Synthesis for observability of logical control networks* 

Kuize Zhang

*Department of Computer Science, University of Surrey, Guildford GU2 7XH, UK

Abstract

Finite-state systems have applications in systems biology, formal verification and synthesis of infinite-state (hybrid) systems, etc. As deterministic finite-state systems, logical control networks (LCNs) consist of a finite number of nodes which can be in a finite number of states and update their states. In this paper, we investigate the synthesis problem for observability of LCNs based on state feedback with exogenous input by using the semitensor product proposed by Daizhan Cheng and the notion of observability graph (previously called weighted pair graph) proposed by us. We show that state feedback with exogenous input can either enforce or weaken observability of an LCN. We prove that for an LCN \( \Sigma \) and another closed-loop LCN \( \Sigma_C \) obtained by feeding a state-feedback controller \( C \) with exogenous input into \( \Sigma \), (1) if \( \Sigma \) is observable, then \( \Sigma_C \) can be either observable or not; (2) if \( \Sigma \) is not observable, \( \Sigma_C \) can also be observable or not. We also prove that if an unobservable LCN can be made observable by state feedback with exogenous input, then it can also be made observable by state feedback (without exogenous input, equivalent to state feedback with constant input). Furthermore, we give an upper bound on the number of state-feedback controllers that are needed to be tested in order to verify whether an unobservable LCN can be made observable by state feedback, and based on the procedure of obtaining the upper bound, we design an observability synthesis algorithm, by additionally combining the ideas of a greedy algorithm and dynamic programming. These results open the study of observability synthesis in LCNs.

Key words: logical control network, observability, state feedback, synthesis, semitensor product, observability graph

1 Introduction

1.1 Background

Finite-state systems have applications in many areas such as formal verification and synthesis of infinite-state (hybrid) systems [1,2,3], systems biology [4], etc.

As special deterministic finite-state systems in which all nodes can be only in one of two states, Boolean control networks (BCNs) were proposed to describe genetic regulatory networks [5,6]. In a BCN, nodes can be in one of two discrete states “1” and “0”, which represent a gene state “on” (high concentration of a protein) and “off” (low concentration), respectively. Every node updates its state according to a Boolean function of the states of several of the network nodes. Although BCNs are a simplified model of genetic regulatory networks, they can be used to characterize many important phenomena of biological systems, e.g., cell cycles [7], cell apoptosis [8]. Hence the study on BCNs has been paid wide attention [9,10,11,12].

A logical control network (LCN) is also a deterministic finite-state system but naturally extends a BCN in the sense that the nodes of the former can be in one of a finite number (but not necessarily 2) of states [13]. From a practical point of view, LCNs can be used to describe more biological systems than BCNs. However, under the semitensor product (STP) framework, they have the same algebraic form [13], hence they can be dealt with by using the same method. In this paper, we study LCNs.

In 2007, Akutsu et al. [14] proved that it is NP-hard to verify whether a BCN is controllable in the number of nodes, hence there exists no polynomial-time algorithm for determining controllability of BCNs unless \( P = NP \). They also pointed out that “One of the major goals of systems biology is to develop a control theory for complex biological systems”. Later in 2013, Laschov et al. [15] proved that it is also NP-hard to verify whether a BCN is observable in the number of nodes. These NP-hardness results show that it is generally intractable to verify controllability and observability for large-scale
BCNs (those with more than approximately 30 nodes, i.e., more than $2^{30}$ states), and also stimulate explorations on how to design efficient verification algorithms that work on large-scale BCNs with special structures, e.g., in [16,17].

Recently, a control-theoretic framework for BCNs based on the STP of matrices (proposed by Cheng [18] in 2001) was established by Cheng and Qi [19] in 2009. Although the STP method cannot make the generally intractable problems related to BCNs become tractable, it provides a matrix method for characterizing BCNs, so that many matrix-based techniques in control theory can be used to study BCNs. As a result, it stimulates the studies on control problems of BCNs based on diverse methods, e.g., controllability [19,20], observability [19,21,22,23], reconstructibility [21,24], identifiability [25,26], invertibility [27], Kalman decomposition [28], disturbance decoupling [29], and other related results [30,31,32,33,34]. Among these results, some was worked out based on a computational-algebra method [29], some based on finite automata and graph theory [22,24], some based on the STP and graph theory [21], and some based on symbolic dynamics [27,35].

1.2 Literature review

Among many control properties, controllability and observability are the most fundamental ones. The former implies that an arbitrary given state of a system can be steered to an arbitrary given state by some input sequence and the corresponding output sequence. The latter implies that an arbitrary given state of a system can be steered to an arbitrary given state by some input sequence and the corresponding output sequence. The importance of controllability and observability of BCNs can be found in [14] and [8], etc. Lack of these properties makes a system lose many good behaviors. So, it is important to investigate how to enforce controllability and observability, e.g., by means of feedback and controller synthesis. Matrix forms of necessary and sufficient conditions for controllability of BCNs were given in [19,20]. Necessary and sufficient conditions for observability of BCNs are much more difficult to obtain, and furthermore, there exist nonequivalent definitions of observability in BCNs, e.g., it was proven in [22] that four definitions of observability are pairwise nonequivalent, showing that observability is not dual to controllability, remarkably differentiating BCNs from linear control systems. A summary on necessary and sufficient conditions of different definitions of observability of BCNs is given in Table 1. In [36,22], a notion of weighted pair graph was proposed (later it was renamed observability graph in [17,37]), and verifiable necessary and sufficient conditions for four definitions of observability (shown in Definitions 2.2, 2.3, 2.4, 2.5) were obtained by computing different types of deterministic finite automata from an observability graph adapted to the four definitions of observability. Later on, the observability graph was used in many papers to solve related problems, e.g., [38] (the observability graph was called observability matrix therein, the set of diagonal vertices and the set of non-diagonal vertices in an observability graph (see Definition 2.6) are exactly the set $D$ and the set $\Xi$ in [38, Page 78]), [39,17] (called observability graph therein), [40,41] 1, [42,43] (called parallel extension therein), [44]. In addition, the non-diagonal subgraph of an observability graph (see Definition 2.6) was proposed in [24] (called weighted pair graph therein, later renamed reconstructibility graph in [17] and detectability graph in [37]) to verify two definitions of reconstructibility of BCNs (the stronger one was earlier studied in [21]). Recently, a variant of the reconstructibility graph was used to study reconstructibility of singular Boolean control networks [45], where such networks are a subclass of nondeterministic finite-transition systems [46] by definition.

Observability results have also been extended to nondeterministic finite-transition systems (NFTS’s) [46] and probabilistic Boolean networks (PBNs) [47,43,48,44], where the stochastic switching signals in PBNs are independent and identically distributed processes, hence the PBNs are actually discrete-time finite-state time-homogeneous Markov chains. Moreover, if all probabilities in such a PBN are removed then it becomes an NFTS in which the input is constant. That is, the systems considered in [46] are more general than the systems considered in [47,43,48,44]. In [46], the notion of observability graph was extended from BCNs to NFTS’s, where the matrix $O$ in Eqn. (5) of [46] is the adjacency matrix of the (extended) observability graph of an NFTS. By additionally computing different types of deterministic finite automata from an observability graph, verifiable necessary and sufficient conditions were given for Definitions 2.2, 2.3, 2.5 extended to NFTS’s. In [47,43,48,44], three definitions of observability for PBNs were studied: observability in probability on $[0, \theta]$ with $\theta \in \mathbb{N}$ (see notation in Section 2.1), finite-time observability in probability, and asymptotic observability in distribution. The main results in [43,44] were obtained by using the observability graph (called parallel extension in [43]). Later, by definition we will show that the first two of the above three definitions of observability in PBNs are actually a slightly stronger version of Definition 2.3 and Definition 2.5 itself, in BCNs, respectively. We will also point out that the results in [44] already show that the third one is also a slightly stronger version of Definition 2.3 of BCNs. That is, the probabilities in the PBNs studied in [47,43,48,44] play no role in adding stochasticity when studying observability. See Remark A.1, in which the deterministic essence of observability of PBNs studied in [47,43,48,44] is revealed.

1 [41, Theorem 1] is a special case of [22, Theorem 3.15] (i.e., Lemma 2.7).
Let $\Sigma$ be an LCN and $\Sigma_C$ an LCN obtained by feeding a
state-feedback controller $C$ with exogenous input into $\Sigma$.

(1) We prove that state feedback with exogenous input sometimes can enforce observability of LCNs. If $\Sigma$ is observable, then $\Sigma_C$ can be either observable or not; if $\Sigma$ is unobservable, then $\Sigma_C$ can also be either observable or not.

(2) We prove that if an unobservable $\Sigma$ can be made observable by state feedback with exogenous input, then it can also be made observable by state feedback (without exogenous input, equivalent to state feedback with constant exogenous input). This result yields an algorithm for verifying whether an unobservable LCN can be made observable by state feedback with exogenous input, since there are finitely many state-feedback controllers (although there are infinitely many state-feedback controllers with exogenous input).

(3) We also obtain an upper bound on the number of state-feedback controllers that are needed to be substituted into an unobservable LCN $\Sigma$ to check whether $\Sigma$ can be made observable by state feedback, and based on the procedure of obtaining the upper bound, we design an observability synthesis algorithm, by additionally combining the ideas of a greedy algorithm and dynamic programming, opening the study of observability synthesis in LCNs.

The above (1) had been presented at the 58th IEEE Conference on Decision and Control 2019 [54] and are also illustrated in Table 2. The other contributions show substantially new results compared with (1).

The remainder of this paper is organized as follows. Section 2 introduces preliminaries, i.e., LCNs with their algebraic form under the STP framework, basic verification methods for observability of LCNs. The main results are shown in Section 3. Section 4 is a short conclusion.

