$ and $T_I = T \setminus T_E$.* □

In a diagnostic BRG, each state is a basis marking and each arc is labeled with a pair $(t, \mathbf{y})$, where $t \in T_o \cup T_f$ and $\mathbf{y}$ is a minimal explanation vector to enable t. Based on the diagnostic BRG, an automaton called a *basis reachability diagnoser* (BRD) [9] is computed and used to verify the diagnosability of the net. To compute a BRD, a series of ILPPs need to be solved to flag the occurrence of faults. In this paper, we will also use the diagnostic BRG structure and will develop a simplified BRD structure to verify the diagnosability of an LPN. In our case, no ILPP has to be solved and the simplified diagnoser structure will be later used to design a supervisor.

Definition 7 *Given an LPN $G = (PN, M_0, E, \ell)$ and its diagnostic BRG $\mathcal{B} = (\mathcal{M}, Tr, \Delta, M_0)$, the* underlying automaton *of $\mathcal{B}$ is a nondeterministic finite state automaton $G_l = (\mathcal{M}, E \cup \{\varepsilon_f\}, \Delta_l, M_0)$, where:*

- *$\mathcal{M}$ is the set of states;*
- *$E \cup \{\varepsilon_f\}$ is the event set, where ε_f denotes a fault event that is unobservable.*
- *$\Delta_l \subseteq \mathcal{M} \times (E \cup \{\varepsilon_f\}) \times \mathcal{M}$ is the transition relation defined as follows: for any $M_1, M_2 \in \mathcal{M}$ and $(t, \mathbf{y}) \in Tr$,*
  $(M_1, (t, \mathbf{y}), M_2) \in \Delta \wedge \ell(t) \in E \Rightarrow (M_1, \ell(t), M_2) \in \ \Delta_l$,
  $(M_1, (t, \mathbf{y}), M_2) \in \Delta \wedge t \in T_f \Rightarrow (M_1, \varepsilon_f, M_2) \in \Delta_l$;
- *M_0 is the initial state.* □

In other words, the *underlying automaton* of a diagnostic BRG $\mathcal{B}$, denoted by G_l, can be obtained by changing the label of each arc in $\mathcal{B}$ from $(t, \mathbf{y})$ to $\ell(t)$ (if $t \in T_o$) or ε_f (if $t \in T_f$). Now we show that the diagnosability of a deadlock-free LPN implies the diagnosability of the corresponding automaton G_l, and vice versa.

Theorem 2 *Given a deadlock-free LPN $G = (PN, M_0, E, \ell)$, let $G_l = (\mathcal{M}, E \cup \{\varepsilon_f\}, \delta_l, M_0)$ be the underlying automaton of its diagnostic BRG $\mathcal{B}$. The net G is diagnosable if and only if G_l is diagnosable with respect to fault ε_f.*

Proof. (If) Assume that G_l is diagnosable with respect to fault ε_f. For any sequence $\sigma_1 \varepsilon_f \in L(G_l)$, there exists $k \in \mathbb{N}$ such that by observing subsequent k events, the occurrence of the fault can be detected. According to Theorem 1, in net G for any sequence $\sigma_1' \in \psi(T_f)$, and all sequences $\sigma_1' \sigma_2' \in L(PN, M_0)$ such that $|\ell(\sigma_2')| \geq k$, all sequences in $\ell^{-1}(\ell(\sigma_1' \sigma_2'))$ contain a fault in T_f. Since the unobservable subnet is acyclic, the length of σ_2' is bounded, which indicates that G is diagnosable.

(Only If) If net G is not diagnosable, there exists two arbitrary long sequence $\sigma_1, \sigma_2 \in L(PN, M_0)$, such that $\ell(\sigma_1) = \ell(\sigma_2)$ and σ_1 contains a fault while σ_2 does not. By Theorem 1 and the definition of G_l, it is obvious that there exist two arbitrary long sequence $\sigma_1', \sigma_2' \in L(G_l)$ such that $P_o(\sigma_1') = P_o(\sigma_2')$ and one contains fault ε_f while the other does not. Therefore, G_l is not diagnosable. ■

Table 1: The basis markings of the BRG in Figure 3(b).

M_0	[1	0	0	0	0	0	0	0	0	1	$0]^T$
M_1	[0	0	1	0	0	0	0	0	0	1	$0]^T$
M_2	[0	0	0	2	0	0	0	0	0	1	$0]^T$
M_3	[0	0	0	1	0	0	0	1	0	1	$0]^T$
M_4	[0	0	0	1	0	0	0	0	0	0	$1]^T$
M_5	[0	0	0	0	0	0	0	2	0	1	$0]^T$
M_6	[0	0	0	0	0	0	0	1	0	0	$1]^T$
M_7	[0	0	0	0	1	0	0	0	0	1	$0]^T$
M_8	[0	0	0	0	0	1	0	0	0	1	$0]^T$
M_9	[0	0	0	0	0	0	1	0	0	1	$0]^T$

