ON IDENTIFICATION OF BOOLEAN CONTROL NETWORKS[*]

BIAO WANG[†], JUN-E FENG[†‡], AND DAIZHAN CHENG[§]

Abstract. A new analytical framework consisting of two phenomena: single sample and multiple samples, is proposed to deal with the identification problem of Boolean control networks (BCNs) systematically and comprehensively. Under this framework, the existing works on identification can be categorized as special cases of these two phenomena. Several effective criteria for determining the identifiability and the corresponding identification algorithms are proposed. Three important results are derived: (1) If a BN is observable, it is uniquely identifiable; (2) If a BCN is O1-observable, it is uniquely identifiable, where O1-observability is the most general form of the existing observability terms; (3) A BN or BCN may be identifiable, but not observable. In addition, remarks present some challenging future research and contain a preliminary attempt about how to identify unobservable systems.

Key words. identification, controllability, observability, Boolean control networks, semi-tensor product.

AMS subject classifications. 06E30, 93B30, 93C55

1. Introduction. Boolean network (BN), proposed by Kauffman in 1969 [16], is an ideal mathematical model of simulating the gene regulation networks. It quantitates the interactions among genes within cells (or within a particular genome). The expression, replication, transcription and other activities of genes can be directly reflected by system states and functions [17]. BN has prompted many researchers to find and ask for similar models. As a result, a large number of models were born. For example, some genes continuously adjust the glucose consumption of cells and so provide the fuel by which they grow and multiply. For analyzing such a biological system, Boolean control network (BCN) becomes a proper model [11, 15]. One of the main tools for studying BNs and BCNs is called the semi-tensor product (STP) of matrices, which was proposed by Prof. Cheng [5, 6, 9]. Its basic idea is to describe the system behavior as a discrete time algebra form, by which, some classical control ideas are incorporated into the analysis of BNs [2, 19, 21, 24, 35] and the control design of BCNs [20, 23, 29, 30, 33, 34], as people have seen in recent years.

Many wild animals carry multiple viruses that have no effect on the animals themselves, but may be both high contagious and deadly to human beings. Antiviral immunity plays a key role against virus diseases. Its research involves the pathologic manifestations, symptoms and detection technologies of viral disease, which is the major cause of network identification being currently an important topic. Network identification aims to find the methods or algorithms for constructing the dynamics of systems. For an unknown biological system or an environment where some viruses survive, only input-output data can be obtained, however, their changes can reflect some particular functions and features of a system. Hence these data are directly used to build the model describing the original complicated network. Some early results

[†]School of Mathematics, Shandong University, Jinan, Shandong 250100, PR China, (wangbiao@sdu.edu.cn, fengjune@sdu.edu.cn).

[‡]Corresponding author

[§]Key Laboratory of Systems and Control, Chinese Academy of Sciences, Beijing 100190, PR China, (dcheng@iss.ac.cn).

considered identification of the network transition mappings [1, 22, 25, 26]. Under the framework of STP, the identification of BNs can be equivalently transformed into the identification of related structure matrices, which was proposed in [7] and was extended to BCNs in [10]. [10] noticed that, a BCN is identifiable if and only if it is controllable and O3-observable. This O3-observability originates from one of five branching paths to the development of observability. We list these five definitions of observability below.

DEFINITION 1.1. *A BCN is Oi-observable, $(i = 1, 2, 3, 4, 5)$, if*
(O1) [32] *for any two distinct states $x(0) \neq \bar{x}(0)$, there exists an input sequence $(u(0), u(1), \ldots)$, such that the corresponding output sequences are distinct: $(y(0), y(1), \ldots) \neq (\bar{y}(0), \bar{y}(1), \ldots)$;*
(O2) [4] *for any a state $x(0)$ there exists an input sequence $(u(0), u(1), \ldots)$, such that for any $\bar{x}(0) \neq x(0)$, the corresponding output sequences are distinct: $(y(0), y(1), \ldots) \neq (\bar{y}(0), \bar{y}(1), \ldots)$;*
(O3) [18] *there exists an input sequence $(u(0), u(1), \ldots)$, such that for any two distinct states $x(0) \neq \bar{x}(0)$, the corresponding output sequences are distinct: $(y(0), y(1), \ldots) \neq (\bar{y}(0), \bar{y}(1), \ldots)$;*
(O4) [12] *for any two distinct states $x(0) \neq \bar{x}(0)$ and for any input sequence $(u(0), u(1), \ldots)$, the corresponding output sequences are distinct: $(y(0), y(1), \ldots) \neq (\bar{y}(0), \bar{y}(1), \ldots)$;*
(O5) [14] *there exists an output-feedback loop $u(t) = f_t(y(t))$ $(u(t) = f(y(t))$ for static control), such that for any two distinct states $x(0) \neq \bar{x}(0)$, the corresponding output sequences are distinct: $(y(0), y(1), \ldots) \neq (\bar{y}(0), \bar{y}(1), \ldots)$.*

Most of the criteria and methods for judging the first four kinds of observability (O1-O4) are not ideal, some are sufficient conditions, some are too complex to apply. [31] proposed a unified approach based on finite automata to determine these four observabilities, and presented corresponding four necessary and sufficient conditions. This automata approach is more suitable for BCNs with fewer state nodes and input nodes due to the high complexity of constructing deterministic finite automata. [8] concentrated on the most general observability (O1), and presented a matrix-based approach with lower complexity by STP. Mathematically speaking, the requirement of input sequences used to recognize the initial state $x(0)$ gradually increases from O1-observability (the most general form) to O4-observability (the sharpest form). Hence, O4 $\implies$ O3 $\implies$ O2 $\implies$ O1, which is shown as a relation diagram in [31]. In particular, determining O3-observability is NP-hard [18]. O5-observability that recognizes the initial state via output feedback, called output-feedback observability, was first proposed in [13]. This one is much sharper than the first two observabilities. [14] used paralled interconnected two identical BCNs to determine O5-observability, by converting the observability problem of the original BCN to the set reachability problem of the interconnected BCN.

As mentioned above, the identification of BCNs requires O3-observability. A natural question arises: what about the most general form, O1-observability? Motivated by that, we further develop the identification problem for BCNs in this paper. Main contributions are summarized as follows:

(1) Three important theoretical results are obtained: (3a) A BN is uniquely identifiable if it is observable; (3b) A BCN is uniquely identifiable if it is O1-observable. It is worth pointing out that O1-observability is the most general one of the existing observability terms. (3c) A BN or BCN may be identifiable, but not observable.

(2) In combination with the phenomena in medical detection, we propose two new

concepts: single sample and multiple samples to deal with the identification problem of BCNs. Based on them, the identification problem is divided into four situations. We point out that the existing works on identification are actually special cases of these four situations.

(3) By virtue of the observability property, we form a one-to-one correspondence between the state and the output sequence. Then four simple criteria to determine the identifiability and four effective algorithms to construct the structure matrices are proposed.

The rest of the paper is organized as follows. Section II contains preliminary notations, fundamental definitions and problem formulation. Section III presents main results on identification of BNs and BCNs, including several discriminant methods for the identification property, several identification algorithms to construct the structure matrices and illustrative examples. Remarks are given to show some challenging and interesting future research. Finally, a table describes the relationships and comparisons of the results obtained in this paper, and Section IV concludes the paper.

2. Preliminaries.

2.1. Semi-tensor product. This section gives some necessary preliminaries. More details can be referred to [9]. First, some notations are listed below:

- $\mathbb{N} = \{0, 1, 2, \ldots\}$: the natural number set.
- $[a, b]_{\mathbb{N}}$: all the natural numbers from a to b.
- δ_n^i: the ith column of the identity matrix I_n.
- $\Delta_n := \{\delta_n^1, \delta_n^2, \ldots, \delta_n^n\}$.
- $\mathbf{1}_n := \sum_{i=1}^{n} \delta_n^i$.
- $[\delta_n^{i_1} \ \delta_n^{i_2} \ \cdots \ \delta_n^{i_m}] := \delta_n[i_1 \ i_2 \ \cdots \ i_m]$.
- $(\delta_n^{i_1}, \delta_n^{i_2}, \ldots, \delta_n^{i_m}) := \delta_n(i_1, i_2, \ldots, i_m)$.
- $Col_i(M)$: the ith column of matrix M.
- $Col(M)$: the set of all columns of M.
- $\mathcal{L}_{m \times n} := \{M | M \in \mathbb{R}^{m \times n}, Col(M) \subseteq \Delta_m\}$.
- $[M]_{i,j}$: the (i, j)th entry of matrix M.
- M^{T}: the transpose of matrix M.
- Kronecker product: $A \otimes B = ([A]_{i,j} \times B)$.
- K-R product: $A * B = C$, $Col_l(C) = Col_l(A) \otimes Col_l(B)$.

DEFINITION 2.1. [9] *The semi-tensor product (STP) of two matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times q}$ is*

$$A \ltimes B = (A \otimes I_{\frac{s}{n}})(B \otimes I_{\frac{s}{p}}),$$

where s is the least common multiple of n and p.

Obviously, the STP becomes the conventional matrix product if $n = p$. Hence the symbol $\ltimes$ is omitted in the sequel.

LEMMA 2.2. [9] *Let $f(x_1, \ldots, x_n)$ be a Boolean function, where $x_1, \ldots, x_n$ are Boolean variables. Within the framework of vector form, f can be converted into $f : \Delta_{2^n} \to \Delta_2$, and there exists a unique matrix $M_f \in \mathcal{L}_{2 \times 2^n}$, called the structure matrix of f, such that*

$$f(x_1, \ldots, x_n) = M_f \ltimes x_1 \ltimes \cdots \ltimes x_n.$$

Consider a BCN with n state nodes, m input nodes and l output nodes as follows:

(2.1)
$$\begin{cases} x_i(t+1) = f_i(u_1(t), \ldots, u_m(t), x_1(t), \ldots, x_n(t)), \\ y_j(t) = h_j(x_1(t), \ldots, x_n(t)), \\ i \in [1, n]_{\mathbb{N}}, \ j \in [1, l]_{\mathbb{N}}, \ t \in \mathbb{N}, \end{cases}$$

where $f_i : \mathcal{D}^{m+n} \to \mathcal{D}$ and $h_j : \mathcal{D}^n \to \mathcal{D}$ are logical functions, $x_i \in \mathcal{D}$, $u_j \in \mathcal{D}$ and $y_k \in \mathcal{D}$ are the state, input and output of the system, respectively.

From Lemma 2.2, each logical function f_i (h_j) has unique structure matrix M_{f_i} (M_{h_j}), then BCN (2.1) can be equivalently transformed into an algebraic form as follows [9]:

(2.2)
$$\begin{cases} x(t+1) = Fu(t)x(t), \\ y(t) = Hx(t), \end{cases}$$

where $F = M_{f_1} * M_{f_2} * \cdots * M_{f_n}$, $H = M_{h_1} * M_{h_2} * \cdots * M_{h_l}$, $x(t) = \ltimes_{i=1}^n x_i(t)$, $u(t) = \ltimes_{j=1}^m u_j(t)$, $y(t) = \ltimes_{k=1}^l y_k(t)$. This form is called the algebraic state space representation of (2.1).

