Critical Observability Verification and Enforcement of Labeled Petri Nets by Using Basis Markings

Citation: X. Cong, M. P. Fanti, A. M. Mangini and Z. Li, "Critical Observability Verification and Enforcement of Labeled Petri Nets by Using Basis Markings," in IEEE Transactions on Automatic Control, vol. 68, no. 12, pp. 8158-8164, Dec. 2023, doi: 10.1109/TAC.2023.3292747.

Critical Observability Verification and Enforcement of Labeled Petri Nets by Using Basis Markings

Xuya Cong, *Member, IEEE,* Maria Pia Fanti, *Fellow, IEEE,* Agostino Marcello Mangini, *Member, IEEE,* and Zhiwu Li, *Fellow, IEEE*

***Abstract*—A discrete event system is said to be critically observable if the observer can always determine whether the current state necessarily belongs to a set of critical states. This paper focuses on two issues related to the safety and security of discrete event systems, namely critical observability verification and enforcement of labeled Petri nets. First, given a bounded net, we verify its critical observability by using basis markings and solving some integer linear programming problems, thus avoiding the enumeration of the full state space of a net system. Moreover, for a non-critically observable net system, we obtain a feasible stop-free event set from a twin basis reachability graph such that a valid control policy can be always found, if the feasible stop-free event set is non-empty. Finally, according to the feasible stop-free event set, a set of disabled edges is generated, and an online control policy is developed based on the supervisory control theory, which guarantees that the closed-loop system is critically observable and deadlock-free.**

***Index Terms*—Discrete event system, Petri net, critical observability, supervisory control.**

I. Introduction

With the wide applications of the cyber-physical systems, security and safety become important in safety-critical applications such as air traffic management systems, where it is usually necessary to check whether the current behavior falls into the dangerous or undesired situations. Inspired by the safety-critical applications in real practice, the notion of critical observability is initially proposed by De Santis *et al.* [1] for linear switching systems.

In the framework of discrete event systems (DESs), the work in [2] computes all the state pairs to verify critical observability for finite state automata. By using critical observers, Pola *et al.* [3] check critical observability for networks of finite state automata. Lai *et al.* [4] introduce a new observer to check critical observability for max-plus automata. A recent work in [5] studies the computational complexity of checking critical observability for both (networks of) finite state automata and PNs. Moreover, the work in [7] touches upon the critical observability for (networks of) finite state automata at a given time step. Recently, our paper [6] uses integer linear programming (ILP) to check critical observability for a bounded and live labeled Petri net (LPN).

For a non-critically observable system, an interesting and significant issue is to enforce the critical observability of the system. According to the notions of critical observability, the sensor selection or activation mechanism [8–10] for diagnosability or prognosability enforcement can also be adopted to enforce critical observability. By using such methods, it is necessary to deploy additional new sensors in a net system, which is not economic or even impractical. In this case, supervisory control theory [11] is selected as a feasible avenue to enforce diagnosability of an LPN without adding new sensors to a plant. Based on the supervisory control theory, several works [12–14] deal with diagnosability enforcement, called active diagnosis in the framework of automata. More precisely, Sampath *et al.* [12] initially propose an approach to break the indeterminate cycles in a diagnoser. Yin and Lafortune [13] develop a supervisor that can achieve the safety, non-blockingness, and maximal permissiveness of the closed-loop system, which is useful in diagnosability enforcement. Moreover, Hu *et al.* [14] present an online control policy to enforce diagnosability by exploiting the structure of a verifier and a feasible stop-free set in the verifier.

Different from an automaton model, few works [16, 17] address the problem of active diagnosis in the framework of PNs that have become a standard model for the study of DESs. In particular, the work in [16] presents a method to enforce diagnosability of the interpreted PNs by using a regulation circuit controller. In addition, a recent work in [17] develops a new structure called a quiescent basis reachability graph (QBRG), and builds its corresponding Q-diagnoser to obtain a diagnosability enforcing supervisor for the LPNs.

This paper proposes two new methodologies to verify and enforce critical observability of a bounded LPN system by using basis markings. Hence, the aim of the paper is to address the research gap between automaton models and PN languages that have been largely defined to represent the behaviour of DESs [17, 19]. The main contributions of this paper are outlined as follows.

1) An approach is proposed to check critical observability by using basis markings and solving some ILP problems, which circumvents enumerating all the reachable markings or all the minimal T-semiflows of an LPN.

This work is supported by the National Key R&D Program of China under Grant 2018TYB1700104, the Natural Science Foundation of Shannxi Province under Grant 2022JQ–606 and the Research Plan of Department of Education of Shaanxi Province under Grant 21JK0752. (*Corresponding author: Zhiwu Li*)

X. Cong is with the Youth Innovation Team of Shaanxi Universities, College of Computer Science and Technology, Xi'an University of Science and Technology, Xi'an 710054, China (email: congxuya@163.com).

M. Fanti and A. Mangini are with the Department of Electrical and Information Engineering, Polytechnic of Bari, 70125 Bari, Italy (email: mariapia.fanti@poliba.it; agostinomarcello.mangini@poliba.it).

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

2) To the best of our knowledge, it is the first work to enforce critical observability of LPNs, which has not been studied in the literature.
3) By computing a feasible stop-free event set in a twin basis reachability graph (twin-BRG), the closed-loop system is critically observable and deadlock free. Moreover, we also prove that the closed-loop system is maximally permissive under a particular condition.
4) By applying the proposed online control policy, we avoid the construction of a whole observer for designing a supervisor.

The remainder of this paper is organized as follows. Section II recalls some basics of LPNs and BRGs. Section III presents a method to verify critical observability based on twin-BRGs. In Section IV, we formulate the problem of critical observability enforcement of LPNs. In Section V, we analyze a stop-free event set and propose a method to compute a feasible stop-free event set. Section VI provides an efficient online control policy to enforce critical observability. Finally, Section VII concludes the paper.

II. Preliminaries

A. Labeled Petri Nets

Let $\mathbb{N}$ be a set of non-negative integers. A PN [18] is a four-tuple $PN = (P, T, Pre, Post)$, where P is a set of m places, T is a set of n transitions with $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$, $Pre : P \times T \to \mathbb{N}$ and $Post : P \times T \to \mathbb{N}$ are the pre- and post-incidence functions that specify the arcs from places to transitions and transitions to places in a net, respectively. In general, Pre ($Post$) can be represented by an $m \times n$ matrix indexed by P and T. The incidence matrix of a PN is defined as $C = Post - Pre$. Given a place $p \in P$, its input and output sets are defined as ${}^\bullet p = \{t \in T | Post(p,t) > 0\}$ and $p^\bullet = \{t \in T | Pre(p,t) > 0\}$, respectively. The notions for ${}^\bullet t$ and $t^\bullet$ are analogously defined. An acyclic PN indicates that there are no oriented cycles.

A marking is a function $M : P \to \mathbb{N}$ that assigns to each place a non-negative integer number of tokens. Similarly, a marking M can be represented by an m-dimensional vector indexed by P. A PN system $\langle PN, M_0 \rangle$ is a net PN with an initial marking M_0. A transition t is enabled at M if $M \geq Pre(\cdot, t)$. The firing of a transition t at M leads to marking $M' = M + C(\cdot, t)$, which is denoted as $M[t\rangle M'$.