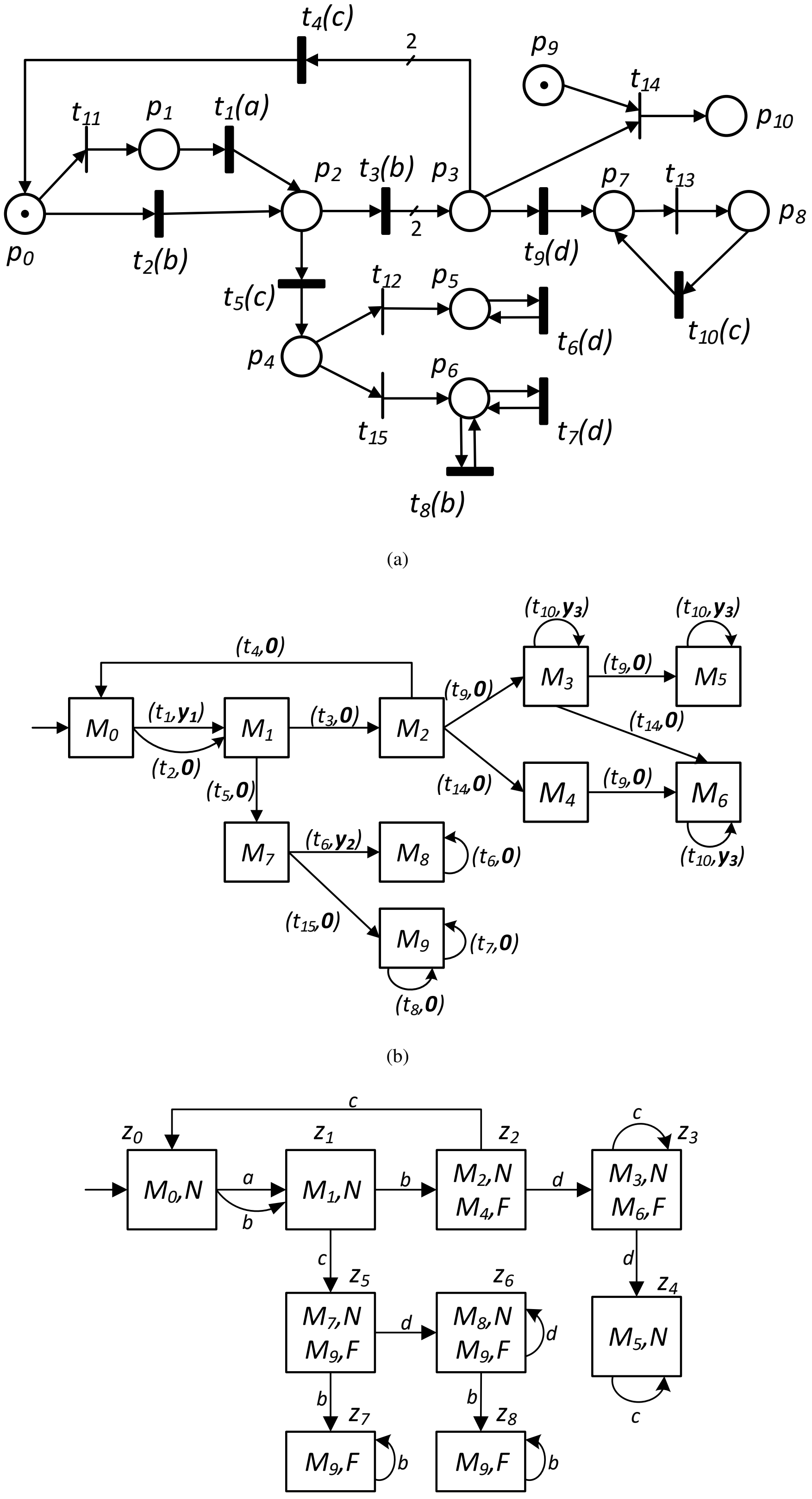
In [41], a structure called *diagnoser* are used to verify the diagnosability of an automaton. The diagnoser of an automaton can be constructed by a standard procedure [11, 41]: we do not present it here for the sake of brevity but illustrate it via Example 2. An important notion related to diagnosability is the *indeterminate cycle* [41]. An indeterminate cycle in a diagnoser is a cycle composed exclusively of uncertain diagnostic states for which there exist: (i) a corresponding cycle in the plant automaton involving only states tagged "N" in the cycle of diagnoser; and (ii) a corresponding cycle in the plant automaton involving only states tagged "F" in the cycle of diagnoser. Sampath *et. al* [41] show that a plant automaton is diagnosable if and only if its diagnoser does not contain any indeterminate cycle. Therefore, by Theorem 2, we immediately have the following corollary.

Corollary 1 *Given an LPN G that is deadlock-free, let G_l be the underlying automaton of its diagnostic BRG $\mathcal{B}$. The net G is diagnosable if and only if the diagnoser of G_l does not contain any indeterminate cycle.* □

Example 2 *Consider the LPN in Figure 3(a), where $T_o = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}\}$, $T_{uo} = \{t_{11}, t_{12}, t_{13}, t_{14}, t_{15}\}$, and $T_f = \{t_{14}, t_{15}\}$. The labeling function is defined as follows: $\ell(t_1) = a$, $\ell(t_2) = \ell(t_3) = \ell(t_8) = b$, $\ell(t_4) = \ell(t_5) = \ell(t_{10}) = c$, and $\ell(t_6) = \ell(t_7) = \ell(t_9) = d$. The diagnostic BRG of this net is shown in Figure 3(b), where $\mathbf{y_1} = [1, 0, 0]^T$, $\mathbf{y_2} = [0, 1, 0]^T$, and $\mathbf{y_3} = [0, 0, 1]^T$. The basis markings are listed in Table 1. According to Definition 7, the underlying automaton G_l can be easily computed based on the diagnostic BRG. The diagnoser of G_l is shown in Figure 3(c). There exist two indeterminate cycles in the diagnoser, i.e., $z_3 \xrightarrow{c} z_3$ and $z_6 \xrightarrow{d} z_6$. According to Corollary 1, the plant net is not diagnosable.* □

4.2 Quiescent basis reachability graph and Q-diagnoser

As mentioned in the previous section, a diagnostic BRG can be used to characterize the behavior of a deadlock-free LPN and to verify its diagnosability without explicitly computing the reachability graph. Since a diagnostic BRG is an abstract model of the plant LPN and does not preserve the information needed to characterize deadlocks, the automaton-based approach [40] for active diagnosis cannot be directly applied to a diagnostic BRG. In fact, a basis marking that has an outgoing arc in a diagnostic BRG does not necessarily imply that all markings reachable from it by firing regular unobservable transitions are not dead. Therefore,



(a)

(b)

(c)

Figure 3: (a) The LPN for Example 2, (b) its diagnostic BRG, and (c) the diagnoser of G_l.

for the purpose of active diagnosis in LPNs with deadlocks, the structure of conventional diagnostic BRGs needs to be augmented to encode the information of deadlocks.

Fact 1 in Section 2 provides us a way to verify if the regular unobservable reach of a (basis) marking contains dead markings, which is stated by the following proposition. Here we denote $R_{reg}(M) = \{M' \mid M[\sigma_{reg}\rangle M', \sigma_{reg} \in T_{reg}^*\}$.

Proposition 2 *Given an LPN $G = (PN, M_0, E, \ell)$ and a marking M, there exists at least one dead marking in $R_{reg}(M)$ if and only if the following linear integer constraint $D(M)$ is feasible:*

$$D(M) = \begin{cases} M_d = M + C_{uo} \cdot \mathbf{y} \geq \mathbf{0}, \\ \mathbf{y} \in \mathbb{N}^{|T_{uo}|}, \\ \sum\limits_{t_f \in T_f} \mathbf{y}(t_f) = 0 \\ \rho(M_d). \end{cases} \tag{2}$$

Proof. (Only If) Suppose that there exists a dead marking $M_d \in R_{reg}(M)$, i.e., there exists a firing vector $\mathbf{y} \in \mathbb{N}^{|T_{uo}|}$ such that $M + C_{uo} \cdot \mathbf{y} = M_d \geq \mathbf{0}$ and $\mathbf{y}(t_f) = 0$ for all $t_f \in T_f$. By Fact 1, marking M_d satisfies $\rho(M_d)$. Therefore, ILPP (2) is feasible.