2.2. Problem statement. The identification problem of BCN (2.2) is to construct two structure matrices F and H via available data. Denote

$$\begin{aligned} U_i(p_i) &:= \{u_i(t)\}_{t=0}^{p_i} = (u_i(0), u_i(1), u_i(2), \ldots, u_i(p_i)), \\ X_i(p_i) &:= \{x_i(t)\}_{t=0}^{p_i} = (x_i(0), x_i(1), x_i(2), \ldots, x_i(p_i)), \\ Y_i(p_i) &:= \{y_i(t)\}_{t=0}^{p_i} = (y_i(0), y_i(1), y_i(2), \ldots, y_i(p_i)), \end{aligned}$$

and

$$\begin{aligned} \{U_i(p_i)\} &:= \{u_i(0), u_i(1), u_i(2), \ldots, u_i(p_i)\}, \\ \{X_i(p_i)\} &:= \{x_i(0), x_i(1), x_i(2), \ldots, x_i(p_i)\}, \\ \{Y_i(p_i)\} &:= \{y_i(0), y_i(1), y_i(2), \ldots, y_i(p_i)\}. \end{aligned}$$

DEFINITION 2.3. *A BCN* (2.2) *is said to be identifiable, if its two structure matrices F and H can be determined via available data: input data $U_1(p_1), U_2(p_2), \ldots,$ $U_k(p_k)$ and observed data $Y_1(p_1), Y_2(p_2), \ldots, Y_k(p_k)$.*

A coordinate transformation $\omega = Gx$ could convert (2.2) into the following algebraic form:

(2.3)
$$\begin{cases} \omega(t+1) = GF(I_{2^m} \otimes G^{\mathrm{T}})u(t)\omega(t) =: \widehat{F}u(t)\omega(t), \\ y(t) = HG^{\mathrm{T}}\omega(t) =: \widehat{H}\omega(t). \end{cases}$$

Due to the arbitrariness of state recognition, (2.2) and (2.3) are considered to be identical in the same input-output data frame, so the set of all possible $(\widehat{F}, \widehat{H})$ becomes the equivalence class of (F, H). A identifiable BCN is also said to be *uniquely identifiable* in the sense of equivalence.

ASSUMPTION 1. *This paper assumes the available data is sufficient. In other words, the input data and the observed data contain all possible situations which the system could generate.*

Generally speaking, densely populated cities are good places for virus or infectious diseases, which could spread easily from person to person. The Centers for Disease

Control and Prevention can collect a large number of samples from different patients infected by the same pathogen. Hence, Assumption 1 is reasonable and its implementation requires *multiple samples* from large numbers of patients (urine sample or blood sample or cheek swab), not a *single sample* from one patient, since a single sample may exhibit only part of characteristics of the virus. Multiple samples mean that the observed data may be generated from different initial states, while, single sample means that the observed data is generated from some initial state.

On the basis of the statement above, the identification of BNs and BCNs can be divided into four cases:

Case 1 : the identification process of single sample in the BN records one group of output data $Y(p)$.

Case 2 : the identification process of multiple samples in the BN records k groups of output data $Y_1(p_1), Y_2(p_2), \ldots, Y_k(p_k)$.

Case 3 : the identification process of single sample in the BCN records r groups of input-output data $U_1(p_1), Y_1(p_1), U_2(p_2), Y_2(p_2), \ldots, U_r(p_r), Y_r(p_r)$.

Case 4 : the identification process of multiple samples in the BCN records rk groups of input-output data $U_1^i(p_1), Y_1^i(p_1), U_2^i(p_2), Y_2^i(p_2), \ldots, U_r^i(p_r), Y_r^i(p_r), \ i \in [1, k]_{\mathbb{N}}$.

Both Cases 1 and 3 collect the blood sample from only one patient, while Cases 2 and 4 collect from k patients. Case 3 divides the blood sample into multiple portions (r portions) for testing with a variety of reagents. That is to say, r groups of input-output data are generated from the same initial state, i.e., $x_1(0) = x_2(0) = \cdots = x_r(0)$ (in Case 3). Similarly, $x_1^i(0) = x_2^i(0) = \cdots = x_r^i(0), i \in [1, k]_{\mathbb{N}}$ in Case 4.

3. Identification of BNs and BCNs.

3.1. Identification of BNs.
[7] investigated the identification of the following BN:

$$(3.1) \qquad \begin{cases} x(t+1) = Fx(t), \\ y(t) = x(t), \end{cases}$$

in which the observed data is presented directly by the system state. With a group of observed data $(x_1(0) = \delta_{2^n}^{i_0}, x_1(1) = \delta_{2^n}^{i_1}, \ldots)$, the i_0th column of F can be identified as $Col_{i_0}(F) = \delta_{2^n}^{i_1}$ and hence the next result is obtained.

LEMMA 3.1. [7] (*Multiple samples*) *BN* (3.1) *is uniquely identifiable, if and only if the observed data contains all possible states:*

$$(3.2) \qquad \{Y_1(p_1)\} \cup \{Y_2(p_2)\} \cup \cdots \cup \{Y_k(p_k)\} = \Delta_{2^n}.$$

It is noted that the observed data considered in [7] may consist of several output sequences (i.e., multiple samples), which is reasonable because a system may contains multiple attractors and multiple attractors mean multiple state trajectories. When the system state cannot be directly observed, BN (3.1) becomes

$$(3.3) \qquad \begin{cases} x(t+1) = Fx(t), \\ y(t) = Hx(t). \end{cases}$$

In the process of identifying this system, it is the most important to distinguish the states.

DEFINITION 3.2. *In BN* (3.3), *a state pair* $(x(0), \bar{x}(0))$, $x(0) \neq \bar{x}(0)$ *is said to be distinguishable if the corresponding output sequences generated by them are distinct:*

$(y(0), y(1), \ldots) \neq (\bar{y}(0), \bar{y}(1), \ldots)$. (3.3) *is said to be observable if any state pair is distinguishable.*

Observability means that 2^n distinct initial states generate 2^n distinct groups of observed data, i.e.,

$$(3.4) \qquad (H\delta_{2^n}^i, HF\delta_{2^n}^i, HF^2\delta_{2^n}^i, \ldots) \neq (H\delta_{2^n}^{i'}, HF\delta_{2^n}^{i'}, HF^2\delta_{2}^{i'}, \ldots), \ i \neq i'.$$

Since each state trajectory will fall into an attractor in 2^n steps, the subsequent state trajectory and output trajectory will repeat the previous data. Lemma 1 and Proposition 1 in [12] show the following result.

PROPOSITION 3.3. [12] *BN* (3.3) *is observable if and only if for any* $i \neq i'$,

$$(3.5) \qquad (H\delta_{2^n}^i, HF\delta_{2^n}^i, \ldots, HF^{2^n-1}\delta_{2^n}^i) \neq (H\delta_{2^n}^{i'}, HF\delta_{2^n}^{i'}, \ldots, HF^{2^n-1}\delta_2^{i'}).$$

We call $(H\delta_{2^n}^i, HF\delta_{2^n}^i, \ldots, HF^{2^n-1}\delta_{2^n}^i)$ the *effective output sequence* of state $\delta_{2^n}^i$. An effective output sequence corresponds to a state and its length is 2^n steps. Under the case of Assumption 1, if the system is observable, 2^n distinct effective output sequences can be found by searching and comparing all 2^n-step output sequences from sufficient observed data.

Assume that the following k groups of observed data are sufficient,

$$(3.6) \qquad\qquad Y_j(T_j) = (y_j(0), y_j(1), \ldots, y_j(T_j)), \ j \in [1, k]_{\mathbb{N}}.$$

Let Y_s^j represent the sth 2^n-step output sequence to show up in $Y_j(T_j)$:

$$(3.7) \qquad\qquad Y_s^j = (y_j(s-1), y_j(s), \ldots, y_j(s+2^n-2)), \ s \in [1, T_j']_{\mathbb{N}},$$

where $T_j' = T_j - 2^n + 2$. Then by retrieval from (3.6), an algorithm (Algorithm 3.1) to find 2^n distinct effective output sequences is established, and this algorithm names the ith effective output sequence that occurs in Algorithm 3.1 as Y_i, ($i \in [1, 2^n]_{\mathbb{N}}$).

Algorithm 3.1 Retrieve all distinct effective output sequences

---
Input: data (3.6).
Output: $Y_1, Y_2, \ldots, Y_{2^n}$.
1: set $Y = \emptyset$ and $i = 1$
2: **for** $j = 1; j < k; j++$ **do**
3:     **for** $s = 1; s < T_j'; s++$ **do**
4:         **if** $Y_s^j \in Y$ **then**
5:             break;
6:         **else**
7:             $Y_i = Y_s^j, Y = Y \cup Y_i^j, i = i+1$;
8:         **end if**
9:     **end for**
10: **end for**

THEOREM 3.4. (*Multiple samples*) *BN* (3.3) *is uniquely identifiable if it is observable.*

Proof. From the analysis above, if (3.3) is observable, all distinct effective output sequences $Y_i, i \in [1, 2^n]_{\mathbb{N}}$ can be obtained by Algorithm 3.1 and enough observed data (3.6).

Identify Y_i as the effective output sequence of state $\delta_{2^n}^i$, $i \in [1, 2^n]_{\mathbb{N}}$, then the following k state sequences:

$$(3.8) \qquad X_j(T_j') = (x_j(0), x_j(1), \ldots, x_j(T_j')), \; j \in [1, k]_{\mathbb{N}}$$

can be derived by

$$(3.9) \qquad x_j(t) = \delta_{2^n}^{i_t}, \; t \in [0, T_j'], \text{ if } (y_j(t), y_j(t+1), \ldots, y_j(t + 2^n - 1)) = Y_{i_t}.$$

Combining the observed data (3.6) and the state data (3.8), F and H can be constructed by

$$(3.10) \qquad \begin{cases} x_j(t+1) = Fx_j(t), \; t \in [0, T_j' - 1]_{\mathbb{N}}, \; j \in [1, k]_{\mathbb{N}}, \\ H = [Y_1(0) \; Y_2(0) \; \cdots \; Y_{2^n}(0)], \end{cases}$$

where $Y_i(0)$ represents the first element in Y_i, $i \in [1, 2^n]_{\mathbb{N}}$.

Note that the appointment $\delta_{2^n}^i$ of the effective output sequence Y_i can be changed in any order, due to the arbitrariness of state recognition. Then another $(\widehat{F}, \widehat{H})$ is derived, which is analogous to the case of (2.3). One kind of order corresponds to one coordinate transformation $w = Gx$, and $(\widehat{F}, \widehat{H})$ can be written as:

$$(3.11) \qquad \begin{cases} w(t+1) = GFG^{\mathrm{T}}w(t) =: \widehat{F}w(t), \\ y(t) = Hx(t) = HG^{\mathrm{T}}w(t) =: \widehat{H}w(t), \end{cases}$$

which implies both (F, H) and $(\widehat{F}, \widehat{H})$ belong to an equivalence class. Hence, BN (3.3) is uniquely identifiable. $\square$

On the basis of Theorem 3.4, an algorithm (Algorithm 3.2) to identify (F, H) is established.