Marking M is said to be reachable from M_0 if there exist a transition sequence $\sigma = t_1 t_2 \cdots t_k$ and markings $M_1, M_2, \ldots, M_{k-1}$ such that $M_0[t_1\rangle M_1[t_2\rangle M_2 \cdots M_{k-1}[t_k\rangle M$ holds, denoted as $M_0[\sigma\rangle M$. We write it as $M_0[\sigma\rangle$ if the destination marking is of no interest. The vector $\vec{y}_\sigma$ is the firing vector of σ, i.e., $\vec{y}_\sigma(t) = k$ if transition t appears k times in σ, where T^* is the Kleene-closure [19] of T. The set of all reachable markings from M_0 is denoted as $R(PN, M_0)$. A net system $\langle PN, M_0 \rangle$ is said to be bounded if the set $R(PN, M_0)$ is finite. A marking M is a deadlock marking if no transition can fire at this marking, and $\langle PN, M_0 \rangle$ is said to be deadlock-free if there is no deadlock marking in $R(PN, M_0)$. The set of all sequences that can fire in $\langle PN, M_0 \rangle$ is defined as $L(PN, M_0) = \{\sigma \in T^* | M_0[\sigma\rangle\}$.

An LPN system is a four-tuple $G = (PN, M_0, E, \lambda)$, where $\langle PN, M_0 \rangle$ is a PN system, E is an alphabet (a finite set of labels) and $\lambda : T \to E \cup \{\varepsilon\}$ is a labeling function that assigns to each transition $t \in T$ either a symbol from E or the empty word ε. Moreover, T is partitioned into $T = T_o \cup T_{uo}$ with $T_o \cap T_{uo} = \emptyset$, where T_o (resp. T_{uo}) is the set of n_o (resp. n_{uo}) observable (resp. unobservable) transitions. For any transition $t \in T$, if $t \in T_o$, then $\lambda(t) \in E$; otherwise $\lambda(t) = \varepsilon$. In an LPN, the same label can be associated with more than one transition. The unobservable reach of M is defined as $\mathcal{U}(M) = \{M' \in \mathbb{N}^m | \exists \sigma_u \in T_{uo}^* : M[\sigma_u\rangle M'\}$. Given an event $e \in E$, $T(e)$ represents its corresponding set of transitions, i.e., $T(e) = \{t \in T | \lambda(t) = e\}$. Moreover, given an event set $E_i \subseteq E$, we use $T(E_i)$ to denote the set of transitions associated with E_i, i.e., $T(E_i) = \{t \in T | \lambda(t) \in E_i\}$.

Let w denote the label sequence associated with a transition sequence $\sigma \in T^*$ such that $w = \lambda(\sigma)$ by extending the labeling function as $\lambda : T^* \to E^*$. Given an LPN system $G = (PN, M_0, E, \lambda)$, we define the language generated by system G as $\mathcal{L}(G) = \{w \in E^* | \exists \sigma \in L(PN, M_0) : \lambda(\sigma) = w\}$. For a word $w \in \mathcal{L}(G)$, let $\mathcal{S}(w) = \{\sigma \in L(PN, M_0) | \lambda(\sigma) = w\}$ and $\mathcal{C}(w) = \{M \in \mathbb{N}^m | \exists \sigma \in \mathcal{S}(w) : M_0[\sigma\rangle M\}$ denote the set of firing sequences consistent with w and the set of markings consistent with w, respectively.

Given an LPN system $G = (PN, M_0, E, \lambda)$ with $PN = (P, T, Pre, Post)$, we denote by $PN' = (P, T', Pre', Post')$ the unobservable subnet of G, i.e., the net is obtained by removing all transitions in T_o from PN, where Pre' and $Post'$ are the restrictions of Pre and $Post$ to T_{uo}, respectively. The incidence matrix of an unobservable subnet is denoted by $C_u = Post' - Pre'$.

B. Basis Reachability Graph

Definition 1: [20] Given a marking M and an observable transition $t \in T_o$, we define:

- $\Sigma(M, t) = \{\sigma \in T_{uo}^* | \exists M' \in \mathbb{N}^m : M[\sigma\rangle M' \wedge M' \geq Pre(\cdot, t)\}$ as the set of explanations of t at M and $Y(M, t) = \{\vec{y}_\sigma \in \mathbb{N}^{n_{uo}} | \exists \sigma \in \Sigma(M, t)\}$ as the set of e-vectors;
- $\Sigma_{min}(M, t) = \{\sigma \in \Sigma(M, t) | \nexists \sigma' \in \Sigma(M, t) : \vec{y}_{\sigma'} \lneqq \vec{y}_\sigma\}$ as the set of minimal explanations of t at M and $Y_{min}(M, t) = \{\vec{y}_\sigma \in \mathbb{N}^{n_{uo}} | \exists \sigma \in \Sigma_{min}(M, t)\}$ as the corresponding set of minimal e-vectors.

Definition 2: [21] Given an LPN $G = (PN, M_0, E, \lambda)$, its set of basis markings $\mathcal{M}_B$ is a subset of $R(PN, M_0)$ such that

1) $M_0 \in \mathcal{M}_B$;
2) $\forall M \in \mathcal{M}_B$, $\forall t \in T_o$, $\forall \vec{y}_\sigma \in Y_{min}(M, t)$, $M' = M + C(\cdot, t) + C_u \cdot \vec{y}_\sigma \Rightarrow M' \in \mathcal{M}_B$.

The BRG $\mathcal{B} = (\mathcal{M}_B, E, \delta, M_0)$ of an LPN $G = (PN, M_0, E, \lambda)$ is an NFA computed by Algorithm 1 in [21], where the state set is the set of basis markings $\mathcal{M}_B$, the event set is the set of labels of G, the transition relation is $\delta : \mathcal{M}_B \times E \to 2^{\mathcal{M}_B}$, and the initial state is the initial marking M_0. The domain of δ can be extended from $\mathcal{M}_B \times E$ to $\mathcal{M}_B \times E^*$ in the usual way. In the following, let $\Gamma(M)$ denote the set of active events at a basis marking M, i.e.,

$\Gamma(M) = \{e \in E | \delta(M, e) \text{ is defined}\}$. For a word $w \in \mathcal{L}(G)$, we use $\mathcal{C}_b(w)$ to denote the basis marking set consistent with w, i.e., $\mathcal{C}_b(w) = \mathcal{C}(w) \cap \mathcal{M}_B$. The complexity of building a BRG is still exponential w.r.t. the number of places and and the number of tokens in the initial marking [21]. However, due to the existence of unobservable transitions, the size of BRG is in general much smaller than that of reachability graph.

III. Critical Observability Verification of LPNs

This section considers the critical observability verification of the LPNs. We first present the following two assumptions that are useful for the application of the BRG-based approach.

A1) An LPN G is bounded;

A2) The unobservable subnet of G is acyclic.

Assumption A1 ensures the finiteness of the BRG of a plant. In addition, Assumption A2 makes the state equation serve as a necessary and sufficient condition for marking reachability. In this paper, a set of critical markings $C_R \subseteq \mathbb{N}^m$ is defined as a set of discrete markings or a general mutual exclusive constraint [22].

Definition 3: [5] Let us consider an LPN $G = (PN, M_0, E, \lambda)$ and a set of critical markings C_R. G is said to be *critically observable* if $[\mathcal{C}(w) \subseteq C_R] \vee [\mathcal{C}(w) \subseteq R(PN, M_0) \setminus C_R]$, for all $w \in \mathcal{L}(G)$.