(If) Suppose that ILPP (2) is feasible. By Assumption 2, the T_{uo}-induced subnet is acyclic. If there exists a firing vector $\mathbf{y} \in \mathbb{N}^{|T_{uo}|}$ such that $M + C_{uo} \cdot \mathbf{y} = M_d \geq \mathbf{0}$ and $\mathbf{y}(t_f) = 0$ for all $t_f \in T_f$, then there necessarily exists a firing sequence $\sigma \in T_{reg}^*$ such that $M[\sigma\rangle M_d$ whose firing vector is $\mathbf{y}$. Since M_d satisfies $\rho(M_d)$, by Fact 1, M_d is a dead marking. ■

Note that the dead markings in $R_{reg}(M_i)$ may not be unique. However, we will shortly see that for the purpose of active diagnosis, it is sufficient to use a single virtual marking $M_{i,d}$ to denote the existence of some dead markings in $R_{reg}(M_i)$ without explicitly enumerating them. In the following, we introduce a structure called a *quiescent-BRG* (QBRG) that is an augmented basis reachability graph in which the information of the quiescent behavior of a plant net is encoded.

Definition 8 *Given an LPN $G = (PN, M_0, E, \ell)$, let $G_l = (\mathcal{M}, E \cup \{\varepsilon_f\}, \Delta_l, M_0)$ be the underlying automaton of its diagnostic BRG. The* quiescent-BRG *(QBRG) of G is a nondeterministic finite state automaton $G_q = (\mathcal{M}_q, E_q, \Delta_q, M_0)$, where:*

- *the state set $\mathcal{M}_q = \mathcal{M} \cup \{M_{i,d} \mid M_i \in \mathcal{M}, D(M_i)$ is feasible$\}$;*
- *$E_q = E \cup \{\varepsilon_f, q\}$ is the event set;*
- *the transition relation Δ_q is defined as follows:*

$$\Delta_q = \Delta_l \cup \{(M_i, q, M_{i,d}) \mid M_i \in \mathcal{M}, D(M_i) \textit{ is feasible}\}$$
$$\cup \{(M_{i,d}, q, M_{i,d}) \mid M_{i,d} \in \mathcal{M}_q \setminus \mathcal{M}\}$$

- *M_0 is the initial state.* □

Given an LPN, Algorithm 1 can be used to compute its QBRG. The difference between the QBRG and the diagnostic BRG of an LPN can be explained as follows. For each basis marking M_i in a diagnostic BRG, if constraint $D(M_i)$ is feasible, then virtual basis marking $M_{i,d}$ is added with arcs $M_i \xrightarrow{q} M_{i,d}$ and $M_{i,d} \xrightarrow{q} M_{i,d}$. Again, we note that $M_{i,d}$ is not a real marking of a Petri net: it is just a modeling primitive to denote the existence of some dead markings in $R_{reg}(M_i)$. However, for simplicity, we also call $M_{i,d}$ a "basis marking" by omitting the term "virtual", since there will be no confusion. If an LPN is deadlock-free, then its QBRG is identical to the underlying automaton of its diagnostic BRG.

Algorithm 1: Computation of QBRG G_q.

```
Input: An LPN $G = (PN, M_0, E, \ell)$
Output: $G_q = (\mathcal{M}_q, E \cup \{\varepsilon_f, q\}, \Delta_q, M_0)$
compute the diagnostic BRG $\mathcal{B}$ of the LPN $G$;
compute the underlying automaton $G_l = (\mathcal{M}, E \cup \{\varepsilon_f\}, \Delta_l, M_0)$ of $\mathcal{B}$;
let $\mathcal{M}_q = \mathcal{M}$, $\Delta_q = \Delta_l$;
for each basis marking $M_i \in \mathcal{M}$ do
    if $D(M_i)$ is feasible then
        let $\mathcal{M}_q = \mathcal{M}_q \cup \{M_{i,d}\}$;
        let $\Delta_q = \Delta_q \cup \{(M_i, q, M_{i,d})\} \cup \{(M_{i,d}, q, M_{i,d})\}$;
output $G_q = (\mathcal{M}_q, E \cup \{\varepsilon_f, q\}, \Delta_q, M_0)$;
```

In the following, we introduce a new projection function $P' : T^* \to (E \cup \{\varepsilon_f\})^*$, defined as follows:

$$\begin{cases} P'(t) = e \ \text{ if } t \in T_o, \ \ell(t) = e; \\ P'(t) = \varepsilon \ \text{ if } t \in T_{reg}; \\ P'(t) = \varepsilon_f \ \text{ if } t \in T_f; \\ P'(\sigma t) = P'(\sigma)P'(t) \ \text{ if } \sigma \in T^*, \ t \in T. \end{cases}$$

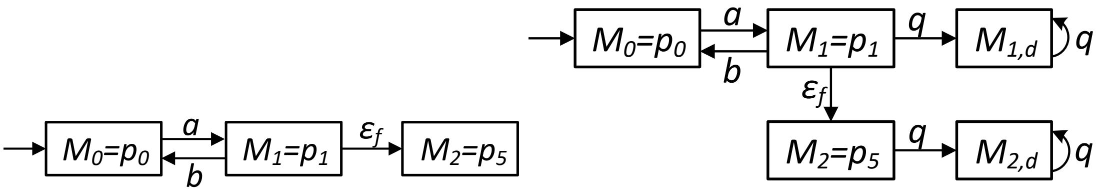
Theorem 3 *Consider an LPN $G = (PN, M_0, E, \ell)$ and its QBRG $G_q = (\mathcal{M}_q, E_q, \Delta_q, M_0)$. There exists a sequence $\sigma \in L(PN, M_0)$ satisfying $P'(\sigma) = w$ and $M_0[\sigma\rangle M$ where M is a dead marking if and only if in the QBRG G_q there exists a path: $M_0 \xrightarrow{w} M_i \xrightarrow{q} M_{i,d}$ such that $M \in R_{reg}(M_i)$.*

Proof. (Only If) Assume that there exists a sequence $\sigma \in L(PN, M_0)$ reaching to a dead marking M. Let G_l be the underlying automaton of the diagnostic BRG. By Theorem 1, in G_l there exists a path labeled by $w = P'(\sigma)$ leading to a basis marking M_i and $M \in R_{reg}(M_i)$. By the definition of G_q, there exists a path $M_0 \xrightarrow{w} M_i \xrightarrow{q} M_{i,d}$ in G_q such that $M \in R_{reg}(M_i)$.

(If) Assume that there exits a path $M_0 \xrightarrow{w} M_i \xrightarrow{q} M_{i,d}$ in G_q. We can infer that in G_l there exists a path $M_0 \xrightarrow{w} M_i$ and $D(M_i)$ is feasible, which implies that there exists a dead marking $M \in R_{reg}(M_i)$. By Theorem 1, there exists a sequence $\sigma \in L(PN, M_0)$ such that $M_0[\sigma\rangle M$ and $P'(\sigma) = w$. ■

Example 3 *Consider the LPN in Figure 2(a), where $T_o = \{t_1, t_3\}$ and $T_f = \{t_6\}$. The underlying automaton G_l of its diagnostic BRG is shown in Figure 4(a). By applying Algorithm 1, its QBRG is depicted in Figure 4(b).* □

Given a QBRG G_q, let $G_{diag} = (Z, E \cup \{q\}, \delta_d, z_0)$ be the diagnoser of G_q. G_{diag} is called a Q-diagnoser that can be computed by the standard diagnoser construction [41, 11]. The following theorem provides us a way to verify the diagnosability of an LPN that contains deadlocks by using its Q-diagnoser G_{diag}.