Algorithm 3.2 Identify (F, H) for BN (3.3) (Deal with Case 2).

Input: data (3.6).
Output: F, H.
Step 1 : Find all distinct effective output sequences by Algorithm 3.1.
Step 2 : For $j \in [1, k]_{\mathbb{N}}$ and $t \in [0, T_j']$, identify the state sequence (3.8) by (3.9).
Step 3 : Construct F and H based on (3.10).

EXAMPLE 1. *Consider a BN with* 3 *state codes and* 1 *output code, and assume there are two groups of observed data:*

$$(3.12) \qquad \begin{cases} Y_1(14) = \delta_2(2, 1, 1, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 1), \\ \qquad = (y_1(0), y_1(1), y_1(2), y_1(3), \ldots, y_1(14)), \\ Y_2(12) = \delta_2(1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1), \\ \qquad = (y_2(0), y_2(1), y_2(2), y_2(3), \ldots, y_2(12)). \end{cases}$$

Step 1: All distinct effective output sequences are

$$(3.13) \qquad \begin{cases} Y_1 = (y_1(0), y_1(1), \ldots, y_1(7)) & = \delta_2(2, 1, 1, 2, 2, 2, 1, 2), \\ Y_2 = (y_1(1), y_1(2), \ldots, y_1(8)) & = \delta_2(1, 1, 2, 2, 2, 1, 2, 2), \\ Y_3 = (y_1(2), y_1(3), \ldots, y_1(9)) & = \delta_2(1, 2, 2, 2, 1, 2, 2, 2), \\ Y_4 = (y_1(3), y_1(4), \ldots, y_1(10)) & = \delta_2(2, 2, 2, 1, 2, 2, 2, 1), \\ Y_5 = (y_1(4), y_1(5), \ldots, y_1(11)) & = \delta_2(2, 2, 1, 2, 2, 2, 1, 2), \\ Y_6 = (y_1(5), y_1(6), \ldots, y_1(12)) & = \delta_2(2, 1, 2, 2, 2, 1, 2, 2), \\ Y_7 = (y_2(0), y_2(1), \ldots, y_2(7)) & = \delta_2(1, 2, 1, 2, 1, 2, 1, 2), \\ Y_8 = (y_2(1), y_2(2), \ldots, y_2(8)) & = \delta_2(2, 1, 2, 1, 2, 1, 2, 1). \end{cases}$$

Step 2: Two state sequences are identified as

$$(3.14) \quad \begin{cases} X_1(8) = (x_1(0), x_1(1), x_1(2), \ldots, x_1(8)) \\ \qquad = \delta_8(1, 2, 3, 4, 5, 6, 3, 4, 5), \\ X_2(5) = (x_2(0), x_2(1), x_2(2), x_2(3), x_2(4), x_2(5)) \\ \qquad = \delta_8(7, 8, 7, 8, 7, 8). \end{cases}$$

Step 3: Based on $x_1(t+1) = Fx_1(t)$, $t \in [0, 8]_{\mathbb{N}}$, we get

$$x_1(1) = Fx_1(0) \Rightarrow \delta_8^2 = F\delta_8^1, x_1(4) = Fx_1(3) \Rightarrow \delta_8^5 = F\delta_8^4,$$
$$x_1(2) = Fx_1(1) \Rightarrow \delta_8^3 = F\delta_8^2, x_1(5) = Fx_1(4) \Rightarrow \delta_8^6 = F\delta_8^5,$$
$$x_1(3) = Fx_1(2) \Rightarrow \delta_8^4 = F\delta_8^3, x_1(6) = Fx_1(5) \Rightarrow \delta_8^3 = F\delta_8^6.$$

Similarly, from $x_2(t+1) = Fx_2(t)$, $t \in [0, 5]_{\mathbb{N}}$, we get

$$x_2(1) = Fx_2(0) \Rightarrow \delta_8^8 = F\delta_8^7, x_2(2) = Fx_2(1) \Rightarrow \delta_8^7 = F\delta_8^8.$$

It is clear that $F = \delta_8[2\ 3\ 4\ 5\ 6\ 3\ 8\ 7]$. On the other hand,

$$H = [Y_1(0)\ Y_2(0)\ \cdots\ Y_8(0)]$$
$$= [y_1(0)\ y_1(1)\ y_1(2)\ y_1(3)\ y_1(4)\ y_1(5)\ y_2(0)\ y_2(1)]$$
$$= \delta_2[2\ 1\ 1\ 2\ 2\ 2\ 1\ 2].$$

To sum up, the system is identified as:

$$(3.15) \quad \begin{cases} x(t+1) = \delta_8[2\ 3\ 4\ 5\ 6\ 3\ 8\ 7]x(t), \\ y(t) = \delta_2[2\ 1\ 1\ 2\ 2\ 2\ 1\ 2]x(t). \end{cases}$$

If we identify $Y_7 \sim \delta_8^1$, $Y_8 \sim \delta_8^2$, $Y_1 \sim \delta_8^3$, $Y_2 \sim \delta_8^4$, $Y_3 \sim \delta_8^5$, $Y_4 \sim \delta_8^6$, $Y_5 \sim \delta_8^7$ and $Y_6 \sim \delta_8^8$, then the corresponding BN becomes:

$$(3.16) \quad \begin{cases} x(t+1) = \delta_8[2\ 1\ 4\ 5\ 6\ 7\ 8\ 5]x(t), \\ y(t) = \delta_2[1\ 2\ 2\ 1\ 1\ 2\ 2\ 2]x(t). \end{cases}$$

Although two systems (3.15) and (3.16) are derived from different appointments of effective output sequences, both of them belong to an equivalence class.

From this example, we can discern a special case: the observed multiple data are exactly 2^n distinct sequences $(y_i(0), y_i(1), \cdots, y_i(2^N - 1))$, $i = 1, 2, \cdots, 2^n$. The BN can also be identified by appropriately extending the length of the observed data (further collection adds one more element to each sequence). In addition, Algorithms 3.1 and 3.2 also work with insufficient data, in which case, there exist some identical groups of observed data induced by some states that are indistinguishable, such that two structure matrices constructed are of low dimensions ($F \in \mathcal{L}_{2^n \times s}, H \in \mathcal{L}_{2^l \times s}, s < 2^n$).

In previous studies [12, 28], the effective output sequences are used to judge the observability, which is usually implemented by a tool called observability matrix:

$$(3.17) \quad \mathcal{O} = [(PH)^{\mathrm{T}}\ (PHF)^{\mathrm{T}}\ \cdots\ (PHF^{2^n-1})^{\mathrm{T}}]^{\mathrm{T}},$$

where $P = [1\ 2\ \cdots\ 2^l]$ is a row vector consisting of $[1, 2^l]_{\mathbb{N}}$ and aims to extract the superscripts of all column vectors in H. For example, if $H = \delta_2[1\ 2\ 2\ 1]$, $PH = [1\ 2\ 2\ 1]$. The effective output sequence of state $\delta_{2^n}^i$ is recorded in the ith column of $\mathcal{O}$: $Col_i(\mathcal{O}) = [PH\delta_{2^n}^i\ PHF\delta_{2^n}^i\ \cdots\ PHF^{2^n-1}\delta_{2^n}^i]^{\mathrm{T}}$. Therefore, finding 2^n effective output sequences is equivalent to constructing the observability matrix.

EXAMPLE 2. *Recall Example 1.*

Y_i is identified as the of state $\delta_{2^n}^i$. Hence, we have

$$
\mathcal{O} = \begin{bmatrix} Py_1(0) & Py_1(1) & \cdots & Py_1(5) & Py_2(0) & Py_2(1) \\ Py_1(1) & Py_1(2) & \cdots & Py_1(6) & Py_2(1) & Py_2(2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ Py_1(7) & Py_1(8) & \cdots & Py_1(12) & Py_2(7) & Py_2(8) \end{bmatrix}
$$

$$
= \begin{bmatrix} 2 & 1 & 1 & 2 & 2 & 2 & 1 & 2 \\ 1 & 1 & 2 & 2 & 2 & 1 & 2 & 1 \\ 1 & 2 & 2 & 2 & 1 & 2 & 1 & 2 \\ 2 & 2 & 2 & 1 & 2 & 2 & 2 & 1 \\ 2 & 2 & 1 & 2 & 2 & 2 & 1 & 2 \\ 2 & 1 & 2 & 2 & 2 & 1 & 2 & 1 \\ 1 & 2 & 2 & 2 & 1 & 2 & 1 & 2 \\ 2 & 2 & 2 & 1 & 2 & 2 & 2 & 1 \end{bmatrix},
$$

which is the observability matrix of system (3.13).

PROPOSITION 3.5. *If $\mathcal{O}$ and $\widehat{\mathcal{O}}$ are the observability matrix of (3.3) and (3.11), respectively, then*

$$
\mathcal{O} = \widehat{\mathcal{O}}G \text{ or } \mathcal{O}G^{\mathrm{T}} = \widehat{\mathcal{O}}.
$$

Proof. From (3.17), it is clear that:

$$
(3.18) \qquad \widehat{\mathcal{O}} = [(P\widehat{H})^{\mathrm{T}} \ (P\widehat{H}\widehat{F})^{\mathrm{T}} \ \cdots \ (P\widehat{H}\widehat{F}^{2^n-1})^{\mathrm{T}}]^{\mathrm{T}}.
$$

Combining $\widehat{H} = HG^{\mathrm{T}}$ and $\widehat{F} = GFG^{\mathrm{T}}$, we get

$$
\begin{aligned}
\widehat{\mathcal{O}} =& [(PHG^{\mathrm{T}})^{\mathrm{T}} \ (PHG^{\mathrm{T}}GFG^{\mathrm{T}})^{\mathrm{T}} \ \cdots \ (PHG^{\mathrm{T}}(GFG^{\mathrm{T}})^{2^n-1})^{\mathrm{T}}]^{\mathrm{T}} \\
=& [(PHG^{\mathrm{T}})^{\mathrm{T}} \ (PHFG^{\mathrm{T}})^{\mathrm{T}} \ \cdots \ (PHF^{2^n-1}G^{\mathrm{T}})^{\mathrm{T}}]^{\mathrm{T}}, \\
\widehat{\mathcal{O}}G =& [(PHG^{\mathrm{T}})^{\mathrm{T}} \ (PHFG^{\mathrm{T}})^{\mathrm{T}} \ \cdots \ (PHF^{2^n-1}G^{\mathrm{T}})^{\mathrm{T}}]^{\mathrm{T}}G \\
=& [(PHG^{\mathrm{T}}G)^{\mathrm{T}} \ (PHFG^{\mathrm{T}}G)^{\mathrm{T}} \ \cdots \ (PHF^{2^n-1}G^{\mathrm{T}}G)^{\mathrm{T}}]^{\mathrm{T}} \\
=& [(PH)^{\mathrm{T}} \ (PHF)^{\mathrm{T}} \ \cdots \ (PHF^{2^n-1})^{\mathrm{T}}]^{\mathrm{T}} = \mathcal{O}. \qquad \square
\end{aligned}
$$

Introducing the observability matrix $\mathcal{O}$ facilitates MATLAB programming. After constructing such an observability matrix from observed data, $x(t)$ can be determined as $\delta_{2^n}^i$ if $[Py(t) \ Py(t+1) \ \cdots \ Py(t+2^n-1)]^{\mathrm{T}} = Col_i(\mathcal{O})$. Proposition 3.5 shows that the observability matrix of some system becomes that of another system by elementary column transformations, which further explains that different orders (3.8) (or observability matrices) give rise to different systems (F, H). However, these systems belong to an equivalence class.