2 Preliminary results

2.1 The semitensor product of matrices

We introduce necessary notation as follows.

- $\subset$ and $\subseteq$ denote subset and strict subset relations, respectively.
- $2^X$: power set of set $X$
- $\mathbb{Z}_+$: set of positive integers
- $\mathbb{N}$: set of natural numbers (including 0)
- $\mathbb{R}^n$: set of $n$-length real column vectors
- $\mathbb{R}^{m \times n}$: set of $m \times n$ real matrices
- $D_k$: set $\{0, \frac{1}{k}, ..., 1\}$ of $k$-value logic
- $\delta_n[i]$: $i$-th column of the identity matrix $I_n$
- $\Delta_k$: $k$-th column of the identity matrix $I_n$

- $\Delta_n$; set $\{\delta_1^n, ..., \delta_n^n\}$ ($\Delta := \Delta_2$)
- $[m, n]$: $\{m, m+1, ..., n\}$, where $m, n \in \mathbb{N}$ and $m \leq n$
- $\delta_n[i_1, ..., i_k]$: logical matrix $[\delta_1^n, ..., \delta_n^n]$, where $i_1, ..., i_k \in \{1, n\}$
- $\mathcal{L}_{n \times s}$: set of $n \times s$ logical matrices
- $\text{Col}_i(A)$: $i$-th column of matrix $A$
- $\text{Col}(A)$: set of columns of matrix $A$
- $A^T$: transpose of matrix $A$
- $|X|$: cardinality of set $X$

Definition 2.1 ([55]) Let $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{p \times q}$, and $\alpha = \text{lcm}(n, p)$, and the STP of $A$ and $B$ is defined as

$$A \kappa B = (A \otimes I_\alpha) \left( B \otimes I_\alpha \right),$$

where $\otimes$ denotes the Kronecker product.

From this definition, it is easy to see that the conventional product of matrices is a particular case of the STP, since if $n = p$ then $A \kappa B = AB$. Since the STP keeps most properties of the conventional product, e.g., the associative law [37], the distributive law, etc. [55], we usually omit the symbol “$\kappa$” hereinafter.

2.2 Logical control networks and their algebraic form

In this paper, we investigate the following LCN with $n$ state nodes, $m$ input nodes, and $q$ output nodes:

$$x_1(t+1) = f_1(x_1(t), ..., x_n(t), u_1(t), ..., u_m(t)),\quad y_1(t) = h_1(x_1(t), ..., x_n(t)),$$

$$x_2(t+1) = f_2(x_1(t), ..., x_n(t), u_1(t), ..., u_m(t)),\quad y_2(t) = h_2(x_1(t), ..., x_n(t)),$$

$$\vdots$$

$$x_q(t+1) = f_q(x_1(t), ..., x_n(t), u_1(t), ..., u_m(t)),\quad y_q(t) = h_q(x_1(t), ..., x_n(t)),$$

where $t \in \mathbb{N}$ denote discrete time steps; $x_i(t) \in D_{n_i}$, $u_j(t) \in D_{m_j}$, and $y_k(t) \in D_{q_k}$ denote values of state node $x_i$, input node $u_j$, and output node $y_k$ at time step $t$, respectively, $i \in \{1, n\}$, $j \in \{1, m\}$, $k \in \{1, q\}$; $\prod_{j=1}^{m} n_j =: N_i$; $\prod_{j=1}^{m} m_j =: M$; $\prod_{k=1}^{q} q_k =: Q$; $f_i : D_{MN} \rightarrow D_{n_i}$

$^2$ In [55], the associative law $(A \kappa B) \kappa C = A \kappa (B \kappa C)$ was proven in the special case that $n$ divides $p$ (or vice versa) and $q$ divides $r$ (or vice versa), where $n$ and $q$ are the numbers of rows of $A$ and $B$, respectively, $p$ and $r$ are the numbers of rows of $B$ and $C$, respectively.
and \( h_k : D_N \rightarrow D_{q_k} \) are logical mappings, \( i \in [1, n] \), \( k \in [1, q] \).

When \( n_1 = \cdots = n_q = m_1 = \cdots = m_m = q_1 = \cdots = q_q = 2 \), Eqn. (1) is a BCN.

Eqn. (1) can be represented in the compact form

\[
\begin{align*}
    x(t + 1) &= f(x(t), u(t)), \\
    y(t) &= h(x(t)),
\end{align*}
\]

where \( t \in \mathbb{N} \); \( x(t) \in D_N \), \( u(t) \in D_M \), and \( y(t) \in D_Q \) stand for the state, input, and output of the LCN at time \( t \); \( f : D_{NM} \rightarrow D_N \) and \( h : D_N \rightarrow D_Q \) are mappings.

For each \( n \in \mathbb{Z}_+ \) greater than 1, we map \( \delta^i_n \) in \( D_n \) to \( \delta^i_n \) in \( \Delta_n \) for all \( i \in [1, n] \), and write \( \delta^i_n \rightarrow \delta^i_n \). Then under the STP framework, Eqn. (2) can be transformed to its equivalent algebraic form as follows [55]:

\[
\begin{align*}
    \ddot{x}(t + 1) &= L \dot{x}(t)\ddot{u}(t) = [L_1, \ldots, L_N]\dot{x}(t)\ddot{u}(t), \\
    \dot{y}(t) &= H \dot{x}(t),
\end{align*}
\]

where \( t \in \mathbb{N} \); \( x(t) \in \Delta_N \), \( u(t) \in \Delta_M \), \( \dot{y}(t) \in \Delta_Q \); \( L \in \mathcal{L}_{NM} \) and \( H \in \mathcal{L}_{QN} \) are called the structure matrices, \( L_i \in \mathcal{L}_{NM} \), \( i \in [1, N] \).

2.3 Preliminary results for observability

In [36,22], four types of observability were verified for BCNs by proposing a unified automaton method (computing four types of deterministic finite automata from the observability graph (proposed in [36,22] and called weighted pair graph therein, and renamed observability graph in [17,37]) of a BCN to verify the corresponding four types of observability). In this paper, we are particularly interested in the linear type (as in Definition 2.2, i.e., the strongest one among the four types, earlier studied in [21]), as if an LCN satisfies this observability property, it is very easy to recover the initial state by using an input sequence and the corresponding output sequence. Note that all results in [36,22] can be trivially extended to LCNs. The necessary and sufficient condition for Definition 2.2 given in [21] is as follows: a BCN satisfies Definition 2.2 if and only if for every pair of different periodic (state, input)-trajectories of the same minimal period \( k \) and the same input trajectory, the corresponding output trajectories are also different and periodic of minimal period \( k \). Obviously, the necessary and sufficient condition given in [21] is much more complex than the one given in the subsequent Lemma 2.7 [22].

The four types of observability studied in [36,22] are as follows (also see Table 1). We adopt the terminology used in [37].

**Definition 2.2** An LCN (2) is called arbitrary-experiment observable if for all different initial states \( x(0), x'(0) \in D_N \), for each input sequence \( u(0)u(1) \ldots \), the corresponding output sequences \( y(0)y(1) \ldots \) and \( y'(0)y'(1) \ldots \) are different.

**Definition 2.3** An LCN (2) is called multiple-experiment observable if for every two different initial states \( x(0), x'(0) \in D_N \), there is an input sequence such that the output sequences corresponding to \( x(0) \) and \( x'(0) \) are different. Such an input sequence is called a distinguishing input sequence of \( x(0) \) and \( x'(0) \).

**Definition 2.4** An LCN (2) is called strongly multiple-experiment observable if for every initial state \( x(0) \in D_N \), there exists an input sequence such that for each initial state \( x(0) \in D_N \) different from \( x(0) \), the output sequences corresponding to \( x(0) \) and \( x(0) \) are different.

**Definition 2.5** An LCN (2) is called single-experiment observable if there exists an input sequence such that for every two different initial states \( x(0), x'(0) \in D_N \), the output sequences corresponding to \( x(0) \) and \( x'(0) \) are different.

From now on when we mention “observability”, we always mean Definition 2.2 unless otherwise stated.

| state feedback with exogenous input | can enforce controllability? | can weaken controllability? | can enforce observability? | can weaken observability? |
|------------------------------------|-----------------------------|-----------------------------|---------------------------|---------------------------|
|                                    | No                          | Yes                         | Yes                       | Yes                       |

Table 2
Influence of state feedback with exogenous input to controllability and observability of LCNs.

\[\frac{\Delta}{\text{Exam. 3.2}}\]
\[ x'_{2} \text{ and } f(x_{1},u) = x_{2} \] of inputs. A vertex \( \{x,x'\} \) is called diagonal if \( x = x' \), and called non-diagonal otherwise. For a vertex \( v \in V \), its outdegree is \( \text{outdeg}(v) := |\bigcup_{(v,v') \in E} W((v,v'))| \), i.e., the number of inputs appearing in the edges starting from \( v \). The diagonal subgraph of an observability graph is defined by all diagonal vertices and all edges between them. Similarly, the non-diagonal subgraph is defined by all non-diagonal vertices and all edges between them.

**Lemma 2.7** ([22]) An LCN (2) is not observable if and only if in its observability graph \( G_o \), there is a non-diagonal vertex \( v \), a cycle \( C \), and a path from \( v \) to a vertex of \( C \). Particularly if in \( G_o \) there exists a path from \( v \) to a diagonal vertex, then (2) is not observable.

Since in a diagonal subgraph, there must exist a cycle and each vertex will go to a cycle, we will denote the subgraph briefly by a symbol \( \diamond \) when drawing an observability graph. Hence if there exists an edge from a non-diagonal vertex to a diagonal vertex, then the LCN is not observable.