(a) (b)

Figure 4: (a) The structure G_l for Example 3, and (b) the corresponding QBRG G_q.

Theorem 4 *Given an LPN $G = (PN, M_0, E, \ell)$ and its Q-diagnoser $G_{diag} = (Z, E \cup \{q\}, \delta_d, z_0)$, G is diagnosable if and only if G_{diag} does not contain any indeterminate cycle.*

Proof. By Corollary 1, there does not exist an undiagnosable non-terminal sequence if and only if in the Q-diagnoser G_{diag} there does not exist an indeterminate cycle labeled by $w \in E^*$. Now we prove that there does not exist an undiagnosable terminal sequence if and only if in the Q-diagnoser G_{diag} there does not exist an indeterminate cycle labeled by event q.

(Only If) Let σ be a faulty sequence in $L(PN, M_0)$ that leads to a dead marking M. If σ does not meet the first condition in Definition 5, then there exists another non-faulty sequence σ' that yields a dead marking M' such that $\ell(\sigma) = \ell(\sigma') = w$. This indicates that by observing wq the corresponding diagnostic state $z = \delta_d^*(z_0, wq)$ necessarily contains two pairs (M_d', N) and (M_d, F). Since by the construction of the Q-diagnoser, $\delta_d(z, q) = z$ holds, there exists an indeterminate cycle at diagnostic state z labeled by q, i.e., $z \xrightarrow{q} z$.

(If) Suppose that the Q-diagnoser contains an indeterminate cycle $z \xrightarrow{q} z$ such that z contains two pairs (M_d, F) and (M_d', N). It means that there necessarily exist two dead markings M and M' that can be reached form the initial marking M_0 by firing a faulty sequence σ and a non-faulty sequence σ', respectively. Since sequences σ and σ' have the same observation, the first condition in Definition 5 is not satisfied. Therefore, the statement holds. ■

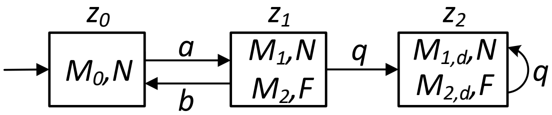


Figure 5: The Q-diagnoser G_{diag} of the plant in Figure 2(a).

Example 4 *Consider the net in Figure 2(a), where $T_o = \{t_1, t_3\}$ and $T_f = \{t_6\}$. Its QBRG G_q is visualized in Figure 4(b). Thanks to G_q, its Q-diagnoser G_{diag} is shown in Figure 5. Since in G_{diag} there exists an indeterminate cycle $z_2 \xrightarrow{q} z_2$, by Theorem 4, the plant is not diagnosable. In fact, the plant can fire normal sequence $\sigma_1 = t_1t_2t_4$ and faulty sequence $\sigma_2 = t_1t_2t_6$ that are both dead and have the same observation, which violates the first condition in Definition 5.* □

5 Diagnosability enforcing supervisor

In this section, we develop a method to design a supervisor for the active diagnosis of a plant LPN. By Theorem 4, a plant LPN is undiagnosable if and only if its Q-diagnoser contains indeterminate cycles. Therefore, to enforce diagnosability, a supervisor must forbid all behaviors of the plant that may evolve along those indeterminate cycles in the Q-diagnoser. Given a Q-diagnoser, indeterminate cycles are classified into two types by the controllability of events in them:

- for an indeterminate cycle that contains controllable events, at least one of these controllable events has to be disabled to prevent the plant circulating in this cycle for infinite times;
- for an indeterminate cycle that does not contain any controllable event, all diagnostic states in it (and all diagnostic states that may uncontrollably reach it) must be forbidden.

As mentioned in Section 3, the supervisor makes control decisions from the knowledge of the consistent diagnostic state obtained by the Q-diagnoser. More precisely, for an observation $w \in (E \cup \{q\})^*$ whose consistent diagnostic state is z_i, the control decision at state z_i is $\xi(z_i) = E_c \setminus E_{d,i}$ where $E_{d,i}$ is the set of disabled events at diagnostic state z_i. Given a set $E_{d,i}$, we use $T_{d,i}$ to denote the corresponding set of disabled transitions, i.e., $T_{d,i} = \{t \in T \mid \ell(t) \in E_{d,i}\}$.

In automata, the two control specifications mentioned above can be easily enforced by inspecting the plant automaton [40]. However, a Q-diagnoser is an abstract model of the plant LPN, in which the firing of implicit transitions is omitted. Hence, a control decision at a diagnostic state z may result in deadlocks that are not explicitly represented in the Q-diagnoser. Therefore, the method to trim a diagnoser in automata in [40] cannot be applied to trim a Q-diagnoser. To detect if there exists a control-induced deadlock at a diagnostic state z, we rewrite Eqs. (1) and (2) as the following Eqs. (3) and (4), respectively.

$$\rho'(M) : \bigwedge_{t \in T \setminus T_{d,i}} \big(\bigvee_{p \in {}^\bullet t} M(p) \leq Pre(p,t) - 1 \big) \tag{3}$$

$$D'(M) = \begin{cases} M_d = M + C_{uo} \cdot \mathbf{y} \geq 0, \\ \mathbf{y} \in \mathbb{N}^{|T_{uo}|}, \\ \sum\limits_{t_f \in T_f} \mathbf{y}(t_f) = 0, \\ \rho'(M_d). \end{cases} \tag{4}$$

Comparing with Eq. (1), in Eq. (3) the token-disabling constraints for control-disabled transitions (i.e., transition in $T_{d,i}$) are removed. On the other hand, Eq. (4) is analogous to Eq. (2) while the constraint ρ from Eq. (1) is replaced by ρ' from Eq. (3). Hence, if Eq. (4) is feasible for a marking M, by firing regular unobservable transitions, some marking M_d is reached from M such that all transitions are either control-disabled or lack of tokens to fire, and vice versa.

Proposition 3 *Given an LPN $G = (PN, M_0, E, \ell)$ with $E = E_c \cup E_{uc}$ and its Q-diagnoser $G_{diag} = (Z, E \cup \{q\}, \delta_d, z_0)$, suppose that the control decision at diagnostic state z_i is $\xi(z_i) = E_c \setminus E_{d,i}$. For a basis marking M such that $(M, \gamma) \in z_i$, there exists at least one marking $M_d \in R_{reg}(M)$ at which no transition can fire if and only if constraint $D'(M)$ in Eq.* (4) *is feasible.*

Proof. (Only If) Suppose that after disablement of transitions in $T_{d,i}$, there exists a dead marking $M_d \in R_{reg}(M)$, i.e., there exists a firing vector $\mathbf{y} \in \mathbb{N}^{|T_{uo}|}$ such that $M + C_{uo} \cdot \mathbf{y} = M_d \geq \mathbf{0}$ and $\mathbf{y}(t_f) = 0$ for all $t_f \in T_f$. Moreover, marking M_d satisfies $\rho'(M_d)$. Therefore, ILPP (4) is feasible.