Remark 3.6. If BN (3.3) is identifiable, then is it observable? Consider a BN with $F = \delta_4[3\ 3\ 4\ 4]$ and $H = \delta_2[1\ 1\ 2\ 1]$. Its state transition diagram is shown in Fig. 1. This BN is unobservable since δ_4^1 and δ_4^2 have same effective output sequence.

When the system is unknown, with sufficient data $Y(8) = \delta_2(1, 2, 1, 1, 1, 1, 1, 1, 1)$ we can only find three distinct output sequences

$$
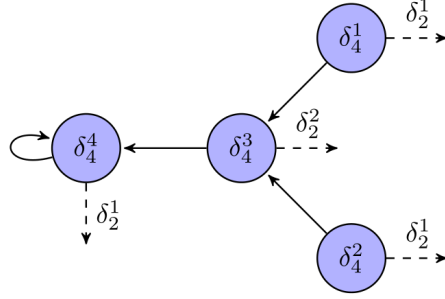Y_1 = \delta_2(1, 2, 1, 1), \ Y_2 = \delta_2(2, 1, 1, 1), \ Y_3 = \delta_2(1, 1, 1, 1).
$$

FIG. 1. *The state transition diagram of an unobservable BN.*

Then the corresponding state sequence can be determined as $X(5) = \delta_4(1, 2, 3, 3, 3, 3)$. If we know $Y_4 = Y_1$, $x(1) = Fx(0)$ produces $\delta_4^2 = F\delta_4^1$ and $\delta_4^2 = F\delta_4^4$. It follows that $F = \delta_4[2\ 3\ 3\ 2]$ and $H = \delta_2[1\ 2\ 1\ 1]$. This system and the original system coincide in the sense of equivalence. This example shows that an unobservable system may be identifiable.

Remark 3.7. To determine the observability, the dimensions of observability matrix $\mathcal{O}$ could be $(2^n - 1) \times 2^n$ (see [12, 28]). That is to say, the condition (3.5) in Proposition 3.3 can be improved as

$$(3.19) \qquad (H\delta_{2^n}^i, HF\delta_{2^n}^i, \ldots, HF^{2^n-2}\delta_{2^n}^i) \neq (H\delta_{2^n}^{i'}, HF\delta_{2^n}^{i'}, \ldots, HF^{2^n-2}\delta_{2^n}^{i'}),$$

which implies that the length of the effective output sequence can be further reduced to $(2^n - 1)$ steps. In a 2^n-step output sequence Y, its first $(2^n - 1)$ steps and last $(2^n - 1)$ steps, respectively denoted by $f_e(Y)$ and $l_e(Y)$, reflect two effective output sequences, and hence they reflect two consecutive states. Assume Y_i and $Y_{i'}$ are two output sequences stemming from $\delta_{2^n}^i$ and $\delta_{2^n}^{i'}$, respectively. If $l_e(Y_i) = f_e(Y_{i'})$, then state $\delta_{2^n}^{i'}$ follows state $\delta_{2^n}^i$, i.e., $\delta_{2^n}^{i'} = F\delta_{2^n}^i$. So, comparing $Y_1, Y_2, \ldots, Y_{2^n}$ generated by Algorithm 3.1, we can directly construct F and H. For example, (3.13) satisfies

$$l_e(Y_1) = f_e(Y_2),\ l_e(Y_2) = f_e(Y_3),\ l_e(Y_3) = f_e(Y_4),\ l_e(Y_4) = f_e(Y_5),$$
$$l_e(Y_5) = f_e(Y_6),\ l_e(Y_6) = f_e(Y_3),\ l_e(Y_7) = f_e(Y_8),\ l_e(Y_8) = f_e(Y_7).$$

Then, $F\delta_8^1 = \delta_8^2, F\delta_8^2 = \delta_8^3, F\delta_8^3 = \delta_8^4, F\delta_8^4 = \delta_8^5, F\delta_8^5 = \delta_8^6, F\delta_8^6 = \delta_8^3, F\delta_8^7 = \delta_8^8, F\delta_8^8 = \delta_8^7$. Obviously, this process is simpler than Algorithm 3.2. However, extending this method to BCNs is not easy and requires consideration of the input sequence.

3.2. Identification of BCNs. The identification problem of BCNs is complicated because more attractors will emerge as the control is embedded. Considering BCN (2.2), the identification of $F \in \mathcal{L}_{2^n \times 2^{m+n}}$ needs the input-state data $(U_1(p_1), X_1(p_1), \ldots, U_k(p_k), X_k(p_k))$ that could cover all the possibilities of $\Delta_{2^{m+n}}$. Then the identification problem is equivalent to checking

$$(3.20) \qquad \bigcup_{i=1}^{k} \{U_i(p_i)X_i(p_i)\} = \Delta_{2^{m+n}},$$

where $U_i(p_i)X_i(p_i)$ represents $(u_i(0)x_i(0), u_i(1)x_i(1), \ldots, u_i(p_i)x_i(p_i))$.

An ideal situation that only one group $(U_1(p_1), X_1(p_1))$ (not k groups) is used to solve the identification problem of F, which was divided into two cases $H = I_{2^n}$

and $H \neq I_{2^n}$, and was analyzed in [10]. The observed data considered in [10] can be seen as a special case of Case 3. Next we slightly explain the method of [10] and use $(U(p), X(p), Y(p))$ instead of $(U_1(p_1), X_1(p_1), Y_1(p_1))$ for simplicity.

Under a designable input sequence $U(p)$, the input-state pair $(u(t), x(t))$ to traverse all the possibilities of $\Delta_{2^{m+n}}$ can be guaranteed by controllability. BCN (2.2) is said to be *controllable*, if for any initial state $x(0) = x_0 \in \Delta_{2^n}$ and any destination state $x_d \in \Delta_{2^n}$, there exists an input sequence $U(T)$, such that $x(T+1) = x_d$ [4]. When the system state is directly observable ($H = I_{2^n}$), the following lemma is immediate.

LEMMA 3.8. [10] *Let H be the identity matrix I_{2^n}, then BCN (2.2) is uniquely identifiable if and only if (2.2) is controllable.*

If a system is controllable, a single test sample could traverse all the possibilities. Hence Lemma 3.8 is based on only one group of input-output data. For the general case of $H \neq I_{2^n}$, the observability is involved again.

The O3-observability property of BCN (2.2) can be viewed as the observability property of BN (3.3) to some extent. Given an input sequence, a BCN is subject to a fixed evolutionary mechanism, similar to a BN. Therefore, our main purpose is to find all distinct output sequences generated by an input sequence that makes (2.2) O3-observable. [10] designed an enough input sequence $U(T')$ which could determine the state sequence $X(T)$ ($T < T'$) by combining the corresponding output sequence $Y(T'+1)$, where $X(T)$ is the previous part of $X(T'+1)$ generated by $U(T')$.

LEMMA 3.9. [10] *BCN (2.2) is uniquely identifiable from input-output data if and only if it is controllable and O3-observable.*

Lemma 3.9 is also based on only one group of input-output data. By controllability, there exists an input sequence $U(T')$, such that the system runs along the following state trajectory $X(T'+1)$ [10]:

$$
\begin{aligned}
X(T'+1) = \ &\underbrace{(x(0), x(1), x(2), \ldots, x(T)}_{controllability},\\
&\underbrace{x(T+1), x(T+2), \ldots, x(\overset{\overset{x(0)}{\shortparallel}}{t_{i_0}})}_{controllability}, \underbrace{x(t_{i_0}+1), x(t_{i_0}+2), \ldots, x(t_{i_0}+p+1)}_{O3-observability},\\
&\underbrace{x(t_{i_0}+p+2), x(t_{i_0}+p+3), \ldots, x(\overset{\overset{x(1)}{\shortparallel}}{t_{i_1}})}_{controllability}, \underbrace{x(t_{i_1}+1), x(t_{i_1}+2), \ldots, x(t_{i_1}+p+1)}_{O3-observability},\\
&\cdots, x(T'+1)),
\end{aligned}
$$
(3.21)

where $T'+1 = t_{i_T} + p$, and $X(T)$ satisfies

$$
\{U(T)X(T)\} = \Delta_{2^{m+n}}. \tag{3.22}
$$

To explain the purpose of driving the system along this state trajectory, we take the second and third rows as an example. The state transition from $x(T+1)$ to $x(t_{i_0}) = x(0)$ aims to find $x(0)$ by controllability, and that from $x(t_{i_0})$ to $x(t_{i_0}+p+1)$ aims to determine the value of $x(0)$ by O3-observability. Continuing this process, BCN (2.2) can be identified if these moments $t_{i_0}, t_{i_1}, \ldots, t_{i_T}$ are in place. Obviously,

it is difficult to design such an input sequence $U(T')$ before the system is known, especially for the input sequence $(u(T+1), u(T+2), \ldots, u(T'))$ after time T, which involves timing accuracy.

Here we provide a method to identify (F, H) with the help of multiple samples (Case 3), which is different from [10] that used a single one. An input sequence is called an O3-*test* of BCN (2.2) if it makes (2.2) O3-observable. From (3.21), we know that the input sequence $(u(T+1), u(T+2), \ldots, u(T'))$ aims to find and identify each state in $(x(0), x(1), \ldots, x(T))$. Next, we simplify this process with the help of Case 3.

Assume that $(u_0', u_1', \ldots, u_p')$ is an O3-test, and $(u_0, u_1, \ldots, u_T)$ satisfies

$$(3.23) \qquad \{u(t)x(t)|u(t) = u_t, t \in [0, T]_{\mathbb{N}}\} = \Delta_{2^{m+n}}.$$

Then an O3-test could infiltrate the state at each time step by the following input sequences:

$$(3.24) \qquad \begin{cases} U_0(p) = (u_0', u_1', \ldots, u_p'), \\ U_1(p+1) = (u_0, u_0', u_1', \ldots, u_p'), \\ U_2(p+2) = (u_0, u_1, u_0', u_1', \ldots, u_p'), \\ \quad \cdots \\ U_{T+1}(p+T+1) = (u_0, u_1, \ldots, u_T, u_0', u_1', \ldots, u_p'). \end{cases}$$

Denote the corresponding observed data stemming from $x_0(0), x_1(0), \ldots, x_{T+1}(0)$ $(x_0(0) = x_1(0) = \cdots = x_{T+1}(0))$ as

$$(3.25) \qquad \begin{cases} Y_0(p+1) = (y_0(0), y_0(1), \ldots, y_0(p+1)), \\ Y_1(p+2) = (y_1(0), y_1(1), \ldots, y_1(p+2)), \\ \quad \cdots \\ Y_{T+1}(p+T+2) = (y_{T+1}(0), y_{T+1}(1), \ldots, y_{T+1}(p+T+2)). \end{cases}$$

Since the system is O3-observable with respect to $(u_0', u_1', \ldots, u_p')$, we can find all distinct output sequences by retrieval from

$$(3.26) \qquad Y^j = (y_j(j), y_j(j+1), \ldots, y_j(j+p+1)), \ j \in [0, T+1]_{\mathbb{N}}.$$

The corresponding retrieval algorithm (Algorithm 3.3) is given.