**Example 2.8** Consider the BCN

\[ \dot{x}(t+1) = L\dot{x}(t)\tilde{u}(t), \] (4)

where \( L = [\delta_4 \ 2, \ 1, \ 3, \ 4, \ 4, \ 2, \ 2], t \in \mathbb{N}, \dot{x}(t) \in \Delta, \tilde{u}(t) \in \Delta. \) Consider the output function

\[ \tilde{y}(t) = \delta_2[1,1,2]\dot{x}(t), \] (5)

where \( t \in \mathbb{N}, \dot{x}(t) \in \Delta, \tilde{y}(t) \in \Delta. \) The observability graph of BCN (4) with output function (5) is shown in Fig. 1. This graph shows that the BCN is not observable by Lemma 2.7 since there is a self-loop on non-diagonal vertex \( \{\delta_1, \delta_3\} \).

![Fig. 1. Observability graph of BCN (4) with output function (5), where number ij in a circle denotes state pair \( \{\delta_i, \delta_j\} \), weight i denotes input \( \delta_i \).](image)

3 Synthesis results for observability

In this section, we show synthesis results for observability of LCN (2) (or its algebraic form (3)) based on state feedback with exogenous input.

3.1 Closed-loop logical control networks by state feedback with exogenous input

Consider an LCN (2). Let a state-feedback controller with exogenous input be

\[ u(t) = g(x(t), v(t)), \] (6)

where \( v(t) \in \mathcal{D}_P \) is the exogenous input, \( P = \prod_{i=1}^{P} P_i \) with each \( P_i \in \mathbb{N} \) greater than 1 (corresponding to \( l \) new input nodes); or \( P = 1 \), which means that there is only one constant input: \( g : \mathcal{D}_{NP} \rightarrow \mathcal{D}_M \) is a mapping. Equivalently in the algebraic form, for an LCN (3), we set a state-feedback controller with exogenous input to be

\[ \tilde{u}(t) = G\dot{x}(t)\tilde{v}(t) = [G_1, \ldots, G_N]x(t)\tilde{v}(t), \] (7)

where \( \tilde{v}(t) \in \Delta_P, G \in \mathbb{L}_{MN \times N} \) is called the structure matrix, \( G_i \in \mathbb{L}_{MN}, i \in [1, N]. \)

In particular, when \( P = 1 \), a state-feedback controller with exogenous input is called a state-feedback controller. When \( P = M \) and \( g(x(t), v(t)) \equiv v(t) \), controller (7) will not change the algebraic form of the original LCN, and hence will not change controllability or observability of the LCN.

Suppose we are free to modify LCN (2) by setting controller (6). Substituting (6) into (2), we obtain a closed-loop LCN (see Fig. 2 for a sketch) as

\[ x(t+1) = f(x(t), g(x(t), v(t))), \]
\[ y(t) = h(x(t)). \] (8)

![Fig. 2. Closed-loop logical control network based on state feedback with exogenous input.](image)

Equivalently, substituting (7) into (3), we obtain the algebraic form of the closed-loop LCN (8) as

\[ \dot{x}(t + 1) = L\dot{x}(t)G\dot{x}(t)\tilde{v}(t), \]
\[ \tilde{y}(t) = H\dot{x}(t). \] (9)

**Proposition 3.1** Eqn (9) is equivalent to

\[ \dot{x}(t + 1) = [L_1G_1, \ldots, L_NG_N]\dot{x}(t)\tilde{v}(t), \]
\[ \tilde{y}(t) = H\dot{x}(t). \] (10)

**Proof** By Lemmas A.2 and A.3, (9) can be rewritten as

\[ \dot{x}(t + 1) = L\dot{x}(t)G\dot{x}(t)\tilde{v}(t) \]
\[
= L(I_N \otimes G)M_N \dot{x}(t)u(t)
\]
\[
= L \begin{bmatrix} G & \cdots & \cdots & \cdots & G \\
\delta_N & \cdots & \cdots & \cdots & \delta_N \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\end{bmatrix} \otimes I_P \dot{x}(t) \dot{v}(t)
\]
\[
= L \begin{bmatrix} G & \cdots & \cdots & \cdots & G \\
\delta_N & \cdots & \cdots & \cdots & \delta_N \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\end{bmatrix} \otimes I_P \dot{x}(t) \dot{v}(t)
\]
\[
= L \begin{bmatrix} G(\delta_N^1 \otimes I_P) & \cdots & \cdots & \cdots & G(\delta_N^N \otimes I_P) \\
\delta_N & \cdots & \cdots & \cdots & \delta_N \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\end{bmatrix} \dot{x}(t) \dot{v}(t)
\]
\[
= [L_1, \ldots, L_N] \begin{bmatrix} G_1 & \cdots & \cdots & \cdots & G_N \\
\delta_N & \cdots & \cdots & \cdots & \delta_N \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\end{bmatrix} \dot{x}(t) \dot{v}(t)
\]
\[
= [L_1G_1, \ldots, L_NG_N] \dot{x}(t) \dot{v}(t).
\]

Consider the newly obtained LCN (10), if \( P = 1 \), then the corresponding structure matrix \([L_1G_1, \ldots, L_NG_N]\) is square. However, generally the structure matrix is not necessarily square, hence the updating of states generally depends on the exogenous input \( \dot{v}(t) \).

### 3.2 How state feedback influences observability of LCNs

Unlike controllability, we next give an example to show that state feedback with exogenous input can enforce observability of an LCN.

**Example 3.2** We have proved that BCN (4) with the output function (5) is not observable in Example 2.8.

Substituting the controller
\[
\dot{u}(t) = G \dot{x}(t) \dot{v}(t),
\]
where \( G = \delta_3[1, 2, 2, 1, 2, 1] \), \( \dot{v}(t) \in \Delta \), \( \dot{x}(t) \in \Delta_4 \), into (4) to obtain the closed-loop BCN
\[
\dot{x}(t + 1) = \bar{L} \dot{x}(t) \dot{u}(t),
\]
where \( \bar{L} = \delta_4[2, 2, 3, 2, 1, 2, 2] \). The observability graph of BCN (12) with output function (5) is shown in Fig. 3. This graph shows that the BCN is observable by Lemma 2.7.

Next we show that a state-feedback controller can make an observable LCN unobservable.

**Example 3.3** Consider the LCN
\[
\dot{x}(t + 1) = \delta_3[1, 3, 3, 2, 1, 1]\dot{x}(t) \dot{u}(t), \quad \dot{y}(t) = \delta_2[1, 1, 2]\dot{x}(t),
\]
where \( t \in \mathbb{N}, \dot{x}(t) \in \Delta_3, \dot{u}(t), \dot{y}(t) \in \Delta. \)

The observability graph of (13) consists of vertex \( \{\delta_1^1, \delta_2^2\} \) and the diagonal subgraph \( \circ \), and there is no path from \( \{\delta_3^1, \delta_3^2\} \) to \( \circ. \) Then by Lemma 2.7, the BCN is observable.

Substituting state-feedback controller
\[
\dot{u}(t) = \delta_2[1, 2, 1]\dot{x}(t)
\]
into (13), by Proposition 3.1, we obtain LCN
\[
\dot{x}(t + 1) = \delta_3[1, 2, 1]\dot{x}(t), \quad \dot{y}(t) = \delta_2[1, 1, 2]\dot{x}(t).
\]

There is a self-loop on vertex \( \{\delta_3^1, \delta_3^2\} \) in the observability graph of (14), then by Lemma 2.7, (14) is not observable.

Next we show that there exists an unobservable LCN such that no state-feedback controller with exogenous input can make it observable. This example also shows that sometimes state feedback with exogenous input never affects observability of LCNs.

**Example 3.4** Consider the BCN
\[
\dot{x}(t + 1) = L \dot{x}(t) \dot{u}(t),
\]
where \( L = \delta_4[1, 1, 2, 1, 1, 2, 3] \), \( t \in \mathbb{N}, \dot{x}(t) \in \Delta_4, \dot{u}(t) \in \Delta. \)

By Lemma 2.7, the BCN with output function (5) is not observable, since there exists a path \( \{\delta_1^1, \delta_2^2\} \xrightarrow{\delta_3^1} \{\delta_4^1, \delta_4^2\} \) in its observability graph.

Substituting an arbitrary state-feedback controller \( \dot{u}(t) = G \dot{x}(t) \dot{v}(t) \) with exogenous input \( \dot{v}(t) \), where \( G \in \mathcal{L}_{2 \times 4P}, \dot{v}(t) \in \Delta_P, P \) is an arbitrary positive integer, into (15), by Proposition 3.1, we obtain closed-loop LCN
\[
\dot{x}(t + 1) = [\delta_1^1 \otimes I_2^P, \delta_1^2 \otimes I_2^P, \delta_1^3 \otimes I_2^P, L_4G_4] \dot{x}(t) \dot{u}(t).
\]
where $L_4G_4 = \delta_4[i_1, \ldots, i_P]$, $i_1, \ldots, i_P \in [2, 3]$.

The observability graph of (16) with output function (5) contains a path \{$\delta_4^1, \delta_4^2$\} \(\delta_4^1 \rightarrow \{\delta_4^1, \delta_4^2\} \rightarrow \{\delta_4^1, \delta_4^1\}$, then the LCN is not observable by Lemma 2.7.

Based on the above discussion, we know that state feedback with exogenous input sometimes can enforce observability of an LCN, sometimes cannot. Next we study when a state-feedback controller can enforce observability.

### 3.3 Controller synthesis for enforcing observability of LCNs

The following main result shows that in order to test whether an unobservable LCN can be made observable by state feedback with exogenous input, it is enough to check whether the LCN can be made observable by state feedback.

**Theorem 3.5** Consider an unobservable LCN (3). If it can be made observable by a state-feedback controller (7) with exogenous input, then it can also be made observable by a state-feedback controller (i.e., (7) with $P = 1$).