Based on Definition 3, given an LPN and a set of critical markings, we introduce the following definition that is useful to check critical observability of the LPN.

Definition 4: Given an LPN $G = (PN, M_0, E, \lambda)$, its BRG $\mathcal{B} = (\mathcal{M}_B, E, \delta, M_0)$, and a set of critical markings C_R, the set of fully-critical basis markings is defined as $\mathcal{F} = \{M \in \mathcal{M}_B | \mathcal{U}(M) \subseteq C_R\}$, the set of partially-critical basis markings is defined as $\mathcal{P} = \{M \in \mathcal{M}_B | \mathcal{U}(M) \cap C_R \neq \emptyset \wedge \mathcal{U}(M) \setminus C_R \neq \emptyset\}$, and the set of non-critical basis markings is defined as $\mathcal{N} = \{M \in \mathcal{M}_B | \mathcal{U}(M) \cap C_R = \emptyset\}$.

By Definition 4, the set of basis markings $\mathcal{M}_B$ is divided into three disjoint subsets. More precisely, a fully-critical basis marking means that it can only reach a critical marking by firing an unobservable transition sequence, a partially-critical basis marking implies that not all but some markings in its unobservable reach are critical, and a non-critical basis marking represents that none of the markings in its unobservable reach is critical.

Given a basis marking set $\mathcal{M}_B$ and a critical marking set C_R, we present a proposition that is applicable to compute the sets of fully-critical, partially-critical, and non-critical basis markings.

Proposition 1: Given an LPN $G = (PN, M_0, E, \lambda)$, its BRG $\mathcal{B} = (\mathcal{M}_B, E, \delta, M_0)$, and a set of critical markings $C_R = \{M_c^1, \ldots, M_c^q\}$, a basis marking $M \in \mathcal{M}_B$ is fully-critical *iff* the integer constraint set (1) is infeasible, a basis marking $M \in \mathcal{M}_B$ is partially-critical *iff* the integer constraint sets (1) and (2) are feasible, and a basis marking $M \in \mathcal{M}_B$ is non-critical *iff* the integer constraint set (2) is infeasible:

$$\begin{cases} \bigwedge\limits_{M_c^k \in C_R} M + C_u \cdot \vec{y}_\sigma \neq M_c^k, \\ M + C_u \cdot \vec{y}_\sigma \geq \vec{0}, \\ \vec{y}_\sigma \in \mathbb{N}^{n_{uo}} \end{cases} \tag{1}$$

$$\begin{cases} \bigvee\limits_{M_c^k \in C_R} M + C_u \cdot \vec{y}_{\sigma'} = M_c^k, \\ \vec{y}_{\sigma'} \in \mathbb{N}^{n_{uo}}. \end{cases} \tag{2}$$

Proof: See the supplementary material in [23]. □

Note that Eq. (1) can be linearized by the technique in [6], and the logic *OR* condition in Eq. (2) can be linearized by the technique in [26]. In this sense, Eqs. (1) and (2) can be represented as integer linear constraints. In fact, since Eqs. (1) and (2) consider all the critical markings, we need to solve at most $2|\mathcal{M}_B|$ ILP problems of Eqs. (1) and (2) for the computation of the sets $\mathcal{F}$, $\mathcal{P}$, and $\mathcal{N}$. Moreover, a critical marking set C_R can be usually described as a generalized mutual exclusion constraint $C_R = \{M \in \mathbb{N}^m | w^T \cdot M \leq k\}$, where $w \in \mathbb{Z}^m$ and $k \in \mathbb{Z}$ ($\mathbb{Z}$ is the set of integer numbers). In this case, for a basis marking $M \in \mathcal{M}_B$, the first constraint in Eqs. (1) and (2) can be rewritten as $w^T \cdot (M + C_u \cdot \vec{y}_\sigma) \geq k + 1$ and $w^T \cdot (M + C_u \cdot \vec{y}_{\sigma'}) \leq k$, respectively.

In the following, we show a sufficient condition to check critical observability of an LPN by analyzing its BRG.

Proposition 2: Given an LPN $G = (PN, M_0, E, \lambda)$, its BRG $\mathcal{B} = (\mathcal{M}_B, E, \delta, M_0)$, and a set of critical markings C_R, G is critically observable if there is no partially-critical basis marking in the BRG $\mathcal{B}$ of G.

Proof: See the supplementary material in [23]. □

In order to obtain a necessary and sufficient condition to check critical observability, we can use the twin-plant structure [24] that is shown in Definition 5.

Definition 5: Given an LPN $G = (PN, M_0, E, \lambda)$ and its BRG $\mathcal{B} = (\mathcal{M}_B, E, \delta, M_0)$, a twin-BRG is an NFA $B = (X, E, \delta_{tw}, x_0)$, where $X \subseteq \mathcal{M}_B \times \mathcal{M}_B$, $x_0 = (M_0, M_0)$, and the transition relation is defined as:

1) $X = \{(M_0, M_0)\}$;
2) for all $(M', M'') \in X$,

$$\begin{cases} (M', e, \bar{M}') \in \delta \\ (M'', e, \bar{M}'') \in \delta \end{cases} \Rightarrow \begin{cases} \bar{x} \in X \\ ((M', M''), e, \bar{x}) \in \delta_{tw}{}^{1}) \end{cases} \tag{3}$$

where $\bar{x} = (\bar{M}', \bar{M}'')^2$.

Definition 6: Given an LPN G and a critical marking set C_R, a state $x = (M, M')$ in the twin-BRG of G is said to be critical observability violative if one of the following two conditions holds:

1) $M \in \mathcal{P}$;
2) $M \in \mathcal{F}$ and $M' \in \mathcal{N}$.

Note that since (M, M') is an unordered pair, it does not specify the first or the second basis marking in Definition 6. In the following, given a twin-BRG $B = (X, E, \delta_{tw}, x_0)$, the set of critical observability violative states is denoted as X_N, and the set of states that does not violate critical observability is denoted as X_Y with $X_Y = X \setminus X_N$.

Proposition 3: Given an LPN $G = (PN, M_0, E, \lambda)$, its twin-BRG $B = (X, E, \delta_{tw}, x_0)$, and a critical marking set C_R, G is critically observable *iff* there is no critical observability violative state in the twin-BRG B of G.

[1]For brevity, the domain of δ_{tw} can be extended from $X \times E$ to $X \times E^*$ in the usual way.

[2]Since the order of two basis markings does not matter to the problem, $(\bar{M}', \bar{M}'')$ is defined as an unordered pair, i.e., $(\bar{M}', \bar{M}'')$ and $(\bar{M}'', \bar{M}')$ are equivalent.

Proof: See the supplementary material in [23]. □

In particular, for an LPN, if the set of fully-critical basis markings or the set of non-critical basis markings is empty, we can derive a necessary and sufficient condition for critical observability by only analyzing its BRG as follows.

Corollary 1: Given an LPN $G = (PN, M_0, E, \lambda)$, its BRG $\mathcal{B} = (\mathcal{M}_B, E, \delta, M_0)$, and a critical marking set C_R, suppose that $\mathcal{F} = \emptyset$ or $\mathcal{N} = \emptyset$. G is critically observable *iff* there is no partially-critical basis marking in the BRG $\mathcal{B}$ of G.