(If) Suppose that ILPP (4) is feasible. By Assumption 2, the T_{uo}-induced subnet is acyclic. If there exists a firing vector $\mathbf{y} \in \mathbb{N}^{|T_{uo}|}$ such that $M + C_{uo} \cdot \mathbf{y} = M_d \geq \mathbf{0}$ and $\mathbf{y}(t_f) = 0$ for all $t_f \in T_f$, then there exists a firing sequence $\sigma \in T_{ref}^*$ such that $M[\sigma\rangle M_d$ whose firing vector is $\mathbf{y}$. Since M_d satisfies $\rho'(M_d)$, at marking M_d all transitions are either control-induced or lack of tokens to fire. Therefore, M_d is a dead marking. ■

Example 5 *Consider the plant in Figure 3(a), where $E_c = \{c, d\}$. Its Q-diagnoser is shown in Figure 3(c). Suppose that a supervisor disables event c at diagnostic state z_3, i.e., the disabled transition set $T_{d,3} = \{t_4, t_5, t_{10}\}$. Since there are two pairs (M_3, N) and (M_6, F) in state z_3, according to Proposition 3 markings M_3 and M_6 need to be considered. For marking M_3, constraint $D'(M_3)$ is not feasible, which means that if the plant is at a marking in the regular unobservable reach of M_3, by disabling event c, there is no control-induced deadlock. On the other hand, constraint $D'(M_6)$ is feasible, which means that if the plant is at a marking in the regular unobservable reach of M_6, by disabling event c, the plant can reach some dead marking in $R_{reg}(M_6)$.* □

In reality, a plant is expected to be deadlock-free, since an unexpected deadlock may greatly reduce the rate of productivity (e.g., long down-time and low use of some critical and expensive resources) or even cause severe consequence [26]. On the other hand, if a fault has occurred, then a deadlock, i.e., a "planned shutdown", is usually harmless and acceptable, since the operator of a plant may examine the plant when it is offline and initiate a recovering process to repair the fault. Therefore, in this section we aim to design a supervisor for active diagnosis which meets the two criteria:

1. the closed-loop system is diagnosable, i.e., the firing of fault transitions can be detected in finite future steps;
2. the closed-loop system is not deadlocked when no fault transition has fired.

To prevent the plant from reaching unfaulty deadlocks, in the following we introduce a notion called a *q-normal cycle*.

Definition 9 *Let $G_{diag} = (Z, E \cup \{q\}, \delta_d, z_0)$ be the Q-diagnoser of an LPN G. A cycle $\mathcal{C}$: $z \xrightarrow{q} z$ in G_{diag} is called a q-normal cycle if for all $(M_i, \gamma_i) \in z$, $\gamma_i = N$ holds.* □

In other words, a cycle $\mathcal{C} : z \xrightarrow{q} z$ is *q-normal* if at diagnostic state z no fault transition has fired. As we have discussed at the beginning of this section, to guarantee diagnosability a supervisor should prevent the plant from circulating in any indeterminate cycle. Besides, to guarantee that the plant is not deadlocked when no fault transition has fired, a supervisor should also prevent the closed-loop system reaching any q-normal cycle.

In the following, we propose Algorithm 2 to design a supervisor for a given labeled Petri net plant such that the closed-loop system is diagnosable and cannot reach deadlock if no fault occurs. Algorithm 2 recursively trims the Q-diagnoser by eliminating all indeterminate cycles and q-normal cycles in it. In Algorithm 2, we use $E_{\mathcal{C}}$ to denote the set of events in a cycle $\mathcal{C}$ in G_{diag} and use set Z_{dis} to denote the set of diagnostic states at which the supervisor makes disablement actions.

Algorithm 2: Computation of an active diagnosis supervisor

```
Input: A labeled Petri net $G = (PN, M_0, E, \ell)$, where $E = E_c \cup E_{uc}$
Output: Diagnosability enforcing supervisor $G_s$.
compute the QBRG $G_q$ using Algorithm 1;
compute the Q-diagnoser $G_{diag} = (Z, E \cup \{q\}, \delta_d, z_0)$ that is the diagnoser of $G_q$;
let $\mathcal{D} = \emptyset$, $Z_{dis} = \emptyset$;
while there exists an indeterminate cycle or a q-normal cycle $\mathcal{C}$ in $G_{diag}$ do
    if $E_{\mathcal{C}} \cap E_c = \emptyset \wedge (\exists \sigma \in E_{uc}^*) \delta_d(z_0, \sigma) = z' \in \mathcal{C}$ then
        output: No solution, END;
    if $E_{\mathcal{C}} \cap E_c \neq \emptyset$ then
        select $e \in E_{\mathcal{C}} \cap E_c$ such that $\delta_d(z_i, e) = z', z_i, z' \in \mathcal{C}$;
        let $\mathcal{D} = \mathcal{D} \cup \{(z_i, e)\}$, $Z_{dis} = \{z_i\}$;
        remove $\delta_d(z_i, e)$ from $\delta_d$;
    else
        for each $z_i \in Z$, $e \in E_c$, such that $\delta_d(z_i, e) = z'$ and $\exists \sigma \in E_{uc}^*$ such that $\delta_d(z', \sigma) = z'' \in \mathcal{C}$
        do
            let $\mathcal{D} = \mathcal{D} \cup \{(z_i, e)\}$, $Z_{dis} = Z_{dis} \cup \{z_i\}$;
            remove $\delta_d(z_i, e)$ from $\delta_d$;
    remove all unreachable states from $Z$;
    remove $(z, e)$ from $\mathcal{D}$ if $z \notin Z$ ;
    for each $z_i \in Z_{dis}$ do
        let $E_{d,i} = \{e \mid \exists (z_i, e) \in \mathcal{D}\}$;
        compute the disabled transitions set $T_{d,i}$;
        for each $(M_j, \gamma_j) \in z_i$ do
            if $D'(M_j)$ is feasible then
                if $\exists z \in Z$, such that $\delta_d(z_i, q) = z$ then
                    let $z = z \cup \{(M_{j,d}, \gamma_j)\}$
                else
                    let $z = \{(M_{j,d}, \gamma_j)\}$, $Z = Z \cup \{z\}$;
                    let $\delta_d(z_i, q) = z, \delta_d(z, q) = z$;
    let $Z_{dis} = \emptyset$
output $G_s = (Z, E \cup \{q\}, \delta_d, z_0)$.
```

We explain how Algorithm 2 works step-by-step. Steps 1 and 2 compute the QBRG and the Q-diagnoser G_{diag}, respectively. If there exists an indeterminate cycle or a q-normal cycle $\mathcal{C}$ that contains only uncontrollable events and can be reached from the initial diagnostic state z_0 by executing a sequence of uncontrollable events, then there does not exist a supervisor that meets the two criteria. In such a case, the algorithm terminates.