**Proof** Assume an unobservable LCN (3) and a controller (7) that makes (3) observable. Then by Lemma 2.7, in the observability graph $G_o$ of the corresponding closed-loop LCN $\Sigma$ obtained by substituting (7) into (3),

there exists no cycle in its non-diagonal subgraph, and there exists no edge from any non-diagonal vertex to any diagonal vertex.

(17)

Now consider the structure matrix $G = [G_1, \ldots, G_N]$ of (7), we choose a new state-feedback controller

$$\hat{u}(t) = [\text{Col}_i(G_1), \ldots, \text{Col}_i(G_N)] \hat{x}(t),$$

(18)

where $i \in [1, P]$ is arbitrarily given, and consider the observability graph $G_o'$ of the closed-loop LCN $\Sigma'$ obtained by substituting (18) into (3).

It can be seen that the vertex sets of $G_o$ and $G_o'$ coincide, since $\Sigma$ and $\Sigma'$ have the same output function. One also sees that for every two vertices $v$ and $v'$ in the vertex set, if there exists an edge from $v$ to $v'$ in $G_o'$, then there also exists an edge from $v$ to $v'$ in $G_o$, i.e., the edge set of $G_o'$ is a subset of that of $G_o$. Hence $G_o'$ also satisfies (17), and $\Sigma'$ is also observable by Lemma 2.7.

Because there are infinitely many state-feedback controllers with exogenous input, generally one cannot directly check whether an unobservable LCN can be made observable by state feedback with exogenous input. However, by Theorem 3.5, one can do the above check because there are totally finitely many state-feedback controllers. Formally, the following Theorem 3.6 holds.

**Theorem 3.6** An unobservable LCN (3) can be made observable by state feedback with exogenous input if and only if it can be made observable by state feedback.

By Proposition 3.1 and the proof of Theorem 3.5, the following result holds. The subsequent discussions on observability synthesis will be based on this result.

**Theorem 3.7** An LCN (3) can be made observable by state feedback with exogenous input if and only if there exist $i_1, \ldots, i_N \in [1, M]$ such that the BN

$$\hat{x}(t+1) = [\text{Col}_{i_1}(L_1), \ldots, \text{Col}_{i_N}(L_N)]\hat{x}(t),$$

$$\tilde{y}(t) = H\hat{x}(t),$$

(19)

is observable.

**Remark 3.1** In order to verify whether an unobservable LCN (3) can be made observable by state feedback, one should substitute several state-feedback controllers into the LCN, and then check whether there exists an observable closed-loop LCN. Now we analyze how many state-feedback controllers should be substituted into the LCN in order to do the verification.

Consider LCN (3), it is sufficient to substitute

$$\prod_{i=1}^N |\text{Col}(L_i)|$$

state-feedback controllers into the LCN to do the above verification. It is because, in order not to do repetitive check, for every two of the chosen state-feedback controllers, where their structure matrices are $G_1 = [g_1^1, \ldots, g_N^1]$ and $G_2 = [g_1^2, \ldots, g_N^2]$, respectively, they must satisfy (see (19))

$$[L_1g_1^1, \ldots, L_Ng_N^1] \neq [L_1g_1^2, \ldots, L_Ng_N^2],$$

(20)

which means the obtained closed-loop LCNs are different. On the other hand, in order not to lose any necessary check, it is sufficient to choose $\prod_{i=1}^N |\text{Col}(L_i)|$ state-feedback controllers every two of which satisfy (20) to do the above check.

Next we give some special conditions for whether an unobservable LCN can be made observable by state feedback, which can be checked under much less computational cost than the equivalent condition in Theorem 3.7. In addition, using these conditions we can furthermore reduce the number (shown in Remark 3.1) of state-feedback controllers that are needed to be substituted into the unobservable LCN to do the above check.

**Theorem 3.8** Consider an unobservable LCN (3). The LCN cannot be made observable by any state-feedback controller if at least one of the following holds.
(i) \( L \) satisfies \( L_j = L_k = \delta^j_N \otimes 1^T_M \) for some different \( j, k \in [1, N] \) and some \( l \in [1, N] \), where \( H\delta^j_N = H\delta^k_N \).

(ii) There exist different \( j, k \in [1, N] \) such that \( L_j = \delta^j_N \otimes 1^T_M \) and \( L_k = \delta^k_N \otimes 1^T_M \), or \( L_j = \delta^j_N \otimes 1^T_M \) and \( L_k = \delta^k_N \otimes 1^T_M \), where \( H\delta^j_N = H\delta^k_N \).

**Proof** Assume (i) holds, then in the observability graph of the closed-loop LCN obtained by feeding an arbitrary state-feedback controller into the original unobservable LCN, there exists an edge \( \{\delta^j_N, \delta^k_N\} \xrightarrow{\delta^l_M} \{\delta^j_N, \delta^l_N\} \), where \( \{\delta^j_N, \delta^k_N\} \) is a diagonal vertex. Then by Lemma 2.7, no obtained closed-loop LCN is observable.

Assume (ii) holds, then in the observability graph of the closed-loop LCN obtained by feeding an arbitrary state-feedback controller into the original unobservable LCN, there exists a self-loop \( \{\delta^j_N, \delta^k_N\} \xrightarrow{\delta^l_M} \{\delta^j_N, \delta^k_N\} \) on the non-diagonal vertex \( \{\delta^j_N, \delta^k_N\} \), then also by Lemma 2.7, no obtained closed-loop LCN is observable. \( \square \)

By Theorem 3.8 and the previously obtained results, we show how to further reduce the number (shown in Remark 3.1) of state-feedback controllers that are needed to be substituted into an unobservable LCN to check whether the LCN can be made observable by state feedback. Consider an unobservable LCN (3) and a state-feedback controller

\[
\hat{u}(t) = G\hat{x}(t) = [g_1, \ldots, g_N]\hat{x}(t),
\]

(21)

where \( g_i \in L_{M \times 1}, i \in [1, N] \). Substituting (21) into (3), we obtain a closed-loop LCN

\[
\begin{align*}
\hat{x}(t+1) &= [L_1g_1, \ldots, L_Ng_N]\hat{x}(t), \\
\hat{y}(t) &= H\hat{x}(t)
\end{align*}
\]

(22)

by Proposition 3.1, which is consistent with Eqn. (19).

Denote

\[
\text{Col}(H) = \left\{ \delta^i_Q, \ldots, \delta^k_Q \right\},
\]

(23)

where \( \delta^i_Q, \ldots, \delta^k_Q \) are distinct. For each \( i \in [1, \ell] \), we denote

\[
S_i := \left\{ \delta^j_N \big| j \in [1, N], H\delta^j_N = \delta^i_Q \right\},
\]

(24)

\[
C_i := |S_i|.
\]

The collection of state sets \( S_1, \ldots, S_\ell \) partitions \( \Delta_N \).

In order to make (22) observable, we must assume that for each \( i \in [1, \ell] \), for all different \( j, k \in [1, N] \) with \( \delta^i_N, \delta^k_N \in S_i \), it holds that \( L_jg_j \neq L_fg_k \). Otherwise, in the observability graph of (22), there exists an edge \( \{\delta^i_N, \delta^k_N\} \rightarrow \{L_jg_j, L_fg_k\} \), where \( \{\delta^i_N, \delta^k_N\} \) is a non-diagonal vertex, and \( \{L_jg_j, L_fg_k\} \) is a diagonal vertex, which shows that (22) is not observable by Lemma 2.7. Hence in order to make (22) observable, we must furthermore assume that

\[
|\{L_jg_j \in [1, N], \delta^i_N \in S_i\}| = |C_i| = c_i,
\]

(25)

e.g., \( L_jg_j \) with \( \delta^i_N \in S_i \) are distinct.

The above analysis yields the following result stronger than the first implication in Theorem 3.8 (i.e., (i) implies that (3) cannot be made observable by any state-feedback controller).

**Theorem 3.9** Given an unobservable LCN (3), assume there exists \( i \in [1, \ell] \) (\( \ell \) is defined in (23)) such that for all \( j_1, \ldots, j_c \in [1, M] \) \( c_i \) is defined in (24), \(|\{\text{Col}(L_{j_1}), \ldots, \text{Col}(L_{j_c})\}| < c_i \), then (3) cannot be made observable by any state-feedback controller.

**Proof** Consider an arbitrary given state-feedback controller (21) and the corresponding closed-loop LCN (22) obtained by substituting (21) into the unobservable (3). By assumption, there exists \( i \in [1, \ell] \) and different \( \alpha, \beta \in [1, N] \) such that \( \delta^\alpha_N, \delta^\beta_N \in S_i \) and \( L_\alpha g_\alpha = L_\beta g_\beta \). Hence in the observability graph of (22), there exists an edge \( \{\delta^\alpha_N, \delta^\beta_N\} \rightarrow \{L_\alpha g_\alpha, L_\beta g_\beta\} \), where \( \{\delta^\alpha_N, \delta^\beta_N\} \) is a non-diagonal vertex, \( \{L_\alpha g_\alpha, L_\beta g_\beta\} \) is a diagonal vertex. By Lemma 2.7, (22) is not observable. \( \square \)

**Remark 3.2** Theorem 3.9 is stronger than the first implication in Theorem 3.8, because (i) in Theorem 3.8 is stronger the assumption in Theorem 3.9, but they have the same conclusion (i.e., (3) cannot be made observable by any state-feedback controller).

Based on these analysis, the following result holds.