Proof: It directly follows from Propositions 2 and 3. □

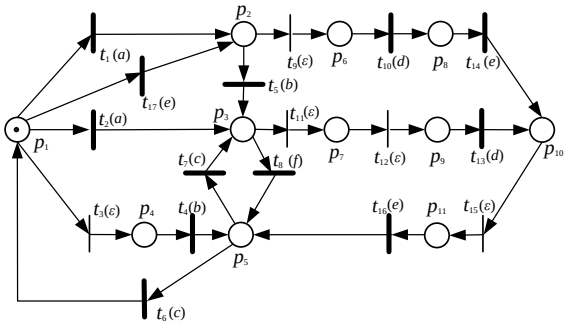

Fig. 1: An LPN G in Examples 1, 2, and 3.

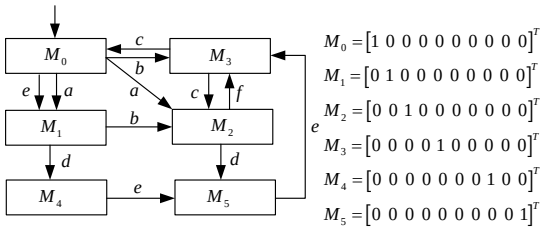

Fig. 2: The BRG of the LPN in Fig. 1.

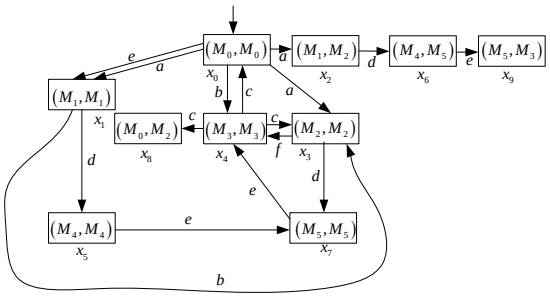

Fig. 3: The twin-BRG of the LPN in Fig. 1.

Example 1: Consider the LPN G in Fig. 1, where $T_o = \{t_1, t_2, t_4, t_5, t_6, t_7, t_8, t_{10}, t_{13}, t_{14}, t_{16}, t_{17}\}$ with $T(a) = \{t_1, t_2\}$, $T(b) = \{t_4, t_5\}$, $T(c) = \{t_6, t_7\}$, $T(d) = \{t_{10}, t_{13}\}$, $T(e) = \{t_{14}, t_{16}, t_{17}\}$, and $T(f) = \{t_8\}$. The BRG and twin-BRG of G in Fig. 1 are shown in Figs. 2 and 3, respectively. Given a critical marking set $C_R = \{M \in \mathbb{N}^{11} | -M(p_{10}) - M(p_{11}) \leq -1\}$, by Proposition 1, the sets of fully-critical, partially-critical, and non-critical basis markings are $\mathcal{F} = \{M_5\}$, $\mathcal{P} = \emptyset$, and $\mathcal{N} = \{M_0\text{-}M_4\}$, respectively. The result in this example illustrates that Proposition 2 is not necessary for critical observability: even though there is no partially-critical basis marking in the BRG, due to the existence of two critical observability violative states $x_6 = (M_4, M_5)$ and $x_9 = (M_5, M_3)$ in the twin-BRG, the LPN is not critically observable based on Proposition 3. □

IV. Critical Observability Enforcement Formulation of LPNs

In the supervisory control framework of LPNs [25], the event set E of a given LPN is divided into two disjoint subsets E_{uc} and E_c, where the events in E_{uc} are uncontrollable and those in E_c are controllable. In particular, if a transition sequence $\sigma \in L(PN, M_0)$ occurs in a plant net, the observer observes the corresponding event sequence $w = \lambda(\sigma)$ and obtains a set of consistent basis markings $\mathcal{C}_b(w)$. According to $\mathcal{C}_b(w)$, a supervisor permits a subset of controllable events $\zeta(\mathcal{C}_b(w)) \subseteq E_c$ to be executed, while the occurrence of other events in $E_c \setminus \zeta(\mathcal{C}_b(w))$ is forbidden. Moreover, the supervisor cannot disable any transition with uncontrollable label. Once an event is disabled, all the transitions associated with this event are disabled. The scheme of critical observability enforcement is shown in Fig. 4.

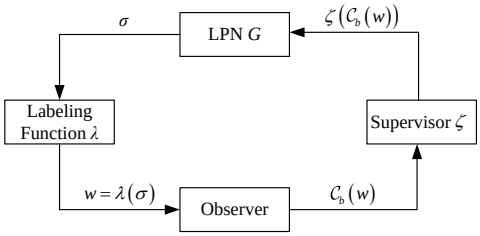

Fig. 4: Critical observability enforcement scheme.

Let (G, ζ) denote the closed-loop system. The language generated by (G, ζ) denoted by $\mathcal{L}(G, \zeta)$, is defined recursively as follows:

1) $\varepsilon \in \mathcal{L}(G, \zeta)$; and
2) $[w \in \mathcal{L}(G, \zeta) \wedge we \in \mathcal{L}(G) \wedge e \in \zeta(\mathcal{C}_b(w)) \cup E_{uc}] \Leftrightarrow [we \in \mathcal{L}(G, \zeta)]$.

Now, the following part of the paper is to design a control policy ζ for a non-critically observable LPN G such that the closed-loop system (G, ζ) is critically observable.

Furthermore, we assume that a plant net to be controlled satisfies the following two assumptions.

A3) The LPN is deadlock-free.

A4) All the unobservable transitions are uncontrollable.

Assumption A3 is made for technical convenience only. In fact, once a plant is deadlocked, it can be converted to a deadlock free net by adding a quiescent event (or stop event) [14, 17]. Moreover, the deadlock-freeness of a plant net is a precondition of the deadlock-freeness of the controlled system in this paper. Assumption A4 is widely used in the context of supervisory control of partially observed DESs. Under Assumption A4, we can omit all the arcs labeled with unobservable transitions and all the markings reached by firing unobservable transitions in the supervisor by using the basis marking technique.

V. Stop-free Control Policy

A. Stop-free Event Set Analysis

In this paper, we explore a control policy ζ to make the closed-loop system critically observable and deadlock-free. However, the control action at a set of basis markings consistent with a word may cause a control-induced deadlock. In this case, we formally define a stop-free control policy ζ to ensure the deadlock-freeness of the closed-loop system as follows.

Definition 7: Given an LPN $G = (PN, M_0, E, \lambda)$ with $E = E_c \cup E_{uc}$, a control policy ζ is stop-free if for any word $w \in \mathcal{L}(G)$ and any basis marking $M \in \mathcal{C}_b(w)$, the following condition holds: by firing any unobservable transition sequence $\sigma_u \in T_{uo}^*$ at M, there exists a transition $t \in T_{uo} \cup T((\zeta(\mathcal{C}_b(w)) \cup E_{uc}) \cap \Gamma(M))$ that satisfies $M[\sigma_u t\rangle$, where $\zeta(\mathcal{C}_b(w))$ is the set of controllable events permitted by the control policy ζ at $\mathcal{C}_b(w)$.

Note that even though a basis marking has an active event in the BRG, this marking may enter a deadlock after firing an unobservable transition sequence. Thus, in Definition 7, we provide a rule to ensure that no deadlock will occur at any set of basis markings $\mathcal{C}_b(w)$ after the control action $\zeta(\mathcal{C}_b(w))$ is taken, i.e., no deadlock can occur in the closed-loop system.