The main body of Algorithm 2 consists of two parts. In the first part (Steps 4 to 16) an indeterminate cycle or a q-normal cycle $\mathcal{C}$ is found and treated. If $\mathcal{C}$ contains at least one arc labeled by a controllable event $e \in E_c$, by Steps 7 to 10 one of such arcs, denoted by (z_i, e), is put into set $\mathcal{D}$, meaning that event e is disabled at diagnostic state z_i. On the other hand, if $\mathcal{C}$ contains no controllable events, then in Steps 12 to 14 all controllable arcs leading to some states that may uncontrollably reach $\mathcal{C}$ are put in $\mathcal{D}$. Steps 15 and 16 remove the unreachable states and the corresponding arcs in $\mathcal{D}$ which are no longer necessary.

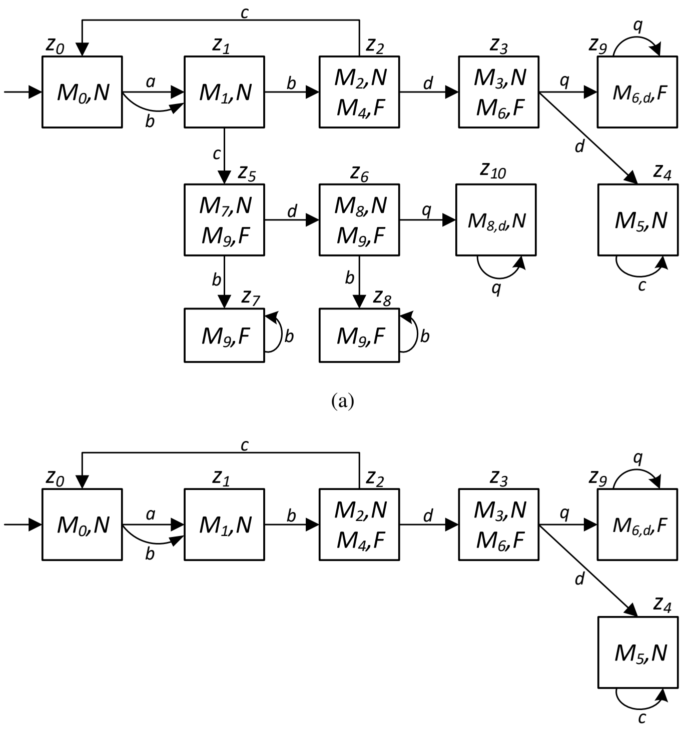
Once the control policy is updated, the second part of the algorithm (Steps 17 to 26) updates the information of control-induced deadlock accordingly. For each diagnostic state $z_i \in Z_{dis}$, Steps 18 and 19 compute the disabled event set $E_{d,i}$ and the disabled transitions set $T_{d,i}$ at state z_i, respectively. By Step 20, for each pair $(M_j, \gamma_j) \in z_i$, ILPP (4) is solved to test if the disablement of transitions in $T_{d,i}$ will block the plant at some marking in $R_{reg}(M_j)$. In Steps 22 to 26 a new diagnostic state is added to Z that contains all $(M_{j,d}, \gamma_j)$ from all (M_j, γ_j) in z_i such that ILPP (4) is feasible. The above procedures (Steps 4 to 27) are iteratively done until all indeterminate cycles and q-normal cycles have been removed. In Algorithm 2, we separate the function that updates the Q-diagnoser (Steps 17 to 26, which adds q-cycles) and the function that trims the Q-diagnoser (Steps 7 to 14). The reason for this is to modularize the algorithm and improve its readability.

Finally, we discuss the complexity of the proposed approach. Consider an LPN $G = (PN, M_0, E, \ell)$ whose diagnostic BRG is $\mathcal{B} = (\mathcal{M}, Tr, \Delta, M_0)$ with the QBRG G_q. Since in QBRG G_q each basis marking is associated with at most one virtual basis marking, there are at most $2|\mathcal{M}|$ states in G_q, i.e., the structural complexity of G_q is $O(2|\mathcal{M}|)$. On the other hand, both Q-diagnoser G_{diag} and active diagnosis supervisor G_s contain two types of nodes: (1) basis marking nodes, i.e., $z = \{(M_1, \gamma_1), (M_2, \gamma_2), \ldots, (M_n, \gamma_n)\}$, where $M_i \in \mathcal{M}$; and (2) virtual basis marking nodes, i.e., $z' = \{(M_{1,d}, \gamma_1), (M_{2,d}, \gamma_2), \ldots, (M_{n,d}, \gamma_n)\}$. Since the number of each type of nodes is at most $2^{2|\mathcal{M}|}$, both G_{diag} and G_s contain at most $2 \cdot 2^{2|\mathcal{M}|}$ states, i.e., their structural complexity is $O(2^{2|\mathcal{M}|})$. Such exponential complexity of $|\mathcal{M}|$ seems unavoidable due to the construction of the diagnoser of QBRG. However, it has been acknowledged that in practice the number of basis markings is usually much smaller than the markings in the reachability graph [9, 27]. Thus, our method is practically more efficient than automaton-based methods (such as [40]).

Example 6 *Consider again the LPN in Figure 3(a) where $T_o = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}\}$ and $T_f = \{t_{14}, t_{15}\}$. The controllable event set $E_c = \{c, d\}$. Its Q-diagnoser is shown in Figure 3(d). In the Q-diagnoser there exist two indeterminate cycles: $z_3 \xrightarrow{c} z_3$ and $z_6 \xrightarrow{d} z_6$.*

In the first iteration, indeterminate cycle $z_3 \xrightarrow{c} z_3$ is picked. Since event c is controllable, by Step 9 event c is disabled at z_3, i.e., $\mathcal{D} = \{(z_3, c)\}$ and $Z_{dis} = \{z_3\}$. For state z_3, the disabled event set $E_{d,3} = \{c\}$ and the disabled transition set is $T_{d,3} = \{t_4, t_5, t_{10}\}$. There are two pairs (M_3, N) and (M_6, F) in state z_3. For marking M_6, constraint $D'(M_6)$ is feasible, and according to Step 25, a new state $z_9 = (M_{6,d}, F)$ is added to the state set. By Step 26, an arc labeled event q from state z_3 to z_9 and a self-loop labeled event q at state z_9 are also added.

In the second iteration, indeterminate cycle $z_6 \xrightarrow{d} z_6$ is picked. Since event d is controllable, we have $\mathcal{D} = \{(z_3, c), (z_6, d)\}$ and $Z_{dis} = \{z_6\}$. By Steps 17 to 26, diagnostic state $z_6 = \{(M_8, N), (M_9, F)\}$ is



(a)

(b)

Figure 6: (a) The trimmed structure G'_{diag} in Example 6, (b) the diagnosability enforcing supervisor G_s.

examined to update the quiescent behavior after disabling event d at it. The trimmed Q-diagnoser G'_{diag} is shown in Figure 6(a).