**Theorem 3.10** Consider an unobservable LCN (3). In order to verify whether (3) can be made observable by state feedback, it is sufficient to substitute

\[
\prod_{i=1}^\ell \text{Num}_i
\]

(26)

state-feedback controllers of the form (21) into (3) to check whether there exists an observable closed-loop LCN, where

\[
\text{Num}_i = |\{(\alpha_{i_1}, \ldots, \alpha_{i_{c_i}}) | \alpha_k \in \text{Col}(L_{i_k}), k \in [1, c_i], \alpha_{i_1}, \ldots, \alpha_{i_{c_i}} \text{ are distinct}\}|,
\]

(27)
ℓ is defined in (23), c₁ and i₁, . . . , iₙ are defined in (24). In addition, for every two of the above (26) chosen state-feedback controllers, their structure matrices $G₁ = [g₁¹, . . . , g₁ₙ]$ and $G₂ = [g₂¹, . . . , g₂ₙ]$ must satisfy

$$[L_j, g^1_j, . . . , L_j, g^1_{jₗₗ}] ≠ [L_j, g^2_j, . . . , L_j, g^2_{jₗₗ}]$$ (28)

for some $j ∈ [1, ℓ]$. Particularly, if (26) is equal to 0, then the unobservable LCN (3) cannot be made observable by state feedback.

**Proof** Firstly, observe that if (26) is equal to 0, then the assumption in Theorem 3.9 is satisfied, and then the unobservable LCN (3) cannot be made observable by state feedback.

On the contrary, if the assumption in Theorem 3.9 is not satisfied, then for all $i ∈ [1, ℓ]$, there exist $j₁, . . . , jₙ ∈ [1, M]$ such that

$$|\{\text{Col}_{j₁}(L_{i₁}), . . . , \text{Col}_{jₙ}(L_{iₙ})\}| = c₁,$$

which implies that $Numi > 0$. Hence (26) is greater than 0.

Secondly, if (26) is greater than 0, we can find state-feedback controllers such that the observability graphs of the obtained closed-loop LCNs must not have an edge from a non-diagonal vertex to a diagonal vertex. We check whether some of these closed-loop LCNs is observable. In order not to do repetitive check, for every two of the found state-feedback controllers, where their structure matrices are denoted by $G₁ = [g₁¹, . . . , g₁ₙ]$ and $G₂ = [g₂¹, . . . , g₂ₙ]$, respectively, they must satisfy that there exists $i ∈ [1, ℓ]$ such that (28) holds, which implies that the obtained two closed-loop LCNs are different. On the other hand, in order not to lose any necessary check, it is sufficient to choose (26) state-feedback controllers every two of which satisfy (28) to do the above check. □

**Remark 3.3** It is easy to see that (26) is no greater than the number $\prod_{i=1}^{n} |\text{Col}(L_{i})|$ shown in Remark 3.1, hence Theorem 3.10 strengthens the result shown in Remark 3.1.

Note that the above number (26) is obtained by avoiding the existence of an edge from a non-diagonal vertex to a diagonal vertex in the observability graph of an obtained closed-loop LCN. In addition to avoiding this, we must also avoid the existence of cycles in the non-diagonal subgraphs of the observability graphs, so the minimal number of state-feedback controllers that are needed to do the above check could be further reduced.

In the remainder of this section, we give an observability synthesis algorithm based on Lemma 2.7 and the results obtained in Section 3.3. To this end, we first give a motivating example.

3.4 A motivating example

**Example 3.11** Consider the following LCN

$$\dot{x}(t + 1) = \deltaₖ[1, 1, 2, 3, 2, 3, 1, 4, 3, 5, 7, 6, 6, 7, 8, 1, 2, 3, 7, 6, 1, 2, 3, 4, 3, 4, 7, 8, 5, 6, 7, 4] \dot{x}(t)u(t),\quad y(t) = \delta₄[1, 1, 1, 1, 1, 2, 2, 2] \dot{x}(t),$$ (29)

where $t ∈ \mathbb{N}, \dot{x}(t) ∈ \Deltaₖ, \dot{u}(t), \dot{y}(t) ∈ Δ₄$. In the observability graph of (29), there exists a path

$$\{\deltaₖ, \delta₄\} \xrightarrow{δ₄} \{\delta₆, \delta₃\} \xrightarrow{δ₃} \{\delta₄, \delta₃\},$$

hence (29) is not observable by Lemma 2.7.

![Observability graph of LCN (31).](image)

Next we try to find a state-feedback controller (if any)

$$\dot{u}(t) = \delta₄[i₁, . . . , iₘ] \dot{x}(t)$$ (30)

to make (29) observable, where $i₁, . . . , iₘ ∈ [1, 4]$.

We might as well firstly choose $i₁ = · · · = iₘ = 1$. Substituting this controller into (29), we obtain the closed-loop LCN

$$\dot{x}(t + 1) = \deltaₖ[1, 2, 3, 6, 2, 1, 3, 5] \dot{x}(t),\quad \dot{y}(t) = \delta₄[1, 1, 1, 1, 1, 2, 2, 2] \dot{x}(t).$$ (31)

The observability graph of (31) (shown in Fig. 4) contains self-loops in its non-diagonal subgraph, hence (31) is not observable by Lemma 2.7.

Secondly, we try to modify $i₁, . . . , iₘ$ to make (29) observable. The basic idea guiding us to choose new $i₁, . . . , iₘ$ is to remove all self-loops and all edges from a non-diagonal vertex to a diagonal vertex in Fig. 4. Since there exists a self-loop on vertex $\{δ₄, δ₃\}$, we keep $i₁ = 1$ invariant, and change $i₂$ from 1 to 2, then the self-loop on $\{δ₄, δ₃\}$ is changed to an edge $\{δ₄, δ₃\} → \{δ₄, δ₃\}$.
Since there also exists a self-loop on vertex \(\{\delta_8^1, \delta_8^3\}\) originally, we change \(i_3\) to 2, then the self-loop on \(\{\delta_8^1, \delta_8^3\}\) is changed to an edge \(\{\delta_8^1, \delta_8^3\} \rightarrow \{\delta_8^5, \delta_8^5\}\). Now consider vertex \(\{\delta_8^5, \delta_8^5\}\). Since originally \(\{\delta_8^5, \delta_8^5\}\) goes to the self-loop on \(\{\delta_8^5, \delta_8^5\}\), we change \(i_5\) from 1 to 3, then there exists no edge from vertex \(\{\delta_8^5, \delta_8^5\}\) to any vertex. After doing these modifications, there exists no edge from vertex \(\{\delta_8^2, \delta_8^2\}\) to any vertex (originally there is a path from \(\delta_8^2, \delta_8^2\) to the diagonal subgraph), there is a path from \(\delta_8^2, \delta_8^2\) to \(\{\delta_8^3, \delta_8^3\}\), and there exists no edge from \(\{\delta_8^3, \delta_8^3\}\) to any vertex (originally there is a self-loop on \(\{\delta_8^3, \delta_8^3\}\)).

Now we substitute the new state-feedback controller

\[
\hat{u}(t) = \delta_4[1, 2, 2, 1, 3, 1, 1, 1]|\hat{x}(t) \tag{32}
\]

into (29), we then obtain a new closed-loop LCN

\[
\dot{x}(t + 1) = \delta_8[1, 3, 5, 6, 7, 1, 3, 5]|\dot{x}(t), \quad \hat{y}(t) = \delta_4[1, 1, 1, 1, 1, 2, 2, 2]|\dot{x}(t). \tag{33}
\]

Luckily, in the observability graph of (33) (shown in Fig. 5), there is no path from any non-diagonal vertex to any cycle, hence (33) is observable by Lemma 2.7. Hence LCN (29) can be made observable by state-feedback controller (32).

Now we compute the upper bounds on the numbers of state-feedback controllers that are needed to be tested in order to verify whether LCN (29) can be made observable by state feedback obtained respectively in Remark 3.1 and Theorem 3.10.

For (29), we have

\[
\text{Col}(H) = \{\delta_1^4, \delta_4^4\},
\]

where we denote \(k_1 = 1\) and \(k_2 = 2\). Then

\[
S_{k_1} = \{\delta_8^1, \delta_8^2, \delta_8^4, \delta_8^5\}, \quad c_1 = |S_{k_1}| = 5,
\]

\[
S_{k_2} = \{\delta_8^6, \delta_8^7, \delta_8^8\}, \quad c_2 = |S_{k_2}| = 3.
\]

\[
\text{Col}(L_1) = \{\delta_8^1, \delta_8^2, \delta_8^3\}, \quad \text{Col}(L_2) = \{\delta_8^4, \delta_8^5, \delta_8^4, \delta_8^4\},
\]

\[
\text{Col}(L_3) = \{\delta_8^2, \delta_8^6, \delta_8^7, \delta_8^8\}, \quad \text{Col}(L_4) = \{\delta_8^3, \delta_8^4, \delta_8^5, \delta_8^6\},
\]

\[
\text{Col}(L_5) = \{\delta_8^2, \delta_8^3, \delta_8^3, \delta_8^3\}, \quad \text{Col}(L_6) = \{\delta_8^1, \delta_8^2, \delta_8^3, \delta_8^5\},
\]

\[
\text{Col}(L_7) = \{\delta_8^1, \delta_8^2, \delta_8^3, \delta_8^5\}, \quad \text{Col}(L_8) = \{\delta_8^1, \delta_8^2, \delta_8^3, \delta_8^3\}.
\]

As shown in Remark 3.1, the upper bound is

\[
\prod_{i=1}^{8} |\text{Col}(L_i)| = 3 \cdot 4^7 = 49152.
\]

As shown in Theorem 3.10, the corresponding upper bound (26) is equal to

\[
\text{Num}_1 \cdot \text{Num}_2 = 153 \cdot 46 = 7038,
\]

where \(\text{Num}_i\)'s are defined by (27).