Definition 8: Given an LPN $G = (PN, M_0, E, \lambda)$ with $E = E_c \cup E_{uc}$, $\hat{E}_c$ is a stop-free event set if $[\hat{E}_c \subseteq E_c] \wedge (\forall \sigma_u \in T_{uo}^* \forall M \in \mathcal{M}_B)[M[\sigma_u\rangle \Rightarrow (\exists t \in T_{uo} \cup T(E \setminus \hat{E}_c \cap \Gamma(M)))\ M[\sigma_u t\rangle]$.

Definition 8 defines a subset of controllable events $\hat{E}_c \subseteq E_c$ such that the disablement of any event $e \in \hat{E}_c$ can never lead to a deadlock, and a supervisor is designed to only disable some controllable events in $\hat{E}_c$ but not any event in $E \setminus \hat{E}_c$.

Proposition 4: Given a stop-free event set $\hat{E}_c$ in an LPN G, the closed-loop system (G, ζ) is deadlock-free if ζ never disables any event $e \notin \hat{E}_c$.

Proof: By contradiction, suppose that there is a deadlock marking in the closed-loop system (G, ζ). After the control action $\zeta(\mathcal{C}_b(w))$ with $M \in \mathcal{C}_b(w)$ is taken, by firing an unobservable transition sequence $\sigma_u \in T_{uo}^*$ at M, the LPN enters a deadlock, which contradicts the fact that for all $\sigma_u \in T_{uo}^*, M[\sigma_u\rangle$ implies that there exists a transition $t \in T_{uo} \cup T(E \setminus \hat{E}_c \cap \Gamma(M))$ such that $M[\sigma_u t\rangle$ holds. □

The stop-free event set is not unique in general. However, not all stop-free event sets are feasible for designing a control policy to enforce critical observability. In the next subsection, we provide an algorithm to compute a feasible stop-free event set to ensure critical observability of the closed-loop system. Given a stop-free event set $\hat{E}_c$, a control policy ζ is said to be stop-free w.r.t. $\hat{E}_c$, if it never disables any event not in $\hat{E}_c$.

Moreover, given a stop-free control policy ζ w.r.t. $\hat{E}_c$, in order to avoid the closed-loop system uncontrollably entering a critical observability violative state, we define $X_{good} = \{x \in X_Y | \nexists w \in (E \setminus \hat{E}_c)^* : \delta_{tw}(x, w) \in X_N\}$ as the set of states that cannot reach a state in X_N via an event sequence $w \in (E \setminus \hat{E}_c)^*$, where $w \in (E \setminus \hat{E}_c)^*$ is an event sequence that cannot be disabled by ζ.

Proposition 5: Given an LPN $G = (PN, M_0, E, \lambda)$ with $E = E_c \cup E_{uc}$, a critical marking set C_R, and a stop-free control policy ζ w.r.t. a stop-free event set $\hat{E}_c$, the closed-loop system (G, ζ) is critically observable *iff* none of the set of consistent basis markings of (G, ζ) includes a state in $X \setminus X_{good}$[1].

Proof: *(If)* By contrapositive, if there exists a set of consistent basis markings $\mathcal{C}_b(w)$ of (G, ζ) that includes a state $x \in X \setminus X_{good}$, then there exists an event sequence $w' \in (E \setminus \hat{E}_c)^*$ from x to x' such that x' is a critical observability violative state in X_N. In addition, due to $w' \in (E \setminus \hat{E}_c)^*$, it cannot be forbidden by the control policy ζ. Based on Proposition 3, (G, ζ) is not critically observable.

(Only if) By contrapositive, if (G, ζ) is not critically observable, based on Proposition 3, there exists a critical observability violative state $x \in X_N$ included in a set of consistent basis markings $\mathcal{C}_b(w)$ of (G, ζ). Since the state $x \in X_N$ is also in the set $X \setminus X_{good}$, the conclusion holds. □

B. Computation of a Feasible Stop-free Event Set

In order to enforce critical observability of a plant net, by Proposition 5, it is necessary to forbid a system to reach any set of consistent basis markings that includes states in $X \setminus X_{good}$. The following proposition provides a condition for the existence of a control policy w.r.t. $\hat{E}_c$.

Proposition 6: Given an LPN $G = (PN, M_0, E, \lambda)$ with $E = E_c \cup E_{uc}$, a critical marking set C_R, and a stop-free event set $\hat{E}_c \subseteq E_c$, there exists a stop-free control policy ζ w.r.t. $\hat{E}_c$ such that (G, ζ) is critically observable *iff* the initial state x_0 in the twin-BRG $B = (X, E, \delta_{tw}, x_0)$ of G is a state in X_{good}.

Proof: *(If)* By contrapositive, suppose that there exists no stop-free control policy ζ w.r.t. $\hat{E}_c$ such that (G, ζ) is critically observable. By Proposition 3, there always exists a critical observability violative state that can be inevitably reached from the initial state x_0 in the twin-BRG of (G, ζ). That is to say, the initial state x_0 is not in the set X_{good}.

(Only if) By contrapositive, suppose that the initial state x_0 is in the set $X \setminus X_{good}$. There exists an event sequence $w \in (E \setminus \hat{E}_c)^*$ from x_0 to x such that x is a critical observability violative state. Since the sequence w cannot be disabled by a supervisor, there exists no stop-free control policy ζ w.r.t. $\hat{E}_c$ such that (G, ζ) is critically observable. □

In order to find a suitable control policy ζ that ensures critical observability of the closed-loop system, we introduce the feasibility of a stop-free event set $\hat{E}_c$ as follows: a stop-free event set $\hat{E}_c$ is feasible, if the initial state of the twin-BRG is a state in the set X_{good}.

Given a twin-BRG $B = (X, E, \delta_{tw}, x_0)$ and a subset of events $E' \subseteq E$, an E'-induced twin-BRG $B_{E'} = (X', E', \delta'_{tw}, x_0)$ is a sub-graph of B by removing all edges labeled with event $e \in E \setminus E'$, followed by removing all the unreachable states.

Based on Proposition 6, the feasibility of a stop-free event set $\hat{E}_c$ implies the existence of a stop-free control policy ζ w.r.t. $\hat{E}_c$ such that (G, ζ) is critically observable. Then, we

[1] A set of consistent basis markings $\mathcal{C}_b(w)$ is said to include a state $x = (M_i, M_j)$ in the twin-BRG, if $M_i \in \mathcal{C}_b(w)$ and $M_j \in \mathcal{C}_b(w)$ hold, and we write it as $x \subseteq \mathcal{C}_b(w)$.

introduce a property for the stop-freeness and feasibility, which is useful to design an algorithm for the computation of a feasible stop-free event set.

Proposition 7: Given two sets $\hat{E}_1, \hat{E}_2 \subseteq E_c$ with $\hat{E}_1 \subseteq \hat{E}_2$:

1) If $\hat{E}_2$ is stop-free, then $\hat{E}_1$ is stop-free;

2) If $\hat{E}_1$ is feasible, then $\hat{E}_2$ is feasible.