Since the structure G'_{diag} contains a new q-normal cycle, i.e., $z_{10} \xrightarrow{q} z_{10}$, it is then trimmed in the third iteration. Since event q is uncontrollable, event d at state z_5 is disabled. Steps 17 to 26 update the quiescent behavior at state z_5. For a marking M_7 in z_5, constraint $D'(M_7)$ is feasible. According to Steps 25, a new state $z = \{(M_{7,d}, N)\}$ is added to state set. By Step 26, an arc labeled event q from z_5 to z and a self-loop labeled event q at state z are also added, which is a new q-normal cycle.

In the forth iteration, to eliminate the q-normal cycle $z \xrightarrow{q} z$, the event c at state z_1 is disabled. After updating the quiescent behavior at state z_1, the final result is shown in Figure 6(b). Since there does not exist any indeterminate cycle and q-normal cycle in the current structure, Algorithm 2 terminates and outputs the final structure in Figure 6(b). □

Once $G_s = (Z, E \cup \{q\}, \delta_s, z_0)$ is obtained, the corresponding control policy is given as follows. Given an observation w whose consistent diagnostic state in diagnosability enforcing supervisor is z, the control decision $\xi(z)$ is to permit all controllable events that are defined at z, i.e.,

$$\xi(z) = \{e \in E_c \mid \delta_s(z, e) \text{ is defined}\}. \tag{5}$$

For example, for the net in Example 2, for an observation $w = abd$ whose consistent diagnostic state is z_3, according to the diagnosability enforcing supervisor in Figure 6(c), the control decision is $\xi(z_3) = \{d\}$, i.e., event c is disabled.

Remark 2 *It is worth noting that the observable quiescent event q provides us extra information about the current marking and the fault status of the system. Consider, for instance, the Q-diagnoser in Figure 6(a) in which the supervisor disables event d at diagnostic state z_6. When observing event sequence acd, one can infer that the plant may be at some marking either in the regular unobservable reach of M_8 while no fault transition has fired, or in the regular unobservable reach of M_9 while a fault transition has fired. However, by further observing quiescent event q and inspecting the new consistent diagnostic state z_{10}, one can infer that the system must be blocked at some marking in the regular unobservable reach of M_8 while no fault transition has fired, i.e., the firing a fault transition is excluded. In such a case, the supervisor may re-enable event d, which may lead to a more permissive control result. However, to re-enable events, all subsequent diagnostic states from z_{10} and the control decisions at those states need to be further explored, since the subsequence diagnostic states may not be already in G_{diag} and may contain new indeterminate cycles. To keep this paper focused, we do not address this issue and simply disable the precursor event that leads to unfaulty deadlocks.* □

Theorem 5 *Given an LPN $G = (PN, M_0, E, \ell)$ with $E = E_c \cup E_{uc}$ and a set of fault transitions T_f, the closed-loop system (G, ξ) is diagnosable and free of unfaulty deadlocks, where ξ is the control policy designed by Algorithm 2 and Eq. (5).*

Proof. Algorithm 2 ensures that the diagnosability enforcing supervisor does not contain any indeterminate cycle and q-normal cycle. Since the diagnosability enforcing supervisor represents the behavior of the closed-loop system, the closed-loop system is diagnosable and does not have unfaulty deadlocks. ■

Remark 3 *Note that in general there exist multiple ways to break an indeterminate cycle with controllable events in a Q-diagnoser. It may happen that the control action prunes some crucial arcs such that the state space is greatly reduced or the normal functionality is greatly affected. As a result, some additional information can be embedded into Algorithm 2 to avoid removing these crucial arcs. To explore this will be part of our future work.* □

6 Conclusion

This paper deals with the active diagnosis problem in the framework of LPNs. We generalize the notion of diagnosability to LPNs with deadlocks and control-induced deadlocks. A structure called quiescent basis reachability graph (QBRG) is proposed to characterize the behavior of a net containing deadlocks without enumerating all its reachable markings. An integer linear programming technique is developed to characterize the deadlocks. We present a QBRG-based method to design a diagnosability enforcing supervisor using a Q-diagnoser. Our supervisor guarantees that the closed-loop system is diagnosable and does not contain unfaulty deadlocks.

References

[1] F. Basile, P. Chiacchio, and G. D. Tommasi. On K-diagnosability of Petri nets via integer linear programming. *Automatica*, 48(9):2047–2058, 2012.

[2] F. Basile, G. D. Tommasi, and C. Sterle. Sensors selection for K-diagnosability of Petri nets via integer linear programming. In *Proc. 23rd Mediterranean Conference on Control and Automation (MED)*, pages 168–175, Torremolinos, Spain, 2015.

[3] F. Basile, G. D. Tommasi, and C. Sterle. Non-interference enforcement in bounded Petri nets. In *Proc. 2018 IEEE Conference on Decision and Control*, pages 4827–4832, Miami Beach, USA, 2018.

[4] F. Basile, G. D. Tommasi, and C. Sterle. Non-interference enforcement via supervisory control in bounded Petri nets. *IEEE Transactions on Automatic Control*, 2020, DOI: 10.1109/TAC.2020.3024274.

[5] N. Bertrand, É. Fabre, S. Haar, S. Haddad, and L. Hélouët. Active diagnosis for probabilistic systems. In *Proc. International Conference on Foundations of Software Science and Computation Structures*, pages 29–42, Grenoble, France, 2014.

[6] M. P. Cabasino, A. Giua, S. Lafortune, and C. Seatzu. A new approach for diagnosability analysis of Petri nets using verifier nets. *IEEE Transactions on Automatic Control*, 57(12):3104–3117, 2012.

[7] M. P. Cabasino, A. Giua, and C. Seatzu. Identification of Petri nets from knowledge of their language. *Discrete Event Dynamic Systems*, 17(4):447–474, 2007.

[8] M. P. Cabasino, A. Giua, and C. Seatzu. Fault detection for discrete event systems using Petri nets with unobservable transitions. *Automatica*, 46(9):1531–1539, 2010.

[9] M. P. Cabasino, A. Giua, and C. Seatzu. Diagnosability of discrete-event systems using labeled Petri nets. *IEEE Transactions on Automation Science and Engineering*, 11(1):144–153, 2014.

[10] M. P. Cabasino, S. Lafortune, and C. Seatzu. Optimal sensor selection for ensuring diagnosability in labeled Petri nets. *Automatica*, 49(8):2373–2383, 2013.

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

[12] F. Cassez and S. Tripakis. Fault diagnosis with static and dynamic observers. *Fundamenta Informaticae*, 88(4):497–540, 2008.

[13] E. Chanthery and Y. Pencolé. Monitoring and active diagnosis for discrete-event systems. In *Proc. 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, pages 1545–1550, Barcelona, Spain, 2009.