Next we give a relatively “smarter” method to made LCN (29) observable. We first choose state-feedback controller

\[
\hat{u}(t) = \delta_4[1, 1, 1, 1, 1, 1, 1, 1]|\hat{x}(t). \tag{34}
\]

After substituting (34) into (29), we obtain the following closed-loop LCN

\[
\dot{x}(t + 1) = \delta_8[1, 2, 3, 6, 7, 1, 3, 5]|\dot{x}(t), \quad \hat{y}(t) = \delta_4[1, 1, 1, 1, 1, 2, 2, 2]|\dot{x}(t). \tag{35}
\]

The purpose of choosing (34) is to force the observability graph of the obtained closed-loop LCN (35) to have no edge from a non-diagonal vertex to a diagonal vertex. The observability graph of (35) is shown in Fig. 6. There are three self-loops on three non-diagonal vertices

As shown in Fig. 6, the next step is to remove these three self-loops.
and meanwhile disable appearance of edges from a non-diagonal vertex to a diagonal vertex. Consider the self-loop on vertex \( \delta_k \). We change (34) to the following

\[
\tilde{u}(t) = \delta[1, 4, 1, 3, 1, 1, 1]x(t). 
\]  

(36)

After substituting (36) into (29), we obtain the following closed-loop LCN

\[
\begin{align*}
\tilde{x}(t + 1) &= \delta[1, 4, 3, 6, 7, 1, 3, 5]x(t), \\
\tilde{y}(t) &= \delta[1, 1, 1, 1, 2, 2, 2]x(t).
\end{align*}
\]

(37)

In the observability graph (Fig. 7) of (37), there is still no edge from a non-diagonal vertex to a diagonal vertex, but only one of the previous three self-loops on non-diagonal vertices is left. We furthermore change (36) to the following

\[
\tilde{u}(t) = \delta[1, 4, 2, 1, 3, 1, 1]x(t). 
\]  

(38)

After substituting (38) into into (29), we obtain the following closed-loop LCN

\[
\begin{align*}
\tilde{x}(t + 1) &= \delta[1, 4, 5, 6, 7, 1, 3, 5]x(t), \\
\tilde{y}(t) &= \delta[1, 1, 1, 1, 2, 2, 2]x(t).
\end{align*}
\]

(39)

In the observability graph (Fig. 8) of (39), there is no cycle in the non-diagonal subgraph, LCN (39) is observable.

In the above example, in the first method, we are lucky that the second chosen state-feedback controller (32) makes LCN (29) observable. If we modify \( i_1, \ldots, i_8 \) in another way, the second chosen controller may not make (29) observable. However, due to the existence of (32), we know that finally we will find a state-feedback controller (which might not be (32)) that makes (29) observable.

In the second method, when updating the state-feedback controllers (from (34) to (36), then to (38)), we always have in the observability graphs of the obtained closed-loop LCNs (from (35) to (37), then to (39)), there is no edge from a non-diagonal vertex to a diagonal vertex, which is guaranteed by forcing the first 5 columns of the structure matrices of these closed-loop LCNs to be distinct and also forcing the last 3 columns to be distinct (i.e., (25)). In this sense, the obtained LCNs are closer and closer to be observable, and it is very likely that after several steps, the obtained closed-loop LCN is observable. Hence this method can result in an observability synthesis algorithm that makes an unobservable LCN become observable (if possible) in a large extent, and the speed is not very slow.

3.5 An observability synthesis algorithm for LCNs

The above procedure of modifying state-feedback controllers is similar to the fundamental idea of a greedy algorithm, i.e., in every step of modification, the purpose is to remove at least one (simple) cycle in the non-diagonal subgraph of the corresponding observability graph, so that after several steps, no cycle exists and the obtained closed-loop LCN is observable. Following the procedure, for a given unobservable LCN (3) that can be made observable by state feedback, if the initial state-feedback controller was chosen appropriately, and the following modifications were done appropriately, then the procedure could return a state-feedback controller that makes the unobservable LCN observable. However, because of the high nonlinearity of the observability synthesis problem, there is no guarantee that such an appropriate initial state-feedback controller could be definitely chosen, neither for appropriate following steps of modifications. Hence in order to implement the procedure, one could additionally add the fundamental idea of dynamic programming, i.e., rollback is permitted when in some step, no modification can reduce the number of cycles. To sum up the hybrid procedure containing the ideas of a greedy algorithm and dynamic programming, we show the following Algorithm 1 for observability synthesis.

**Algorithm 1 An observability synthesis algorithm**
Input: An unobservable LCN \( \Sigma \) as in (3)

Output: “Yes” if \( \Sigma \) can be made observable by state feedback, “No” otherwise; in case of “Yes”, a state-feedback controller as in (21) that makes \( \Sigma \) observable

1: initialization: A state-feedback controller \( C \) as in (21) such that the closed-loop LCN \( \Sigma_C \) as in (22) (obtained by feeding \( C \) into \( \Sigma \)) satisfies (25) (this implies that the observability graph of \( \Sigma_C \) contains no edge from any non-diagonal vertex to any diagonal vertex), and a threshold \( 1 \leq \alpha \leq N \)
2: if such an initial controller \( C \) does not exist then
3: return “No”
4: stop
5: else
6: if \( \Sigma_C \) is observable (i.e., the non-diagonal subgraph of its observability graph contains no cycle by Lemma 2.7) then
7: return “Yes” and \( C \)
8: stop
9: else
10: while the current \( \Sigma_C \) is not observable, and, not all closed-loop LCNs as in (22) satisfying (25) have been tested do
11: Choose to modify a number at most \( \alpha \) of columns of the structure matrix of \( C \) so that the number of cycles in the non-diagonal subgraph of the observability graph of the updated \( \Sigma_C \) decreases, \( \Sigma_C \) has not been tested, and meanwhile \( \Sigma_C \) still satisfies (25)
12: if such a modification does not exist then
13: move backward to some of the previously tested controllers until such a modification exists and then do the modification as in Line 11 (if after moving to the initial state-feedback controller but such a modification still does not exist, then reinitialize the procedure as in Line 1)
14: end if
15: if the current \( \Sigma_C \) is observable then
16: return “Yes” and \( C \)
17: stop
18: end if
19: end while
20: return “No”
21: stop
22: end if
23: end if

Algorithm 1 could be refined in several ways to change its running performance, e.g., by adding additional rules, e.g., choosing \( \alpha = 1 \), setting priorities for columns of the structure matrix of \( C \) to be chosen, etc.

Example 3.12 Recall the LCN (29) in Example 3.11. The second method follows the procedure in Algorithm 1, see Table 3.

4 Conclusion

In this paper, we showed that state feedback with exogenous input sometimes can enforce or weaken observability of a logical control network (LCN). We also characterized how to verify whether observability of an LCN can be enforced by state feedback with exogenous input. In addition, we gave an upper bound on the number of state-feedback controllers that are needed to be tested in order to verify whether an unobservable LCN can be made observable by state feedback with exogenous input. Finally, based on the method of obtaining the upper bound, an observability synthesis algorithm was designed by additionally combining the ideas of a greedy algorithm and dynamic programming. Note that the observability synthesis algorithm is preliminary, a lot of work could be done to improve its running performance.

In this paper, we only studied the synthesis problem for observability of LCNs in the sense of Definition 2.2. The other three types of observability as in Definitions 2.3, 2.4, and 2.5 can also be studied by using our observability graph, where in order to study Definitions 2.4 and 2.5, one additionally needs to compute deterministic finite automata from an observability graph as introduced in Section 1.2.

In order to make the obtained results be applied to the simulation-based method for controller synthesis of hybrid systems over their finite abstractions introduced in the Introduction section, a future topic is to generalize the obtained results to nondeterministic finite-transition systems, since usually nondeterministic finite-
transition systems better simulate hybrid systems. In addition, another natural generalization of the paper is to consider output-feedback controllers instead of state-feedback controllers.