Proof: Suppose that $\hat{E}_2$ is a stop-free event set. Due to $\hat{E}_1 \subseteq \hat{E}_2$, we have $(E_c \setminus \hat{E}_2) \subseteq (E_c \setminus \hat{E}_1)$. That is to say, if the condition $(\forall \sigma_u \in T_{uo}^*)[M[\sigma_u\rangle \Rightarrow (\exists t \in T_{uo} \cup T(E \setminus \hat{E}_2 \cap \Gamma(M)))M[\sigma_u t\rangle]$ holds, then the condition $(\forall \sigma_u \in T_{uo}^* \forall M \in \mathcal{M}_B)[M[\sigma_u\rangle \Rightarrow (\exists t \in T_{uo} \cup T(E \setminus \hat{E}_1 \cap \Gamma(M)))M[\sigma_u t\rangle]$ also holds. Therefore, 1) is proved.

We now prove 2). Due to $\hat{E}_1 \subseteq \hat{E}_2$, we have $(E \setminus \hat{E}_1) \supseteq (E \setminus \hat{E}_2)$. Based on the definition of $\hat{E}_i$-induced twin-BRG, $B_{E\setminus\hat{E}_i}$ only contains the events in $E \setminus \hat{E}_i$. If, in $B_{E\setminus\hat{E}_1}$, the initial state cannot reach a critical observability violative state, it implies that the initial state can reach a critical observability violative state in $B_{E\setminus\hat{E}_2}$ neither. Thus, the event set $\hat{E}_2$ is feasible. □

Based on the notion of a stop-free event set, by solving a set of integer linear constraints, we propose a necessary and sufficient condition to check the stop-freeness of a subset of controllable events.

Proposition 8: Given an LPN $G = (PN, M_0, E, \lambda)$ with $E = E_c \cup E_{uc}$, let $\mathcal{B} = (\mathcal{M}_B, E, \delta, M_0)$ be its BRG. An event set $\hat{E}_c \subseteq E_c$ is stop-free *iff* for each basis marking $M \in \mathcal{M}_B$ the following set of integer constraints $F(M)$ is infeasible:

$$F(M) = \begin{cases} M' = M + C_u \cdot \vec{y}_\sigma \geq \vec{0}, \\ \vec{y}_\sigma \in \mathbb{N}^{n_{uo}}, \\ \varphi(M') \end{cases} \tag{4}$$

where $\varphi(M') = \bigwedge_{t \in T_{uo} \cup T(E\setminus\hat{E}_c \cap \Gamma(M))} (\bigvee_{p \in {}^\bullet t} M'(p) \leq Pre(p,t) - 1)$.

Proof: *(If)* If Eq. (4) is infeasible, there exists no firing vector $\vec{y}_\sigma \in \mathbb{N}^{n_{uo}}$ such that $M' = M + C_u \cdot \vec{y}_\sigma \geq \vec{0}$, and there is no transition $t \in T_{uo} \cup T(E \setminus \hat{E}_c \cap \Gamma(M))$ enabled at M', i.e., M' is a deadlock marking due to control-disabled or lack of tokens. Thus, by Theorem 1 in [20], by firing any unobservable transition sequence at M, M can never enter a deadlock. Based on Definition 8, the event set $\hat{E}_c \subseteq E_c$ has stop-freeness.

(Only if) We prove it by contrapositive. Suppose that Eq. (4) is feasible, there exists a firing vector $\vec{y}_\sigma \in \mathbb{N}^{n_{uo}}$ such that $M' = M + C_u \cdot \vec{y}_\sigma \geq \vec{0}$ and M' is a deadlock marking due to control-disabled or lack of tokens. By Theorem 1 in [20], there exists an unobservable transition sequence $\sigma_u \in T_{uo}^*$ with $\vec{y}_{\sigma_u} = \vec{y}_\sigma$ such that $M[\sigma_u\rangle M'$ and M' is a deadlock marking, which violates the condition in Definition 8. □

Note that the logical OR condition in $\varphi(M')$ of Eq. (4) can be linearized by the method in [26]. Thus, Eq. (4) can be represented as an integer linear constraint set.

Given a bounded LPN G, a stop-free and feasible event set $\hat{E}_c$ is not unique in general. In fact, among all the feasible stop-free event sets, we want to find a set with the maximal cardinality, since it can provide the maximum flexibility to design a control policy. To this end, we propose an algorithm to compute such a feasible stop-free event set.

Now, we explain the details of Algorithm 1. Step 1 computes the BRG of a net system. Step 2 sets a power set $Z = 2^{E_c}$ as the set of candidates. Steps 3–16 are the procedures of the search process. At each iteration, a minimal set z among all the untested sets is chosen, and we test if the corresponding maximal set $\hat{E}_c = E_c \setminus z$ is feasible and stop-free. In particular, the stop-freeness is tested by Proposition 8. As for the feasibility, it is tested by checking whether there exists no critical observability violative state in the $(E \setminus \hat{E}_c)$-induced twin-BRG $B_{E\setminus\hat{E}_c}$. If the above two conditions (feasibility and stop-freeness) hold, then we can obtain a feasible stop-free event set $\hat{E}_c$ that has the maximal number of elements. Otherwise, if at least one of the two conditions (feasibility and stop-freeness) does not hold, z is deleted from the candidate set Z, and by Steps 11 and 14, some other sets can be deleted based on the contraposition of Proposition 7.

Example 2: Let us consider again the LPN G in Fig. 1 and a critical marking set $C_R = \{M \in \mathbb{N}^{11} | -M(p_{10}) - M(p_{11}) \leq -1\}$. The controllable event set is $E_c = \{a, b, e\}$ and the uncontrollable event set is $E_{uc} = \{c, d, f\}$. Now, we use Algorithm 1 to compute a feasible stop-free event set with the maximal cardinality as $\hat{E}_c = \{a\}$. □

Algorithm 1 Computation of a feasible stop-free event set $\hat{E}_c$.

Input: An LPN $G = (PN, M_0, E, \lambda)$ with $E = E_c \cup E_{uc}$ and a critical marking set C_R

Output: A feasible stop-free event set $\hat{E}_c$

1: compute the BRG $\mathcal{B} = (\mathcal{M}_B, E, \delta, M_0)$ of G by Algorithm 1 in [21];
2: let $Z = 2^{E_c}$;
3: **while** $Z \neq \emptyset$ **do**
4: select a minimal event set $z \in Z$;
5: let $\hat{E}_c = E_c \setminus z$;
6: compute the $(E \setminus \hat{E}_c)$-induced twin-BRG $B_{E\setminus\hat{E}_c}$;
7: **if** $\hat{E}_c$ has stop-freeness **then**
8: **if** $\hat{E}_c$ has feasibility **then**
9: output $\hat{E}_c$;
10: **else**
11: let $Z = Z \setminus \{z' \in Z | z' \supseteq z\}$;
12: **end if**
13: **else**
14: let $Z = Z \setminus \{z' \in Z | z' \subseteq z\}$;
15: **end if**
16: **end while**
17: output: No solution.

Now, we analyze the complexity of Algorithm 1. Consider an LPN $G = (PN, M_0, E, \lambda)$ whose BRG and twin-BRG are $\mathcal{B} = (\mathcal{M}_B, E, \delta, M_0)$ and $B = (X, E, \delta_{tw}, x_0)$, respectively. The stop-freeness is tested by solving at most $|\mathcal{M}_B|$ ILP problems of Eq. (4). Moreover, the feasibility is tested by examining the existence of a critical observability violative state in $B_{E\setminus\hat{E}_c}$, whose complexity is $O(|\mathcal{M}_B|^2)$. Finally, the loop of Steps 3–16 executes at most $2^{|E_c|}$ times if z is equal to E_c, which means $\hat{E}_c = \emptyset$. In this case, x_0 is a state in the set $X \setminus X_{good}$, and there is no stop-free control policy to enforce criticl observability according to Proposition 6.