Algorithm 3.3 Retrieve all distinct output sequences

Input: data (3.25).
Output: $Y_1, Y_2, \ldots, Y_{2^n}$.
 1: set $Y = \emptyset$ and $i = 1$
 2: **for** $j = 0; j < T+1; j++$ **do**
 3:    **if** $Y^j \in Y$ **then**
 4:       continue;
 5:    **else**
 6:       $Y_i = Y^j$, $Y = Y \cup \{Y_i\}$, $i = i + 1$;
 7:    **end if**
 8: **end for**

Identify Y_i as the output sequence of state $\delta_{2^n}^i$ with respect to $(u_0', u_1', \ldots, u_p')$, then BCN (2.2) can be identified by input-output data (3.24) and (3.25). On the basis of the discussion above, we provide an algorithm (Algorithm 3.4) to identify (F, H).

Algorithm 3.4 Identify (F, H) for a controllable and O3-observable BCN (2.2) (Deal with Case 3).

Input: an O3-test $(u'_0, u'_1, \ldots, u'_p)$, and $(u_0, u_1, \ldots, u_T)$ satisfying (3.23).
Output: F, H.
Step 1 : Construct input sequences (3.24) and record the observed data (3.25).
Step 2 : Find all distinct output sequences $Y_1, Y_2, \ldots, Y_{2^n}$ by Algorithm 3.3.
Step 3 : Identify the state sequence $X_{T+1}(T+1) = (x_{T+1}(0), x_{T+1}(1), \ldots, x_{T+1}(T+1))$ by

$$x_{T+1}(t) = \delta_{2^n}^{i_t}, \ t \in [0, T+1]_{\mathbb{N}}, \ \text{if} \ (y_t(t), y_t(t+1), \ldots, y_t(t+p+1)) = Y_{i_t}.$$

Step 4 : Construct F and H based on

$$(3.27) \qquad \begin{cases} x_{T+1}(t+1) = Fu_t x_{T+1}(t), \ t \in [0, T], \\ H = [Y_1(0) \ Y_2(0) \ \cdots \ Y_{2^n}(0)], \end{cases}$$

where $Y_i(0)$ represents the first element in Y_i, $i \in [1, 2^n]_{\mathbb{N}}$.

The following example originates from literature [10], in which an identification process with single input sequence is shown. We use it to show how to use Algorithm 3.4.

EXAMPLE 3. [10] *Consider an O3-observable and controllable BCN* (2.2) *with*

$$(3.28) \qquad \begin{cases} F = \ \delta_4[2 \ 4 \ 1 \ 1 \ 2 \ 3 \ 2 \ 2], \\ H = \ \delta_2[2 \ 1 \ 1 \ 2]. \end{cases}$$

Fix the initial state $x(0) = \delta_4^1$ and set

$$\begin{cases} (u_0, u_1, \ldots, u_{10}) = \delta_2(1, 1, 1, 2, 2, 1, 1, 1, 2, 2, 2), \\ (u'_0, u'_1) = \delta_2(1, 1). \end{cases}$$

Case 3 divides the test sample into enough portions to be tested ($x_0(0) = x_1(0) = \cdots = x_r(0) = \delta_4^1$).

Step 1: 12 groups of input sequence are constructed and are infiltrated to 12 test samples: $x_k(0) = \delta_4^1, k \in [0, 11]$. Then the corresponding output sequences are

$$\begin{aligned}
Y_0(2) &= \delta_2(\underline{2, 1, 2}), \\
Y_1(3) &= \delta_2(2, \underline{1, 2, 2}), \\
Y_2(4) &= \delta_2(2, 1, \underline{2, 2, 1}), \\
Y_3(5) &= \delta_2(2, 1, 2, 2, 1, 2), \\
Y_4(6) &= \delta_2(2, 1, 2, 2, 1, 2, 2), \\
Y_5(7) &= \delta_2(2, 1, 2, 2, 1, \underline{1, 2, 1}) \\
Y_6(8) &= \delta_2(2, 1, 2, 2, 1, 1, 2, 1, 2), \\
Y_7(9) &= \delta_2(2, 1, 2, 2, 1, 1, 2, 1, 2, 2), \\
Y_8(10) &= \delta_2(2, 1, 2, 2, 1, 1, 2, 1, 2, 2, 1), \\
Y_9(11) &= \delta_2(2, 1, 2, 2, 1, 1, 2, 1, 2, 1, 2, 2), \\
Y_{10}(12) &= \delta_2(2, 1, 2, 2, 1, 1, 2, 1, 2, 1, 1, 2, 1), \\
Y_{11}(13) &= \delta_2(2, 1, 2, 2, 1, 1, 2, 1, 2, 1, 1, 1, 2, 2).
\end{aligned}$$

Step 2: All distinct output sequences generated by (u'_0, u'_1) are

$$
\begin{aligned}
Y_1 &= (y_0(0), y_0(1), y_0(2)) = \delta_2(2, 1, 2), \\
Y_2 &= (y_1(1), y_1(2), y_1(3)) = \delta_2(1, 2, 2), \\
Y_3 &= (y_2(2), y_2(3), y_2(4)) = \delta_2(2, 2, 1), \\
Y_4 &= (y_5(5), y_5(6), y_5(7)) = \delta_2(1, 2, 1),
\end{aligned}
$$

which are the last three elements of $Y_0(2)$, $Y_1(3)$, $Y_2(4)$ and $Y_5(7)$, respectively.

Step 3: Since

$$
(y_0(0), y_0(1), y_0(2)) = Y_1, \quad (y_1(1), y_1(2), y_1(3)) = Y_2,
$$
$$
\cdots \qquad\qquad\qquad \cdots
$$
$$
(y_{10}(10), y_{10}(11), y_{10}(12)) = Y_4, \quad (y_{11}(11), y_{11}(12), y_{11}(13)) = Y_2,
$$

one gets the state sequence

$$
\begin{aligned}
X_{11}(11) &= (x_{11}(0), x_{11}(1), \ldots, x_{11}(11)) \\
&= \delta_4(1, 2, 3, 1, 2, 4, 1, 2, 3, 2, 4, 2).
\end{aligned}
$$

Step 4: Based on (3.27), this BCN is identified as

$$
(3.29) \qquad
\begin{cases}
x(t+1) = \delta_4[2\ 3\ 1\ 1\ 2\ 4\ 2\ 2]u(t)x(t), \\
y(t) = \delta_2[2\ 1\ 2\ 1]x(t).
\end{cases}
$$

In the sense of O3-observability, the observability matrix of BCN (2.2) is analogous to the observability matrix of BN (3.3), and can be written as:

$$
(3.30) \qquad
\mathcal{O} =
\begin{bmatrix}
PH \\
PHFu'_0 \\
PHFu'_1 Fu'_0 \\
\vdots \\
PHFu'_p \cdots Fu'_1 Fu'_0
\end{bmatrix}_{(p+2) \times 2^n}
$$

which consists of 2^n distinct output sequences stemming from 2^n initial states under the input sequence $(u'_0, u'_1, \ldots, u'_p)$. For example, the observability matrix $\mathcal{O}$ of (3.29) is

$$
(3.31) \qquad
\mathcal{O} =
\begin{bmatrix}
2 & 1 & 2 & 1 \\
1 & 2 & 2 & 2 \\
2 & 2 & 1 & 1
\end{bmatrix},
$$

where the ith column of $\mathcal{O}$ corresponds with Y_i, $i \in [1, 4]_{\mathbb{N}}$.

From the discussion above, one sees that, the identification problem is transformed into three conditions:

1) The state trajectory of the system covers the entire state space;

2) There exists a method to distinguish all states;

3) There exist methods to implement 1) and 2).

These three conditions can be guaranteed by controllability, observability, and Assumption 1, respectively. Under the case of Assumption 1, we can find 2^n distinct effective output sequences, or 2^n distinct output sequences generated by an O3-test,

(or say the observability matrix $\mathcal{O}$), based on available data. If such output sequences are found or the observability matrix is constructed, the state sequence will be determined, and then the identification problem will be solved.

Case 3 relaxes the limitation on the number of input sequences and output sequences. Although $(T + 2)$ input sequences are used in Algorithm 3.4, it can be implemented in biological experiments by cloning or segmentation, like diagnosis of disease in blood samples. Inspired by the idea of multiple input sequences, next Lemma 3.9 will be extended to the most general form O1-observability.

Considering an O1-observable BCN (2.2), there exist $N := 2^{2n-1} - 2^{n-1}$ input sequences $U_{(i,i')}(p)$, $1 \le i \ne i' \le 2^n$, such that $\delta_{2^n}^i$ and $\delta_{2^n}^{i'}$ can be distinguishable by $U_{(i,i')}(p)$. Arrange these input sequences as

$$(3.32) \qquad U_{s(i,i')}(p) = (u_{s(i,i')}(0), \dots, u_{s(i,i')}(p)) = U_{(i,i')}(p),$$

where the function $s(i, i')$ is defined by the following rule:

$$(3.33) \qquad s(i, i') = \begin{cases} i' - i, & i = 1, \\ \sum_{j=1}^{i-1}(n - j) + i' - i, & i \ge 2. \end{cases}$$

We call (3.32) an O1-*test* of BCN (2.2). Let $Y_{i,s} = (y_i^s(0), y_i^s(1), \dots, y_i^s(p + 1))$ represent the output sequence stemming from $\delta_{2^n}^i$ with respect to $U_s(p)$, where $y_i^s(t) = HFu_s(t-1) \cdots Fu_s(1)Fu_s(0)\delta_{2^n}^i$. Construct data arrays D_i, $i \in [1, 2^n]_{\mathbb{N}}$ as

$$(3.34) \qquad D_i = (Y_{i,1}, Y_{i,2}, \dots, Y_{i,N}).$$

Then for any distinct state $\delta_{2^n}^i$ and $\delta_{2^n}^{i'}$, we have $Y_{i,s(i,i')} \ne Y_{i',s(i,i')}$ and $D_i \ne D_{i'}$.

PROPOSITION 3.10. *BCN* (2.2) *is O1-observable if and only if there exist* 2^n *data arrays constructed by* (3.34) *satisfying* $D_i \ne D_{i'}$, $1 \le i \ne i' \le 2^n$.