[14] J. Chen and R. Kumar. Failure detection framework for stochastic discrete event systems with guaranteed error bounds. *IEEE Transactions on Automatic Control*, 60(6):1542–1553, 2015.

[15] Z. Chen, F. Lin, C. Wang, L. Y. Wang, and M. Xu. Active diagnosability of discrete event systems and its application to battery fault diagnosis. *IEEE Transactions on Control Syetems Technology*, 22(5):1892–1898, 2014.

[16] A. Giua. Supervisory control of Petri nets with language specifications. In C. Seatzu, M. Silva, and J. van Schuppen, editors, *Control of Discrete-Event Systems*, volume 433, pages 235–255. Springer, London, U.K., 2013.

[17] A. Giua, S. Lafortune, and C. Seatzu. Divergence properties of labeled Petri nets and their relevance for diagnosability analysis. *IEEE Transactions on Automatic Control*, 65(7):3092–3097, 2020.

[18] A. Giua, C. Seatzu, and F. Basile. Observer-based state-feedback control of timed Petri nets with deadlock recovery. *IEEE Transactions on Automatic Control*, 49(1):17–29, 2004.

[19] K. Hernández-Rueda, M. E. Meda-Campaña, and J. Arámburo-Lizárraga. Enforcing diagnosability in interpreted Petri nets. *IFAC-PapersOnline*, 48(7):56–63, 2015.

[20] K. Hernández-Rueda, M. E. Meda-Campaña, and B. Haro-Martínez. Detección activa de faltas en sistemas de eventos discretos. *Pistas Educativas*, 39(128):730–748, 2018.

[21] Y. Hu, Z. Ma, and Z. Li. Active diagnosis of Petri nets using Q-diagnoser. In *Proc. 15th IEEE Conference on Automation Science and Engineering*, pages 203–208, Vancouver, Canada, 2019.

[22] Y. Hu, Z. Ma, and Z. Li. Design of supervisors for active diagnosis in discrete event systems. *IEEE Transactions on Automatic Control*, 65(12):5159–5172, 2020.

[23] S. Jiang, Z. Huang, V. Chandra, and R. Kumar. A polynomial algorithm for testing diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, 46(8):1318–1321, 2001.

[24] G. Jiroveanu and R. K. Boel. The diagnosability of Petri net models using minimal explanations. *IEEE Transactions on Automatic Control*, 55(7):1663–1668, 2010.

[25] D. Lefebvre and C. Delherm. Diagnosis of des with Petri net models. *IEEE Transactions on Automation Science and Engineering*, 4(1):114–118, 2007.

[26] Z. Li, M. Zhou, and N. Wu. A survey and comparison of Petri net-based deadlock prevention policies for flexible manufacturing systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):173–188, 2008.

[27] 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 Transactions on Automatic Control*, 62(3):1078–1093, 2017.

[28] Z. Ma, X. Yin, and Z. Li. Marking predictability and prediction in labeled Petri nets. *IEEE Transactions on Automatic Control*, 2020, DOI: 10.1109/TAC.2020.3024270.

[29] Z. Ma, X. Yin, and Z. Li. Marking diagnosability verification in labeled Petri nets. *Automatica*, 2021, in press.

[30] Z. Ma, G. Zhu, and Z. Li. Marking estimation in Petri nets using hierarchical basis reachability graphs. *IEEE Transactions on Automatic Control*, 66(2):810–817, 2020.

[31] M. V. Moreira, T. C. Jesus, and J. C. Basilio. Polynomial time verification of decentralized diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 56(7):1679–1684, 2011.

[32] G. T. Murata. Petri nets: properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[33] A. Ramírez-Treviño, E. Ruiz-Beltrán, J. Arámburo-Lizárraga, and E. López-Mellado. Structural diagnosability of DES and design of reduced Petri net diagnosers. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Syetems and Humans*, 42(2):416–429, 2012.

[34] A. Ramírez-Treviño, E. Ruiz-Beltrán, I. Rivera-Rangel, and E. López-Mellado. Online fault diagnosis of discrete event systems. a Petri net-based approach. *IEEE Transactions on Automation Science and Engineering*, 4(1):31–39, 2007.

[35] N. Ran, A. Giua, and C. Seatzu. Enforcement of diagnosability in labeled Petri nets via optimal sensor selection. *IEEE Transactions on Automatic Control*, 64(7):2997–3004, 2019.

[36] N. Ran, H. Su, A. Giua, and C. Seatzu. Codiagnosability analysis of bounded Petri nets. *IEEE Transactions on Automatic Control*, 63(4):1192–1199, 2018.

[37] Y. Ru, M. P. Cabasino, A. Giua, and C. N. Hadjicostis. Supervisor synthesis for discrete event systems with arbitrary forbidden state specifications. In *Proc. 47th IEEE Conference on Decision and Control*, pages 1048–1053, Cancun, Mexico, 2008.

[38] Y. Ru, M. P. Cabasino, A. Giua, and C. N. Hadjicostis. Supervisor synthesis for discrete event systems under partial observation and arbitrary forbidden state specifications. *Discrete Event Dynamic Systems*, 24(3):275–307, 2014.

[39] E. Ruiz-Beltrán, A. Ramírez-Treviño, and J. L. Orozco-Mora. Fault diagnosis in Petri nets. In J. Campos, C. Seatzu, and Xiaolan Xie, editors, *Formal Methods in Manufacturing*, chapter 22, pages 627–651. CRC Press, Boca Raton, 2014.

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

[41] M. Sampath, R. Sengupta, S. Lafortune, and K. Sinnamohideen. Diagnosability of discrete-event system. *IEEE Transactions on Automatic Control*, 40(9):1555–1575, 1995.

[42] M. Schmidt and J. Lunze. Active diagnosis of deterministic I/O automata. In *Proc. 4th IFAC Workshop on Dependable Control of Discrete Systems*, pages 79–84, York, UK, 2013.

[43] W. Wang, S. Lafortune, A. R. Girard, and F. Lin. Optimal sensor activation for diagnosing discrete event systems. *Automatica*, 46(7):1165–1175, 2010.

[44] Y. Wu and C. N. Hadjicostis. Algebraic approaches for fault identification in discrete-event systems. *IEEE Transactions on Automatic Control*, 50(12):2048–2055, 2005.

[45] X. Yin, J. Chen, Z. Li, and S. Li. Robust fault diagnosis of stochastic discrete event systems. *IEEE Transactions on Automatic Control*, 64(10):4237–4244, 2019.

[46] X. Yin and S. Lafortune. A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems. *IEEE Transactions on Automatic Control*, 61(8):2140–2154, 2016.

[47] X. Yin and S. Lafortune. On the decidebility and complexity of diagnosability for labeled Petri nets. *IEEE Transactions on Automatic Control*, 62(11):5931–5938, 2017.

[48] T-S. Yoo and S. Lafortune. Polynomial-time verification of diagnosability of partially observed discrete-event systems. *IEEE Transactions on Automatic Control*, 47(9):1491–1495, 2002.