Note that the complexity of constructing a BRG is exponential w.r.t. the number of places and the number of tokens in the initial marking, and the complexity of solving an ILP problem is exponential w.r.t. its number of variables and constraints. Thus, the complexity of Algorithm 1 is exponential w.r.t. the number of places, the number of transitions, the number of tokens in the initial marking, and the number of controllable events.

VI. Online Control Policy for Critical Observability Enforcement

In this section, an online control policy for critical observability enforcement is presented. By applying the proposed online control policy, the closed-loop system is critically observable and deadlock-free.

Algorithm 2 Online control policy ζ.

Input: An LPN $G = (PN, M_0, E, \lambda)$ with $E = E_c \cup E_{uc}$ and a critical marking set C_R

Output: Online control policy ζ

Offline Stage:

1: call Algorithm 1 to compute an event set $\hat{E}_c$ with stop-freeness and feasibility;
2: **if** Algorithm 1 has no solution **then**
3: exit;
4: **end if**
5: compute the twin-BRG B and the set of states X_{good};
6: let $\mathcal{D} = \emptyset$;
7: **for** each $x_i \in X_{good}$, $e \in \hat{E}_c$ such that $\delta_{tw}(x_i, e) = x_j \in X \setminus X_{good}$ **do**
8: let $\mathcal{D} = \mathcal{D} \cup \{(x_i, e, x_j)\}$;
9: remove edge (x_i, e, x_j) from B;
10: **end for**

Online Stage:

11: let $w = \varepsilon$;
12: compute the set of basis markings $\mathcal{C}_b(w)$;
13: let $\zeta(\mathcal{C}_b(w)) = E_c$;
14: **for** each $(x_i, e, x_j) \in \mathcal{D}$ **do**
15: **if** $x_i \subseteq \mathcal{C}_b(w)$ **then**
16: let $\zeta(\mathcal{C}_b(w)) = \zeta(\mathcal{C}_b(w)) \setminus \{e\}$;
17: **end if**
18: **end for**
19: wait until an event e occurs, let $w = we$;
20: go to Step 12.

Algorithm 2 is composed by an offline stage and an online stage. In the offline stage, Step 1 computes an event set $\hat{E}_c$ with stop-freeness and feasibility by Algorithm 1. Step 5 computes the twin-BRG and the set of states X_{good}, and Steps 6–10 generate the set of disabled edges $\mathcal{D}$, which includes the edges of the twin-BRG such that those edges should be forbidden by the supervisor. The online control policy is designed according to the set of disabled edges $\mathcal{D}$. For each word w, the supervisor first updates the set of basis markings $\mathcal{C}_b(w)$ consistent with w. If the set $\mathcal{C}_b(w)$ contains a state x_i such that $(x_i, e, x_j) \in \mathcal{D}$, then event e is disabled by the supervisor.

Theorem 1: Given an LPN $G = (PN, M_0, E, \lambda)$ that satisfies Assumptions A1–A4, and a critical marking set C_R, the closed-loop system (G, ζ) is critically observable, where ζ is the control policy obtained by Algorithm 2.

Proof: By contradiction, suppose that the closed-loop system (G, ζ) is not critically observable. There is a state $x \in X_N$ in the twin-BRG B and a path $x_0 \xrightarrow{w} x_j \xrightarrow{e} x_k \xrightarrow{w'} x$ in B with $e \in \zeta(\mathcal{C}_b(w)) \cap \hat{E}_c$ and $w' \in (E \setminus \hat{E}_c)^*$. However, $e \in \hat{E}_c$ is disabled by Steps 14–18, which contradicts $e \in \zeta(\mathcal{C}_b(w))$. □

Theorem 2: Given an LPN $G = (PN, M_0, E, \lambda)$ that satisfies Assumptions A1–A4, and a critical marking set C_R, the critically observable closed-loop system (G, ζ) derived from Algorithm 2 is deadlock-free.

Proof: Since Algorithm 2 never disables any event $e \notin \hat{E}_c$, based on Proposition 4, the critically observable closed-loop system (G, ζ) is deadlock-free. □

Then, we prove that the critically observable closed-loop system obtained by Algorithm 2 is maximally permissive under a particular condition.

Theorem 3: Given an LPN $G = (PN, M_0, E, \lambda)$ that satisfies Assumptions A1–A4, and a critical marking set C_R. If $E_c = \hat{E}_c$, the critically observable closed-loop system (G, ζ) derived from Algorithm 2 is maximally permissive, i.e., for any other critically observable closed-loop system (G, ζ'), it holds $\mathcal{L}(G, \zeta) \not\subset \mathcal{L}(G, \zeta')$.

Proof: By contradiction, suppose that a critically observable closed-loop system (G, ζ) is not maximally permissive, i.e., there exists another critically observable closed-loop system (G, ζ') with a language $\mathcal{L}(G, \zeta')$ such that $\mathcal{L}(G, \zeta) \subset \mathcal{L}(G, \zeta')$. This implies the following two facts:

1) $\forall w \in \mathcal{L}(G, \zeta)$, $\zeta(\mathcal{C}_b(w)) \subseteq \zeta'(\mathcal{C}_b(w))$;
2) $\exists w_1 \in \mathcal{L}(G, \zeta)$, $\zeta(\mathcal{C}_b(w_1)) \subset \zeta'(\mathcal{C}_b(w_1))$.

Let us take into account the word $w_1 \in \mathcal{L}(G, \zeta)$ such that $\zeta(\mathcal{C}_b(w_1)) \subset \zeta'(\mathcal{C}_b(w_1))$ and $\zeta(\mathcal{C}_b(w_1')) = \zeta'(\mathcal{C}_b(w_1'))$, $\forall w_1' \in \bar{w}_1 \setminus \{w_1\}$. Then, for the control action at $\mathcal{C}_b(w_1)$ in ζ, there exists an event e such that $e \in \zeta'(\mathcal{C}_b(w_1))$ but $e \notin \zeta(\mathcal{C}_b(w_1))$. Due to $e \in \zeta'(\mathcal{C}_b(w_1))$, e is not disabled by Algorithm 2. Since $E_c = \hat{E}_c$, we have $E \setminus \hat{E}_c = E \setminus E_c = E_{uc}$. Then, a state $x \in X \setminus X_{good}$ can be reachable from x_0, which uncontrollably yields a critical observability violative state through $w_{uc} \in E_{uc}^*$. According to Proposition 3, (G, ζ') is not critically observable, which leads to a contradiction. □

Note that the output of Algorithm 2 is unique under Assumption A4, which is consistent with the results shown in [13, 19].

Example 3: Let us consider again the LPN G in Fig. 1. Its BRG and twin-BRG are shown in Figs. 2 and 3, respectively. Given a critical marking set $C_R = \{M \in \mathbb{N}^{11} | -M(p_{10}) - M(p_{11}) \leq -1\}$, by using Algorithm 1, a feasible stop-free event set $\hat{E}_c = \{a\}$ is obtained. The set of states of X_{good} is $X_{good} = \{x_0, x_1, x_3, x_4, x_5, x_7, x_8\}$. The set of disabled edges is $\mathcal{D} = \{(x_0, a, x_2)\}$. Then, we can obtain an online control policy by Algorithm 2.