Proposition 3.10 shows that 2^n distinct states correspond to 2^n distinct data arrays in the sense of O1-observability. Hence, analogous to the method used in O3-observability, by infiltrating O1-test (3.32) into $x(0), x(1), \dots, x(T + 1)$ at each time step, we construct input sequences $U_0^s, U_1^s, \dots, U_{T+1}^s$, $s \in [1, N]_{\mathbb{N}}$ as

$$(3.35) \qquad \begin{cases} U_0^s = (u_s(0), u_s(1), \dots, u_s(p)), \\ U_1^s = (u_0, u_s(0), u_s(1), \dots, u_s(p)), \\ U_2^s = (u_0, u_1, u_s(0), u_s(1), \dots, u_s(p)), \\ \qquad \dots \\ U_{T+1}^s = (u_0, u_1, \dots, u_T, u_s(0), u_s(1), \dots, u_s(p)), \end{cases}$$

and denote the corresponding observed data as,

$$(3.36) \qquad \begin{cases} Y_0^s = (y_0^s(0), y_0^s(1), \dots, y_0^s(p + 1)), \\ Y_1^s = (y_1^s(0), y_1^s(1), \dots, y_1^s(p + 2)), \\ Y_2^s = (y_2^s(0), y_2^s(1), \dots, y_2^s(p + 3)), \\ \qquad \dots \\ Y_{T+1}^s = (y_{T+1}^s(0), y_{T+1}^s(1), \dots, y_{T+1}^s(T + p + 2)), \end{cases}$$

where $(u_0, u_1, \dots, u_T)$ satisfies (3.23).

All data arrays generated by O1-test are recorded in

$$(3.37) \qquad D_s^j = (y_j^s(j), y_j^s(j + 1), \dots, y_j^s(j + p + 1)), \; j \in [0, T + 1]_{\mathbb{N}}.$$

According the following retrieval algorithm (Algorithm 3.5), we can find all distinct data arrays D_i, $i \in [1, 2^n]_{\mathbb{N}}$.

Algorithm 3.5 Retrieve all distinct data arrays

Input: data (3.37).
Output: $D_1, D_2, \ldots, D_{2^n}$.
 1: set $D = \emptyset$ and $i = 1$
 2: **for** $j = 0$; $j < T + 1$; $j + +$ **do**
 3:     $D^j = (D_1^j, D_2^j, \ldots, D_N^j)$
 4:     **if** $D^j \in D$ **then**
 5:         continue;
 6:     **else**
 7:         $D_i = D^j$, $D = D \cup \{D_i\}$, $i = i + 1$;
 8:     **end if**
 9: **end for**

Identify O_i as the data array stemming from $\delta_{2^n}^i$, then the state sequence $(x(0), x(1), \ldots, x(T+1))$ can be identified by

$$(3.38) \qquad\qquad x(t) = \delta_{2^n}^{i_t}, \text{ if } D^t = D_{i_t}.$$

Combining $(u_0, u_1, \ldots, u_T)$, $(x(0), x(1), \ldots, x(T+1))$ and $(y_{T+1}^1(0), \ldots, y_{T+1}^1(T+1))$, F and H can be easily obtained.

THEOREM 3.11. (*Single sample*) *BCN* (2.2) *is uniquely identifiable if it is controllable and O1-observable.*

On the basis of the discussion above, we give an algorithm (Algorithm 3.6) to identify BCN (2.2) which is O1-observable.

Algorithm 3.6 Identify (F, H) for a controllable and O1-observable BCN (2.2) (Deal with Case 3).

Input: O1-test: $U_1(p), U_2(p), \ldots, U_N(p)$, and $(u_0, u_1, \ldots, u_T)$ satisfying (3.32).
Output: F, H.
Step 1 : Construct input sequences (3.35) and record the observed data (3.36).
Step 2 : Find all distinct data arrays by Algorithm 3.5.
Step 3 : Identify the state sequence $X(T+1) = (x(0), x(1), \ldots, x(T+1))$ by (3.38).
Step 4 : Construct F and H based on $X(T + 1)$, $(u_0, u_1, \ldots, u_T)$, $(y_{T+1}^1(0), \ldots, y_{T+1}^1(T+1))$ and

$$(3.39) \qquad\qquad \begin{cases} x(t+1) = Fu_t x(t), \\ y_{T+1}^1(t) = Hx(t), \\ t \in [0, T]. \end{cases}$$

Algorithm 3.6 is not a complex process, for the method used in [10], should be relatively straightforward. The core of the identification problem is the complete determination of state sequence $X(T + 1)$. Compared with [10] (see (3.21)), the method proposed here considers only the first $T + 1$ states and inputs, and avoids the design of subsequent inputs $u(T + 1), u(T + 2), \ldots$. Note that the determination of the state sequence depends on the appointment of D_i. Hence (F, H) generated by Algorithm 3.6 belongs to the equivalence class of the original system.

Remark 3.12. One idea for proving the necessity is to employ Lemma 3.9 and a fact that O3-observability implies O1-observability. Logically, this is misleading, because, Theorem 3.11 is based on multiple input sequences, and Lemma 3.9 is based on single input sequence. Hence, we have identifiability $\Longleftarrow$ controllability + O1-observability $\Longleftarrow$ controllability + O3-observability.

As predicted, the implement of multiple input sequences (Case 3) allows more numerous systems to be identified. Next we consider whether the conditions of Theorem 3.11 be further relaxed. The answer is yes. In Case 4, a large number of samples from different patients are collected and each sample is divided into multiple portions. If the state sequences $X_j^i(p) = (x_j^i(0), x_j^i(1), \ldots, x_j^i(p))$ and the input sequences $U_j^i(p) = (u_j^i(0), u_j^i(1), \ldots, u_j^i(p))$ cover all the possibilities of $\Delta_{2^{m+n}}$, i.e.,

$$(3.40) \qquad \bigcup_{i=1,j=1}^{k,r} \{U_j^i(p) X_j^i(p)\} = \Delta_{2^{m+n}},$$

then by infiltrating the O1-test to all states $x_j^i(t), i \in [1,k]_\mathbb{N}, j \in [1,r]_\mathbb{N}, t \in [0,p]_\mathbb{N}$, the identification problem can be solved.

DEFINITION 3.13. *Consider BCN* (2.2) *with a set of initial states P_0 and a set of destination states P_d.*
    i) *P_d is set-P_0 controllable, if for any $x_d \in P_d$, there exist a state $x(0) = x_0 \in P_0$ and an input sequence $U(T)$, such that $x(T+1) = x_d$;*
    ii) *(2.2) is set-P_0 controllable, if Δ_{2^n} is set-P_0 controllable.*
    iii) *A state x_0 is controllable, if (2.2) is set-P_0 controllable and P_0 is a singleton set $P_0 = \{x_0\}$.*

THEOREM 3.14. (*Multiple samples*) *Suppose the set of initial states is P_0, then BCN* (2.2) *is uniquely identifiable if it is set-P_0 controllable and O1-observable.*

Proof. Without loss of generality, assume $P_0 = \{\delta_{2^n}^1, \delta_{2^n}^2, \ldots, \delta_{2^n}^k\}$, and the state space is split into $\Delta_{2^n} = P_d^1 \cup P_d^2 \cup \cdots \cup P_d^k$, such that P_d^a is set-$\{\delta_{2^n}^a\}$ controllable, $a \in [1,k]_\mathbb{N}$. (P_0 may contain more (than k) states and we only use those which make the above condition true.)
    Consider state $\delta_{2^n}^a$ and set P_d^a, if $P_d^a = \delta_{2^n}\{d_1^a, d_2^a, \ldots, d_{\theta_a}^a\}$, then for any state $\delta_{2^n}^i \in P_d^a$, there exists an input sequence

$$(3.41) \qquad U_{a \to i}(p_i^a) = (u_{a \to i}(0), u_{a \to i}(1), \ldots, u_{a \to i}(p_i^a)),$$

such that $F u_{a \to i}(p_i^a) \cdots F u_{a \to i}(1) F u_{a \to i}(0) \delta_{2^n}^a = \delta_{2^n}^i$.
    Construct an input sequence as

$$
\begin{aligned}
U_{a \to i}^j(p_i^a + 1) =& (u_{a \to i}^j(0), u_{a \to i}^j(1), \ldots, u_{a \to i}^j(p_i^a), u_{a \to i}^j(p_i^a + 1)) \\
=& : (u_{a \to i}(0), u_{a \to i}(1), \ldots, u_{a \to i}(p_i^a), \delta_{2^m}^j),
\end{aligned}
\tag{3.42}
$$

under which, the state sequence stemming from $\delta_{2^n}^a$ becomes

$$
\begin{aligned}
X_{a \to i}^j(p_i^a + 2) =& (x_{a \to i}^j(0), x_{a \to i}^j(1), \ldots, x_{a \to i}^j(p_i^a + 1), x_{a \to i}^j(p_i^a + 2)) \\
=& : (\delta_{2^n}^a, x_{a \to i}(1), \ldots, \delta_{2^n}^i, x_{a \to i}^j(p_i^a + 2)),
\end{aligned}
\tag{3.43}
$$

then we have

$$(3.44) \qquad \delta_{2^m}^j \delta_{2^n}^i = u_{a \to i}^j(p_i^a + 1) x_{a \to i}^j(p_i^a + 1) \in \{U_{a \to i}^j(p_i + 1) X_{a \to i}^j(p_i + 1)\}.$$

It follows that

$$\Delta_{2^{m+n}} = \bigcup_{j=1}^{2^m} \left\{ \delta_{2^m}^j \delta_{2^n}^i \mid \delta_{2^n}^i \in \Delta_{2^n} \right\}$$

$$= \bigcup_{j=1}^{2^m} \bigcup_{a=1}^{k} \left\{ \delta_{2^m}^j \delta_{2^n}^i \mid \delta_{2^n}^i \in P_d^a \right\}\}$$

$$\subseteq \bigcup_{j=1}^{2^m} \bigcup_{a=1}^{k} \left\{ U_{a \to i}^j(p_i + 1) X_{a \to i}^j(p_i + 1) \mid i \in \{d_1^a, d_2^a, \ldots, d_{\theta_a}^a\} \right\}\}$$

$$(3.45) \qquad = \bigcup_{j=1}^{2^m} \bigcup_{a=1}^{k} \bigcup_{i=1}^{2^n} \left\{ U_{a \to i}^j(p_i + 1) X_{a \to i}^j(p_i + 1) \right\}, \qquad \qquad \square$$

which implies that enough input-state data could cover $\Delta_{2^{m+n}}$.

By infiltrating the O1-test (3.32) to all states in $X_{a \to i}^j(p_i^a + 2)$, BCN (2.2) can be identified.