References

[1] P. Tabuada. Verification and Control of Hybrid Systems: A Symbolic Approach. Springer Publishing Company, Incorporated, 1st edition, 2009.
[2] C. Belta, B. Yordanov, and E. A. Gol. Formal Methods for Discrete-Time Dynamical Systems. Springer International Publishing AG, 2017.
[3] C. Baier and J. P. Katoen. Principles of Model Checking. The MIT Press, 2008.
[4] T. Akutsu. Algorithms for Analysis, Inference, and Control of Boolean Networks. World Scientific, 2018.
[5] S.A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. Journal of Theoretical Biology, 22(3):437–467, 1969.
[6] T. Idezue, T. Galitski, and L. Hood. A new approach to decoding life: systems biology. Annual Review of Genomics and Human Genetics, 2:343–372, 2001.
[7] A. Fauré, A. Naldi, C. Chaouiya, and D. Thieffry. Dynamic analysis of a generic Boolean model for the control of the mammalian cell cycle. Bioinformatics, 22(14):e124, 2006.
[8] S. Sridharan, R. Layek, A. Datta, and J. Venkatraj. Boolean modeling and fault diagnosis in oxidative stress response. BMC Genomics, 13(Suppl 6), S4:1–16, 2012.
[9] H. Kitano. Systems biology: A brief overview. Science, 295:1662–1664, 2002.
[10] T. Akutsu, S. Miyano, and S. Kuhara. Inferring qualitative relations in genetic networks and metabolic pathways. Bioinformatics, 16 (8):727–734, 2000.
[11] R. Albert and A.-L. Barabási. Dynamics of complex systems: Scaling laws for the period of Boolean networks. Phys. Rev. Lett., 84:5660–5663, Jun 2000.
[12] Y. Zhao, J. Kim, and M. Filippone. Aggregation algorithm towards large-scale boolean network analysis. IEEE Transactions on Automatic Control, 58(8):1976–1985, 2013.
[13] Y. Zhao, Z. Li, and D. Cheng. Optimal control of logical control networks. IEEE Transactions on Automatic Control, 56(8):1760–1776, Aug 2011.
[14] T. Akutsu, M. Hayashida, W. K. Ching, and M. K. Ng. Control of Boolean networks: Hardness results and algorithms for tree structured networks. Journal of Theoretical Biology, 244(4):670–679, 2007.
[15] D. Laschov, M. Margaliot, and G. Even. Observability of Boolean networks: A graph-theoretic approach. Automatica, 49(8):2351–2362, 2013.
[16] Y. Zhao, B.K. Ghosh, and D. Cheng. Control of large-scale boolean networks via network aggregation. IEEE Transactions on Neural Networks and Learning Systems, 27(7):1527–1536, 2016.
[17] K. Zhang and K.H. Johansson. Efficient verification of observability and reconstructibility for large Boolean control networks with special structures. IEEE Transactions on Automatic Control, 65(12):5144–5158, 2020.
[18] D. Cheng. Semi-tensor product of matrices and its application to Morgen’s problem. Science in China Series : Information Sciences, 44(3):195–212, 2001.
[19] D. Cheng and H. Qi. Controllability and observability of Boolean control networks. Automatica, 45(7):1659–1667, 2009.
[20] Y. Zhao, H. Qi, and D. Cheng. Input-state incidence matrix of Boolean control networks and its applications. Systems & Control Letters, 59(12):767–774, 2010.
[21] E. Fornasini and M.E. Valcher. Observability, reconstructibility and state observers of Boolean control networks. IEEE Transactions on Automatic Control, 58(6):1390–1401, June 2013.
[22] K. Zhang and L. Zhang. Observability of Boolean control networks: A unified approach based on finite automata. IEEE Transactions on Automatic Control, 61(9):2733–2738, Sept 2016.
[23] R. Li, M. Yang, and T. Chu. Controllability and observability of Boolean networks arising from biology. Chaos: An Interdisciplinary Journal of Nonlinear Science, 25(2):023104–(1–15), 2015.
[24] K. Zhang, L. Zhang, and R. Su. A weighted pair graph representation for reconstructibility of Boolean control networks. SIAM Journal on Control and Optimization, 54(6):3040–3060, 2016.
[25] D. Cheng and Y. Zhao. Identification of Boolean control networks. Automatica, 47(4):702–710, 2011.
[26] Z. Zhang, T. Leifeld, and P. Zhang. Identification of Boolean control networks incorporating prior knowledge. In 2017 IEEE 56th Annual Conference on Decision and Control (CDC), pages 5839–5844, Dec 2017.
[27] K. Zhang, L. Zhang, and L. Xie. Invertibility and nonsingularity of Boolean control networks. Automatica, 60:155–164, 2015.
[28] Y. Zou and J. Zhu. Kalman decomposition for Boolean control networks. Automatica, 54:65–71, 2015.
[29] Y. Li, J. Zhu, B. Li, Y. Liu, and J. Lu. A necessary and sufficient graphic condition for the original disturbance

| STEP | controller C | closed-loop LCN ΣC | number of cycles in the non-diagonal subgraph of the observability graph of ΣC |
|------|-------------|-------------------|---------------------------------|
| 1    | (34)        | (35)              | 3                               |
| 2    | (36)        | (37)              | 1                               |
| 3    | (38)        | (39)              | 0                               |

Table 3
The second method in Example 3.11 for synthesizing a state-feedback controller that makes LCN (29) observable which follows the procedure shown in Algorithm 1 (rollback is not needed in this example), where all closed-loop LCNs (35), (37), and (39) satisfy (25), i.e., the non-diagonal subgraphs of their observability graphs contain no edge from a non-diagonal vertex to a diagonal vertex. (39) is observable.
decoupling of Boolean networks. *IEEE Transactions on Automatic Control*, 66(8):3765–3772, 2021.

[30] F. Li. Pinning control design for the synchronization of two coupled Boolean networks. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 63 (3):309–313, 2016.

[31] Y. Wu and T. Shen. A finite convergence criterion for the discounted optimal control of stochastic logical networks. *IEEE Transactions on Automatic Control*, 63(1):262–268, Jan 2018.

[32] Y. Wu, R. Zhou, Y. Wu, W. Gui, and C. Yang. Stability and set stability in distribution of probabilistic Boolean networks. *IEEE Transactions on Automatic Control*, page on line, 2018.

[33] R. Li, T. Chu, and X. Wang. Bisimulations of Boolean control networks. *SIAM Journal on Control and Optimization*, 56(1):388–416, 2018.

[34] H. Li and Y. Wang. Output feedback stabilization control design for Boolean control networks. *Automatica*, 49(12):3641–3645, 2013.

[35] G. Hochma, M. Margaliot, E. Fornasini, and M.E. Valcher. Symbolic dynamics of Boolean control networks. *Automatica*, 49(8):2525–2530, 2013.

[36] K. Zhang and L. Zhang. Observability of Boolean control networks: A unified approach based on the theories of finite automata and formal languages. In *Proceedings of the 33rd Chinese Control Conference*, pages 6854–6861, Sept 2014.

[37] K. Zhang, L. Zhang, and L. Xie. *Discrete-Time and Discrete-Space Dynamical Systems*. Communications and Control Engineering. Springer International Publishing, 2020.

[38] D. Cheng, H. Qi, T. Liu, and Y. Wang. A note on observability of Boolean control networks. *Systems & Control Letters*, 87:76–82, 2016.

[39] Q. Zhu, Y. Liu, J. Lu, and J. Cao. Observability of Boolean control networks. *Science China Information Sciences*, 61(9):092201, Oct 2018.

[40] D. Cheng, C. Li, and F. He. Observability of Boolean networks via set controllability approach. *Systems & Control Letters*, 115:22–25, 2018.

[41] S. Wang and H. Li. Graph-based function perturbation analysis for observability of multivalued logical networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(11):4839–4848, 2021.

[42] Y. Guo. Observability of Boolean control networks using parallel extension and set controllability. *IEEE Transactions on Neural Networks and Learning Systems*, 29(12):6402–6408, Dec 2018.

[43] R. Zhou, Y. Guo, and W. Gui. Set reachability and observability of probabilistic Boolean networks. *Automatica*, 106:230–241, 2019.

[44] Y. Yu, M. Meng, J. Feng, and G. Chen. Observability criteria for Boolean networks. *IEEE Transactions on Automatic Control*, page in press, 2021.

[45] T. Li, J. Feng, and B. Wang. Reconstructibility of singular Boolean control networks via automata approach. *Neurocomputing*, 416:19–27, 2020.

[46] K. Zhang, T. Liu, and D. Cheng. Observability of finite labeled transition systems. *IEEE Transactions on Automatic Control*, 63(6):1501–1512, June 2018.

[47] J. Zhao and Z. Liu. Observability of probabilistic Boolean networks. In *2015 34th Chinese Control Conference (CCC)*, pages 183–186, 2015.

[48] E. Fornasini and M.E. Valcher. Observability and reconstructibility of probabilistic Boolean networks. *IEEE Control Systems Letters*, 4(2):319–324, 2020.

[49] R. Li, M. Yang, and T. Chu. Observability conditions of Boolean control networks. *International Journal of Robust and Nonlinear Control*, 24(17):2711–2723, 2014.

[50] K. Zhang, X. Yin, and M. Zamani. Opacity of nondeterministic transition systems: A (bi)simulation relation approach. *IEEE Transactions on Automatic Control*, 64(12):5116–5123, Dec 2019.

[51] M. Zamani, P. Mohajerin Esfahani, R. Majumdar, A. Abate, and J. Lygeros. Symbolic control of stochastic systems via approximately bisimilar finite abstractions. *IEEE Transactions on Automatic Control*, 59(12):3135–3150, 2014.

[52] S. Coogan, M. Arcak, and C. Belta. Finite state abstraction and formal methods for traffic flow networks. In *2016 American Control Conference (ACC)*, pages 864–879, July 2016.

[53] W.M. Wonham. *Linear Multivariable Control: a Geometric Approach*, 3rd Ed. Springer-Verlag New York, 1985.

[54] K. Zhang and K.H. Johansson. Synthesis for controllability and observability of logical control networks. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 108–113, Dec 2019.

[55] D. Cheng, H. Qi, and Z. Li. *Analysis and Control of Boolean Networks: A Semi-tensor Product Approach*. Springer-Verlag London, 2011.

[56] E.F. Moore. Gedanken-experiments on sequential machines. *Automata Studies, Annals of Math. Studies*, 34:129–153, 1956.

[57] Y. Li and J. Zhu. Observability decomposition of Boolean control networks. In *2021 33rd Chinese Control and Decision Conference (CCDC)*, pages 4503–4508, 2021.

[58] K. Zhang, L. Zhang, and L. Xie. Finite automata approach to observability of switched Boolean control networks. *Nonlinear Analysis: Hybrid Systems*, 19:186–197, 2016.

[59] H. Li and Y. Wang. On reachability and controllability of switched Boolean control networks. *Automatica*, 48(11):2917–2922, 2012.

A Concepts and properties related to the STP of matrices

**Definition A.1** The matrix $M_{k_r} = δ^1_k ⊕ · · · ⊕ δ^k_r$ is called the power-reducing matrix. Particularly, we denote $M_2 := M_r$.

By definition, the following result holds.

**Lemma A.2** ([55]) For power-reducing matrix $M_{k_r}$, we have

$$P^2 = M_{k_r} P$$

for each $P ∈ Δ_k$.