If no event occurs, the set of basis markings consistent with $w = \varepsilon$ is $\mathcal{C}_b(\varepsilon) = \{M_0\}$. Since $(x_0, a, x_2) \in \mathcal{D}_2$ and $x_0 \subseteq \mathcal{C}_b(\varepsilon)$, event a is disabled at $\mathcal{C}_b(\varepsilon)$, i.e., $\zeta(\mathcal{C}_b(\varepsilon)) = E_c \setminus \{a\} = \{b, e\}$. If event b occurs, the set of basis markings consistent with $w = b$ is $\mathcal{C}_b(b) = \{M_3\}$. By $x_0 \notin \mathcal{C}_b(b)$, no event

is disabled, i.e., $\zeta(\mathcal{C}_b(b)) = \{a, b, e\}$. If events c, d, and e occur successively, we can obtain $\zeta(\mathcal{C}_b(bc)) = \zeta(\mathcal{C}_b(bcd)) = \zeta(\mathcal{C}_b(bcde)) = \{a, b, e\}$ for the same reason. Fig. 5 shows the critical observability enforcing supervisor obtained by the online control policy, which also describes part of the evolution of the closed-loop system (G, ζ). □

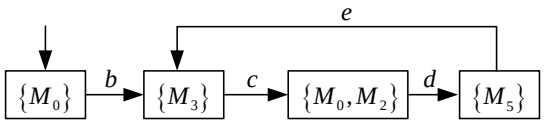

Fig. 5: The critical observability enforcing supervisor obtained by Algorithm 2 for Example 3.

Finally, we analyze the complexity of Algorithm 2. In the offline stage, the complexity of Steps 1–4 in Algorithm 2 is equal to that of Algorithm 1. Since the twin-BRG has at most $|\mathcal{M}_B|^2$ states and $|\mathcal{M}_B|^4 \cdot |E|$ transitions, the complexity of Steps 5–10 is $O(|\mathcal{M}_B|^4 \cdot |E|)$. Moreover, for each observed event, the online stage of Algorithm 2 (Steps 11–20) is a one-step-look-ahead procedure by analyzing the set $\mathcal{D}$, whose complexity is negligible. In summary, the complexity of the online control policy in Algorithm 2 is polynomial w.r.t. the size of the BRG but exponential w.r.t. the number of controllable events.

VII. Conclusion

In this paper, we propose two methods to verify and enforce critical observability of an LPN by using basis markings. First, we provide a necessary and sufficient condition to verify critical observability of an LPN based on the twin-BRG and the solutions of some ILP problems. Then, we formulate the problem of critical observability enforcement and introduce the notion of a stop-free event set in the BRG. Furthermore, we propose a method to compute a feasible stop-free event set in the twin-BRG. Finally, based on the aforementioned results, we present an online control policy by disabling the edges in the twin-BRG such that the closed-loop system is critically observable and deadlock-free. Future work will extend the proposed methods to decentralized and distributed settings.

References

[1] E. De Santis, M. D. Di Benedetto, S. Di Gennaro, A. D'Innocenzo, and G. Pola, "Critical observability of a class of hybrid systems and application to air traffic management," *Lecture Notes in Control and Inform. Sci.*, Springer-Verlag, vol. 337, pp. 141–170, 2006.

[2] E. De Santis and M. D. Di Benedetto, "Observability and diagnosability of finite state systems: A unifying framework," *Automatica*, vol. 81, pp. 115–122, 2017.

[3] G. Pola, E. De Santis, M. D. Di Benedetto, and D. Pezzuti, "Design of decentralized critical observers for networks of finite state machines: A formal method approach," *Automatica*, vol. 86, pp. 174–182, 2017.

[4] A. W. Lai, S. Lahaye, and J. Komenda, "Observer construction for polynomially ambiguous max-plus automata," *IEEE Trans. Autom. Control*, vol. 67, no. 3, 1582–1588.

[5] T. Masopust, "Critical observability for automata and Petri nets," *IEEE Trans. Autom. Control*, vol. 65, no. 1, pp. 341–346, Jan. 2020.

[6] X. Y. Cong, M. P. Fanti, A. M. Mangini, and Z. W. Li, "Critical observability of discrete-event systems in a Petri net framework," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 5, pp. 2789–2799, 2022.

[7] Y. Y. Yan, H. Deng, and Z. Q. Chen, "A new look at the critical observability of finite state machines from an algebraic viewpoint," *Asian J. Control*, vol. 24, pp. 3056–3065, 2022.

[8] W. Wang, S. Lafortune, A. R. Girard, and F. Lin, "Optimal sensor activation for diagnosing discrete event systems," *Automatica*, vol. 46, no. 7, pp. 1165–1175, 2010.

[9] M. P. Cabasino, S. Lafortune, and C. Seatzu, "Optimal sensor selection for ensuring diagnosability in labeled Petri nets," *Automatica*, vol. 49, no. 8, pp. 2373–2383, 2013.

[10] N. Ran, J. Y. Hao, and C. Seatzu, "Prognosability analysis and enforcement of bounded labeled Petri nets," *IEEE Trans. Autom. Control*, vol. 67, no. 10, pp. 5541–5547, 2022.

[11] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Control Optim.*, vol. 25, no. 1, pp. 206–230, 1987.

[12] 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.

[13] X. Yin and S. Lafortune, "A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems," *IEEE Trans. Autom. Control*, vol. 61, no. 8, pp. 2140–2154, Aug. 2016.

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

[15] S. Jiang, Z. Huang, V. Chandra, and R. Kumar, "A polynomial algorithm for testing diagnosability of discrete-event systems," *IEEE Trans. Autom. Control*, vol. 46, no. 8, pp. 1318–1321, Aug. 2001.

[16] K. Hernández-Rueda, M. E. Meda-Campaña, and J. Arámburo-Lizrraga, "Enforcing diagnosability in interpreted Petri nets," *IFAC-PapersOnline*, vol. 48, no. 7, pp. 56–63, 2015.

[17] Y. H. Hu, Z. Y. Ma, Z. W. Li, and A. Giua, "Diagnosability enforcement in labeled Petri nets using supervisory control," *Automatica*, vol. 131, 109776, 2021.

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

[19] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, Cham, Switzerland: Springer, 3rd, 2021.

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

[21] Y. Tong, Z. W. 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.

[22] A. Giua, F. DiCesare, and M. Silva, "Generalized mutual exclusion constraints for Petri nets with uncontrollable transitions," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Chicago, IL, USA, 1992, pp. 974–979.

[23] X. Y. Cong, M. P. Fanti, A. M. Mangini, and Z. W. Li, "Supplementary document to "Critical observability verification and enforcement of labeled Petri nets by using basis marking"," 2023. Available:

[24] Z. Y. Ma, X. Yin, and Z. W. Li, "Marking predictability and prediction in labeled Petri nets," *IEEE Trans. Autom. Control*, vol. 66, no. 8, pp. 3608–3623, Aug. 2020.

[25] A. Giua, "Supervisory control of Petri nets with language specifications," in C. Seatzu, M. Silva, and J. van Schuppen (Eds.), Control of discrete-event systems, vol. 433, (pp. 235–255). London, U.K.: Springer, 2013.

[26] M. P. Cabasino, A. Giua, and C. Seatzu, "Identification of Petri nets from kowledge of their language," *Discrete Event Dyn. Syst.*, vol. 17, no. 4, pp. 447–474, 2007.