COROLLARY 3.15. (*Multiple samples*) *Suppose BCN* (2.2) *is O1-observable and the initial state set is* P_0. *If* P_d *is the maximum set which is set-*P_0 *controllable, then* $|P_d|$ *states can be identified.*

Theorem 3.14 and Corollary 3.15 show that, the maximum level of identifying BCN (2.2) depends on the relationship of all possible states stemming from the initial state set. If the controllability is not available, the identification problem can still be well done with the same method. In densely populated cities, the Centers for Disease Control and Prevention can collect a large number of samples, which most likely contain all possible initial states. That is to say, all possible test samples endowed with 2^n distinct initial states are collected. In Case 4, assume that U_s (3.32) is an O1-test of the system, and $x_1^i(0) = x_2^i(0) = \cdots = x_r^i(0) = \delta_{2^n}^i$, $i \in [1, 2^n]_{\mathbb{N}}$. Construct input sequences

$$(3.46) \qquad \begin{cases} U_s^i(p) = (u_s^i(0), u_s^i(1), \ldots, u_s^i(p)) = U_s, \ s \in [1, N]_{\mathbb{N}}, \\ U_{jN+s}^i(p+1) = (u_{jN+s}^i(0), u_{jN+s}^i(1), \ldots, u_{jN+s}^i(p+1)) \\ \qquad = (\delta_{2^m}^j, U_s), \ j \in [1, 2^m]_{\mathbb{N}}, \ s \in [1, N]_{\mathbb{N}}, \end{cases}$$

then we have

$$\Delta_{2^{m+n}} = \bigcup_{j=1}^{2^m} \bigcup_{i=1}^{2^n} \{ \delta_{2^m}^j \delta_{2^n}^i \}$$

$$= \bigcup_{j=1}^{2^m} \bigcup_{i=1}^{2^n} \left\{ u_{jN+s}^i(0) x_{jN+s}^i(0) \right\}$$

$$(3.47) \qquad \subseteq \bigcup_{j=1}^{2^m} \bigcup_{i=1}^{2^n} \left\{ U_{jN+s}^i(p) X_{jN+s}^i(p) \right\}.$$

Then the system can be easily identified analogous to Theorem 3.14. Hence we have the following result under the case of Assumption 1.

THEOREM 3.16. (*Multiple samples*) *BCN* (2.2) *is uniquely identifiable if it is O1-observable.*

On the basis of the analysis above, the corresponding identification algorithm (Algorithm 3.7) can be established.

Algorithm 3.7 Identify an O1-observable BCN (2.2) (Deal with Case 4).

Input: O1-test (3.32): $U_1(p), U_2(p), \ldots, U_N(p)$.

Output: F, H.

Step 1 : Construct input sequences (3.46) and record the observed data

$$(3.48)
\begin{cases}
Y_s^i(p+1) = (y_s^i(0), y_s^i(1), \ldots, y_s^i(p+1)), \ s \in [1, N]_{\mathbb{N}}, \\
Y_{jN+s}^i(p+2) = (Y_{jN+s}^i(0), \ldots, Y_{jN+s}^i(p+2)), \ j \in [1, 2^m]_{\mathbb{N}}, \ s \in [1, N]_{\mathbb{N}}.
\end{cases}$$

Step 2 : Find all distinct data arrays by Algorithm 3.8.

Step 3 : Identify the states $x_1^i(0), x_{jN+1}^i(1), j \in [1, 2^m]_{\mathbb{N}}, i \in [1, 2^n]_{\mathbb{N}}$, by

$$(3.49)
\begin{cases}
x_1^i(0) = \delta_{2^n}^{a_i}, \ \text{if} \ (Y_1^i(p+1), \ldots, Y_N^i(p+1)) = D_{a_i}, \\
x_{Nj+1}^i(1) = \delta_{2^n}^{a_i}, \ \text{if} \ (\overline{Y}_{jN+1}^i, \ldots, \overline{Y}_{jN+s}^i) = D_{a_i},
\end{cases}$$

where $\overline{Y}_{jN+s}^i = (y_{jN+s}^i(1), \ldots, y_{jN+s}^i(p+2))$.

Step 4 : Construct F and H based on

$$(3.50)
\begin{cases}
x_{Nj+1}^i(1) = F\delta_{2^m}^j x_{Nj+1}^i(0) = F\delta_{2^m}^j x_1^i(0), \\
y_1^i(0) = Hx_1^i(0).
\end{cases}$$

Algorithm 3.8 Retrieve all distinct data arrays

Input: data (3.48).

Output: $D_1, D_2, \ldots, D_{2^n}$.

 1: set $D = \emptyset$ and $a = 1$
 2: **for** $i = 0; i < 2^n; i++$ **do**
 3:   **for** $j = 0; j < 2^n; j++$ **do**
 4:     **if** $j = 0$ **then**
 5:       $D_s^j = Y_s^i(p+1)$
 6:     **else**
 7:       $D_s^j = (y_{jN+s}^i(1), \ldots, y_{jN+s}^i(p+2))$
 8:     **end if**
 9:     $D^j = (D_1^j, D_2^j, \ldots, D_N^j)$
10:     **if** $D^j \in D$ **then**
11:        continue;
12:     **else**
13:        $D_a = D^j, D = D \cup D_a, a = a + 1$;
14:     **end if**
15:   **end for**
16: **end for**

Remark 3.17. There are two ways to construct the structure matrix H. One is based on the state sequence and the output sequence, like (3.39) and (3.50) (in Algorithms 3.6 and 3.7), the other is based on the first elements of the effective

output sequences or the output sequences generated by O3-test or the data arrays generated by O1-test, like (3.10) and (3.27) (in Algorithms 3.2 and 3.4).

Remark 3.18. It is important to point out that, the input sequence required for identification is assumed to be known, whether in [10] (used O3-observability) or our paper (O1-observability). For unknown systems, different input sequences are used to implement Algorithm 3.7 (Algorithm 3.4, Algorithm 3.6) until the system is fully identified. The former (O3-observability) covers an NP-hard problem[18], which has been mentioned in the introduction part. So does our method relax, in a sense, the difficulty of dealing with the identification problem? In other words, is the difficulty of finding an O1-test also NP-hard?

EXAMPLE 4. *Consider the reduced model for the lac operon in the bacterium Escherichia coli. This BCN has three state nodes $\{x_1, x_2, x_3\}$ and three input nodes $\{u_1, u_2, u_3\}$. x_1: lac mRNA, x_2: lactose in high concentration, x_3: lactose in medium concentration, u_1: extracellular glucose, u_2: high extracellular lactose, and u_3: medium extracellular lactose. The dynamics of this system can be written as [27]*

$$(3.51) \quad \begin{cases} x_1(t+1) = \neg u_1(t) \wedge (x_2(t) \vee x_3(t)), \\ x_2(t+1) = \neg u_1(t) \wedge u_2(t) \wedge x_1(t), \\ x_3(t+1) = \neg u_1(t) \wedge (u_2(t) \vee (u_3(t) \wedge x_1(t))). \end{cases}$$

This BCN is O1-observable when the outputs are considered as [3]

$$(3.52) \quad \begin{cases} y_1(t) = x_1(t) \vee \neg x_2(t) \vee x_3(t), \\ y_2(t) = \neg x_1(t) \vee x_2(t) \wedge \neg x_3(t), \\ y_3(t) = \neg x_1(t) \wedge \neg x_2(t) \vee x_3(t). \end{cases}$$

Its algebraic form is

$$(3.53) \quad \begin{cases} F = \delta_8[8\,8\,8\,8\,8\,8\,8\,8\,8\,8\,8\,8\,8\,8\,8\,8 \\ 8\,8\,8\,8\,8\,8\,8\,8\,8\,8\,8\,8\,8\,8\,8\,8 \\ 1\,1\,1\,5\,3\,3\,3\,7\,1\,1\,1\,5\,3\,3\,3\,7 \\ 3\,3\,3\,7\,4\,4\,4\,8\,4\,4\,4\,8\,4\,4\,4\,8], \\ H = \delta_8[8\,6\,3\,6\,5\,6\,7\,6]. \end{cases}$$

Here we analyze the identification problem of this system. Fix the initial state $x_1^i(0) = \cdots = x_r^i(0) = \delta_8^i$, $i \in [1,8]_{\mathbb{N}}$ and choose the following O1-test

$$(3.54) \quad U_s = \begin{cases} (\delta_8^5), & s \in S, \\ (\delta_8^1), & s \in S^c, \end{cases}$$

where $S = \{9, 10, 11, 12, 13, 20, 21, 22, 27\}$ and $S^c = [1, 28]_{\mathbb{N}} \setminus \{9, 10, 11, 12, 13, 20, 21, 22, 27\}$.

Step 1: Construct input sequences

$$\begin{cases} U_s^i(0) = (u_s^i(0)) = U_s, \ s \in [1, 28]_{\mathbb{N}}, \\ U_{28j+s}^i(1) = (u_{28j+s}^i(0), u_{28j+s}^i(1)) = (\delta_8^j, U_s), \ j \in [1, 8]_{\mathbb{N}} \ s \in [1, 28]_{\mathbb{N}}, \end{cases}$$

then the output sequences stemming from states $x_1^1(0), x_2^1(0), \ldots, x_r^1(0)$ are

$$
\begin{cases}
Y_s^1(1) = \delta_8(\underline{8,6}), \ s \in S^c, \\
Y_s^1(1) = \delta_8(\underline{8,8}), \ s \in S, \\
Y_{28+s}^1(2) = Y_{56+s}^1(2) = Y_{84+s}^1(2) = Y_{112+s}^1(2) = \delta_8(8, 6, 6), \ s \in S^c, \\
Y_{28+s}^1(2) = Y_{56+s}^1(2) = Y_{84+s}^1(2) = Y_{112+s}^1(2) = \delta_8(8, \underline{6,7}), \ s \in S, \\
Y_{140+s}^1(2) = Y_{168+s}^1(2) = \delta_8(8, 8, 6), \ s \in S^c, \\
Y_{140+s}^1(2) = Y_{168+s}^1(2) = \delta_8(8, 8, 8), \ s \in S, \\
Y_{196+s}^1(2) = \delta_8(8, \underline{3,6}), \ s \in S^c, \\
Y_{196+s}^1(2) = \delta_8(8, \underline{3,8}), \ s \in S, \\
Y_{224+s}^1(2) = \delta_8(8, \underline{6,6}), \ s \in S^c, \\
Y_{224+s}^1(2) = \delta_8(8, \underline{6,5}), \ s \in S,
\end{cases}
$$

and others are shown in Appendix A.

   Step 2: All distinct data arrays generated by the O1-test (3.54) are

$$
D_1 = (D_1^1, D_2^1, \ldots, D_{28}^1), \quad
\begin{cases}
D_s^1 = \delta_8(8,6), \ s \in S^c, \\
D_s^1 = \delta_8(8,8), \ s \in S,
\end{cases}
$$

$$
D_2 = (D_1^2, D_2^2, \ldots, D_{28}^2), \quad
\begin{cases}
D_s^2 = \delta_8(6,6), \ s \in S^c, \\
D_s^2 = \delta_8(6,7), \ s \in S,
\end{cases}
$$

$$
\ldots
$$

$$
D_8 = (D_1^8, D_2^8, \ldots, D_{28}^8), \quad
\begin{cases}
D_s^8 = \delta_8(6,6), \ s \in S^c, \\
D_s^8 = \delta_8(6,3), \ s \in S,
\end{cases}
$$

which are the last two elements of $Y_s^1(1), Y_{28+s}^1(2), Y_{196+s}^1(2), Y_{224+s}^1(2), Y_s^2(2),$ $Y_{196+s}^4(2), Y_s^5(1)$ and $Y_s^6(1)$, respectively.