**Lemma A.3** ([55]) Let $A ∈ ℝ^{m×n}$ and $z ∈ ℝ^t$. Then

$$A ⊗ z^T = z^T ⊗ (I_t ⊗ A),$$

$$z ⊗ A = (I_t ⊗ A) ⊗ z.$$
Remark A.1 The PBNs studied in [47,43,48,44] are as follows:

\[ \begin{align*}
  x(t + 1) &= L_\sigma(t)x(t), \\
  y(t) &= Hx(t),
\end{align*} \tag{A.1} \]

where \( \sigma : \mathbb{N} \rightarrow [1, s] \) with \( s \in \mathbb{N} \) is an independent and identically distributed process and at each time \( t \in \mathbb{N} \), \( \text{Prob}(\sigma(t) = i) = p_i \), \( i \in [1, s] \), with \( p_i > 0 \) and \( \sum_{i=1}^s p_i = 1 \). \([p_1, \ldots, p_s] =: \mathbf{p} \) is the probability distribution of \( \sigma \); \( L_1, \ldots, L_s \in \mathbb{L}_{2^n \times 2^n} \) are the system matrices; \( H \in \mathbb{L}_{2^n \times 2^n} \) is the output matrix; \( x(t) \in \Delta_{2^n} \); \( y(t) \in \Delta_{2^n} \).

Because \( \sigma \) is independent and identically distributed, (A.1) can be reformulated as the following BCN:

\[ \begin{align*}
  x(t + 1) &= [L_1, \ldots, L_s]u(t)x(t), \\
  y(t) &= Hx(t),
\end{align*} \tag{A.2} \]

where \( u(t) \in \Delta_s \), the other variables are the same as above.

Let \( x(\theta; \sigma, x_0) \) (resp., \( y(\theta; \sigma, x_0) \)) be any of the admissible state (resp., output) sequences of (A.1) starting from initial state \( x_0 \in \Delta_{2^n} \) on discrete time set \([0, \theta]\).

A PBN (A.1) is called observable in probability on \([0, \theta]\) with \( \theta \in \mathbb{N} \) if for every two different \( x_0, x_0' \in \Delta_{2^n} \), it holds that

\[ \text{Prob}\{ y(\theta; \sigma, x_0) \neq y(\theta; \sigma, x_0') \} > 0. \tag{A.3} \]

By definition, (A.3) holds, if and only if, in the corresponding BCN (A.2), \( x_0 \) and \( x_0' \) have a distinguishing input sequence of length \( \theta \). Hence a PBN (A.1) is observable in probability on \([0, \theta]\) with \( \theta \in \mathbb{N} \) if and only if the BCN (A.2) satisfies Definition 2.3 and additionally all pairs of different initial states \( x_0, x_0' \) have a distinguishing input sequence of fixed length \( \theta \).

A PBN (A.1) is called finite-time observable in probability \([43]\) if there is \( \theta \in \mathbb{N} \) such that (A.1) is observable in probability on \([0, \theta]\). Then a PBN (A.1) is finite-time observable in probability if and only if the BCN (A.2) satisfies Definition 2.3.

A PBN (A.1) is called asymptotically observable in distribution \([49]\) if for every two different \( x_0, x_0' \in \Delta_{2^n} \), it holds that

\[ \lim_{\theta \to \infty} \text{Prob}\{ y(\theta; \sigma, x_0) \neq y(\theta; \sigma, x_0') \} = 1. \tag{A.4} \]

If \( x_0 \) and \( x_0' \) have no distinguishing input sequence in (A.2), then \( \text{Prob}\{ y(\theta; \sigma, x_0) \neq y(\theta; \sigma, x_0') \} = 0 \) for any \( \theta \in \mathbb{N} \). Hence in order to make (A.1) asymptotically observable in distribution, every pair of different states must have at least one distinguishing input sequence in (A.2), i.e., (A.2) satisfies Definition 2.3. One can see that (A.2) satisfies Definition 2.3 if and only if \( \theta \) is in the observability graph of (A.2), for every non-diagonal vertex \( v \), either \( \text{outdeg}(v) < 2^n \) or some vertex \( v' \) with \( \text{outdeg}(v') < 2^n \) is reachable from \( v \) \( \text{outdeg}(v') < 2^n \). In order to make (A.1) asymptotically observable in distribution, one additionally must have in the observability graph of (A.2), there is no path from any non-diagonal vertex to any diagonal vertex, because for a non-diagonal vertex \( v = (x_0, x_0') \) from which there is a path to some diagonal vertex, there exists \( \beta < 1 \) such that for any \( \theta \) greater than the length of the path, \( \text{Prob}\{ y(\theta; \sigma, x_0) \neq y(\theta; \sigma, x_0') \} \leq \beta \). Then \( \lim_{\theta \to \infty} \text{Prob}\{ y(\theta; \sigma, x_0) \neq y(\theta; \sigma, x_0') \} \leq \beta < 1 \).

Conversely, assume that (i) (A.2) satisfies Definition 2.3 and (ii) in the observability graph \( G_o := (V, E, W) \) of (A.2), there is no path from any non-diagonal vertex to any diagonal vertex. We endow the edges of \( G_o \) with probabilities according to the probability distribution \( \mathbf{p} = [p_1, \ldots, p_s] \) as in (A.1): for all \( v, v' \in V \) with \( (v', v) \notin E \), \( p_{v,v'} = 0 \); for all \( v, v' \in V \) with \( (v', v) \in E \), \( p_{v,v'} = \sum_{(\sigma, x) \in M(\nu', v) \in W} p_\sigma \). Denote the set of diagonal vertices and the set of non-diagonal vertices of \( G_o \) by \( V_d \) and \( V_{nd} \), respectively. Then by (ii), for all \( v \in V_d \) and \( v' \in V_{nd} \), \( p_{v,v'} = 0 \). Denote the adjacency matrix of the non-diagonal subgraph of \( G_o \) by \( M_{V_{nd}} = (p_{v,v'})_{v,v' \in V_{nd}} \). Then also by (ii), the sum of the \( v \)-th column of \( M_{V_{nd}} \) is equal to \( \text{Prob}\{ y(\theta; \sigma, x_0) = y(\theta; \sigma, x_0') \} \), where \( (x_0, x_0') = v \). By (i) (i.e., for every \( (x_0, x_0') \) in \( V_{nd} \), \( x_0 \) and \( x_0' \) have a distinguishing input sequence in \( G_o \), there exists \( \theta \) in \( \mathbb{N} \) such that in \( M_{V_{nd}} \), the sum of each column is less than 1. Then the spectral radius of \( M_{V_{nd}} \) is less than 1, so is \( M_{V_{nd}} \). Hence \( \lim_{\theta \to \infty} (M_{V_{nd}}) \) has all entries equal to 0. That is, for all \( x_0, x_0' \) with \( (x_0, x_0') \in V_{nd} \), \( \lim_{\theta \to \infty} \text{Prob}\{ y(\theta; \sigma, x_0) = y(\theta; \sigma, x_0') \} = 0 \).

This necessary and sufficient condition is exactly the result shown in [22, Theorem 3.7], [36, Theorem 5.7], [38, Theorem 3.5], and [39, Algorithm 1].
\[
\lim_{\theta \to \infty} \text{Prob}\{y(\theta; \sigma, x_0) \neq y(\theta; \sigma, x'_0)\} = 1.
\]
For all \(x_0, x'_0\) with \(Hx_0 \neq Hx'_0\), \(\text{Prob}\{y(\theta; \sigma, x_0) \neq y(\theta; \sigma, x'_0)\} = 1\) for any \(\theta \in \mathbb{N}\). Then (A.1) is asymptotically observable in distribution.

Based on the above discussion, a PBN (A.1) is asymptotically observable in distribution if and only if the corresponding BCN (A.2) satisfies Definition 2.3 and in the observability graph of (A.2), there is no path from any non-diagonal vertex to any diagonal vertex\(^9\).

The above three conclusions show that the three definitions of observability in probability on \([0, \theta]\) with \(\theta \in \mathbb{N}\), finite-time observability in probability, and asymptotic observability in distribution are rather close to each other and do not depend on probability distributions of the stochastic switching signal \(\sigma\). Formally, given two probability distributions \(p_i, i = 1, 2\), for PBN (A.1), one has PBN (A.1) with \(p = p^1\) is observable if and only if PBN (A.1) with \(p = p^2\) is observable, both in the sense of any one of the above three definitions.

Moreover, [44, Lemma 3] is exactly [56, Theorem 6], and the central idea therein is the equivalence relation \(\sim_k\) for a BCN for which two states have the relation if and only if they do not have any distinguishing input sequence of length \(k\), also see [37, Remark 4.1]. This idea was also used in [21] (see Eqn. (4) therein) to study observability and reconstructibility of BCNs, as well as in [29, 57] to give a necessary and sufficient condition for the disturbance decoupling problem and to solve the observability (Definition 2.3) decomposition problem, in BCNs. In addition, there exist two mistakes in [44, Table II], the time complexity of Algorithm 3.4 of [16] in the table (i.e., of [38] in the current paper) is not \(O(2^{2n})\) but \(O(2^{2n+m})\), the time complexity of Algorithm 1 of [18] in the table (i.e., of [39] in the current paper) is not \(O(2^{2n+m})\) but also \(O(2^{2n+m})\). That is, the algorithms obtained in [38, 39] are more efficient than [44, Algorithm 1].

In [44], observability of switched Boolean networks was mentioned. As one can easily see, observability of switched Boolean control networks can be equivalently transformed to observability of BCNs if inputs and switching signals of switched Boolean control networks are with the same quantifier (either \(\exists\) or \(\forall\)). For a given switched Boolean control network, one could regard the Cartesian product of the set of inputs and the codomain of the switching signal as the new set of inputs so that a new equivalent Boolean control network is obtained. The results in [58] showed that Definition 2.5 extended to switched Boolean control networks for which the switching signals are with the \(\exists\) quantifier is actually Definition 2.5 of BCNs. As a sequence, controllability of switched Boolean control networks studied in [59] (in which inputs and switching signals are both with \(\exists\) quantifier) is actually controllability of BCNs.

\(^9\) This is exactly the result in [44, Proposition 1].