   Step 3: Identify D_i as the data array stemming from δ_8^i, $i \in [1,8]_\mathbb{N}$, then from Step 2, we have

$$
\begin{aligned}
x_{28+1}^1(1) &= F u_{28+1}^1(0) x_{28+1}^1(0) \Rightarrow \delta_8^2 = F \delta_8^1 \delta_8^1, \\
x_{56+1}^1(1) &= F u_{56+1}^1(0) x_{56+1}^1(0) \Rightarrow \delta_8^2 = F \delta_8^2 \delta_8^1, \\
x_{84+1}^1(1) &= F u_{84+1}^1(0) x_{84+1}^1(0) \Rightarrow \delta_8^2 = F \delta_8^3 \delta_8^1, \\
x_{112+1}^1(1) &= F u_{112+1}^1(0) x_{112+1}^1(0) \Rightarrow \delta_8^2 = F \delta_8^4 \delta_8^1, \\
x_{140+1}^1(1) &= F u_{140+1}^1(0) x_{140+1}^1(0) \Rightarrow \delta_8^1 = F \delta_8^5 \delta_8^1, \\
x_{168+1}^1(1) &= F u_{168+1}^1(0) x_{168+1}^1(0) \Rightarrow \delta_8^1 = F \delta_8^6 \delta_8^1, \\
x_{196+1}^1(1) &= F u_{196+1}^1(0) x_{196+1}^1(0) \Rightarrow \delta_8^3 = F \delta_8^7 \delta_8^1, \\
x_{224+1}^1(1) &= F u_{224+1}^1(0) x_{224+1}^1(0) \Rightarrow \delta_8^4 = F \delta_8^8 \delta_8^1, \\
&\quad\quad \ldots
\end{aligned}
$$

and

$$
\begin{aligned}
y_1^1(0) &= H x_1^1(0) \Rightarrow \ \delta_8^8 = H \delta_8^1, & y_1^2(0) &= H x_1^2(0) \Rightarrow \ \delta_8^6 = H \delta_8^5, \\
y_1^3(0) &= H x_1^3(0) \Rightarrow \ \delta_8^3 = H \delta_8^3, & y_1^4(0) &= H x_1^4(0) \Rightarrow \ \delta_8^6 = H \delta_8^4, \\
y_1^5(0) &= H x_1^5(0) \Rightarrow \ \delta_8^5 = H \delta_8^7, & y_1^6(0) &= H x_1^6(0) \Rightarrow \ \delta_8^6 = H \delta_8^8, \\
y_1^7(0) &= H x_1^7(0) \Rightarrow \ \delta_8^7 = H \delta_8^6, & y_1^8(0) &= H x_1^8(0) \Rightarrow \ \delta_8^6 = H \delta_8^2.
\end{aligned}
$$

This BCN therefore is identified as

$$(3.55) \quad \begin{cases} F = \delta_8[2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2 \\ \qquad\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2 \\ \qquad\ 1\ 6\ 1\ 7\ 1\ 3\ 3\ 3\ 1\ 6\ 1\ 7\ 1\ 3\ 3\ 3 \\ \qquad\ 3\ 2\ 3\ 6\ 3\ 4\ 4\ 4\ 4\ 2\ 4\ 2\ 4\ 4\ 4\ 4], \\ H = \delta_8[8\ 6\ 3\ 6\ 6\ 7\ 5\ 6]. \end{cases}$$

Remark 3.19. A BN may be identifiable but not observable, as mentioned in Remark 3.6. Similar to the method in BNs (see Remark 3.6), it is still possible to identify an unobservable BCN. Therefore, observability is a stronger property than identifiability in BNs and BCNs.

Remark 3.20. About the identification algorithm in the Matlab programming, the storage method and storage space of the identification standard affect the execution time of the algorithm, because the programming involves the storage and retrieval of data. Hence, it is vital to plan for storage space needs at the beginning of the design phase. Compared with 2^n distinct effective output sequences or 2^n distinct output sequences generated by an O3-test or the corresponding observability matrix, 2^n distinct data arrays generated by an O1-test require more storage space. How to adjust the storage method and how to reduce the storage space are two challenging and interesting topics.

Up to now, we have provided several ways to deal with different situations. The comparison of them is needed and is shown in Table I, where Case 3' considers one group of input-output data, a special case of Case 3.

Approach	System	Case	Condition 1	Condition 2
Lemma 3.1 [7]	BN	Case 2		$H = I_{2^l}$
Theorem 3.4 Algorithm 3.2	BN	Case 2		Observability
Lemma 3.8[10]	BCN	Case 3	controllability	$H = I_{2^l}$
Lemma 3.9[10] Algorithm 3.4	BCN	Case 3' Case 3	controllability	O3-observability
Theorem 3.11 Algorithm 3.6	BCN	Case 3	controllability	O1-observability
Theorem 3.14	BCN	Case 4	Set controllability	O1-observability
Theorem 3.16 Algorithm 3.7	BCN	Case4		O1-observability

TABLE 1
A comparison table of various methods

4. Conclusions. In this paper, we systematically explored the identification problem of BNs and BCNs, gained new cognition. Based on the practical application, we built a new analytical framework that consists of single sample and multiple samples, and then divided the identification problem into four situations. Four simple criteria were proposed for determining the identifiability of BNs and BCNs, and the corresponding identification algorithms were provided to identify related structure matrices. It is worth noting that these algorithms are easy to implement by MAT-LAB. Under this analytical framework, we found three novel and important results: (1) A BN is uniquely identifiable if it is observable; (2) A BCN is uniquely identifiable

if it is O1-observable; (3) The necessity of (1) or (2) does not hold. At last, we presented a table to reveal the relationships of the proposed results, which could be used to further analyze the relationships of identifiability, controllability, O1-observability and O3-observability, as a foreshadowing.

The authors believe that this new analytical framework is a very powerful explanation tool. On the basis of this paper, there are several natural and interesting problems remaining for further study. For example: (1) If a BN or BCN is unobservable, some states produce same output sequence, which implies that some columns in the structure matrices F and H are indeterminate. A very natural question is whether an unobservable system can be identified, or, how to identify an unobservable system? (2) The assumption (Assumption 1) that the input-output data is sufficient is a highlight of this paper. How to generalize this idea to other systems, like singular BNs or switched BCNs?

Appendix A. The observed data of Step 1 in Example 4. The output sequences stemming from states $x_1^2(0), x_2^2(0), \ldots, x_r^2(0)$ are

$$
\begin{cases}
Y_s^2(1) = \delta_8(\underline{6,6}), \ s \in S^c, \\
Y_s^2(1) = \delta_8(\underline{6,8}), \ s \in S, \\
Y_{28+s}^2(2) = \overline{Y_{56+s}^2}(2) = Y_{84+s}^2(2) = Y_{112+s}^2(2) = \delta_8(6,6,6), \ s \in S^c, \\
Y_{28+s}^2(2) = Y_{56+s}^2(2) = Y_{84+s}^2(2) = Y_{112+s}^2(2) = \delta_8(6,6,7), \ s \in S, \\
Y_{140+s}^2(2) = Y_{168+s}^2(2) = \delta_8(6,8,6), \ s \in S^c, \\
Y_{140+s}^2(2) = Y_{168+s}^2(2) = \delta_8(6,8,8), \ s \in S, \\
Y_{196+s}^2(2) = \delta_8(6,3,6), \ s \in S^c, \\
Y_{196+s}^2(2) = \delta_8(6,3,8), \ s \in S, \\
Y_{224+s}^2(2) = \delta_8(6,6,6), \ s \in S^c, \\
Y_{224+s}^2(2) = \delta_8(6,6,5), \ s \in S.
\end{cases}
$$

The output sequences stemming from states $x_1^3(0), x_2^3(0), \ldots, x_r^3(0)$ are

$$
\begin{cases}
Y_s^3(1) = \delta_8(3,6), \ s \in S^c, \\
Y_s^3(1) = \delta_8(3,8), \ s \in S, \\
Y_{28+s}^3(2) = Y_{56+s}^3(2) = Y_{84+s}^2(2) = Y_{112+s}^3(2) = \delta_8(3,6,6), \ s \in S^c, \\
Y_{28+s}^3(2) = Y_{56+s}^3(2) = Y_{84+s}^2(2) = Y_{112+s}^3(2) = \delta_8(3,6,7), \ s \in S, \\
Y_{140+s}^3(2) = Y_{168+s}^3(2) = \delta_8(3,8,6), \ s \in S^c, \\
Y_{140+s}^3(2) = Y_{168+s}^3(2) = \delta_8(3,8,8), \ s \in S, \\
Y_{196+s}^3(2) = \delta_8(3,3,6), \ s \in S^c, \\
Y_{196+s}^3(2) = \delta_8(3,3,8), \ s \in S, \\
Y_{224+s}^3(2) = \delta_8(3,6,6), \ s \in S^c, \\
Y_{224+s}^3(2) = \delta_8(3,6,5), \ s \in S.
\end{cases}
$$

The output sequences stemming from states $x_1^4(0), x_2^4(0), \ldots, x_r^4(0)$ are

$$
\begin{cases}
Y_s^4(1) = \delta_8(6,6), \ s \in S^c, \\
Y_s^4(1) = \delta_8(6,5), \ s \in S, \\
Y_{28+s}^4(2) = Y_{56+s}^4(2) = Y_{84+s}^4(2) \\
\qquad = Y_{112+s}^4(2) = Y_{224+s}^4(2) = \delta_8(6,6,6), \ s \in S^c, \\
Y_{28+s}^4(2) = Y_{56+s}^4(2) = Y_{84+s}^4(2) \\
\qquad = Y_{112+s}^4(2) = Y_{224+s}^4(2) = \delta_8(6,6,7), \ s \in S, \\
Y_{140+s}^4(2) = Y_{168+s}^4(2) = \delta_8(6,5,6), \ s \in S^c, \\
Y_{140+s}^4(2) = Y_{168+s}^4(2) = \delta_8(6,5,3), \ s \in S, \\
Y_{196+s}^4(2) = \delta_8(6,\underline{7,6}), \ s \in S^c, \\
Y_{196+s}^4(2) = \delta_8(6,\underline{7,3}), \ s \in S.
\end{cases}
$$

The output sequences stemming from states $x_1^5(0), x_2^5(0), \ldots, x_r^5(0)$ are

$$
\begin{cases}
Y_s^5(1) = \delta_8(5, 6), \ s \in S^c, \\
Y_s^5(1) = \delta_8(\overline{5, 3}), \ s \in S, \\
Y_{28+s}^5(2) = \overline{Y_{56+s}^5}(2) = Y_{84+s}^5(2) = Y_{112+s}^5(2) = \delta_8(5, 6, 6), \ s \in S^c, \\
Y_{28+s}^5(2) = Y_{56+s}^5(2) = Y_{84+s}^5(2) = Y_{112+s}^5(2) = \delta_8(5, 6, 7), \ s \in S, \\
Y_{140+s}^5(2) = Y_{168+s}^5(2) = \delta_8(5, 3, 6), \ s \in S^c, \\
Y_{140+s}^5(2) = Y_{168+s}^5(2) = \delta_8(5, 3, 8), \ s \in S, \\
Y_{196+s}^5(2) = Y_{224+s}^5(2) = \delta_8(5, 6, 6), \ s \in S^c, \\
Y_{196+s}^5(2) = Y_{224+s}^5(2) = \delta_8(5, 6, 5), \ s \in S.
\end{cases}
$$

The output sequences stemming from states $x_1^6(0), x_2^6(0), \ldots, x_r^6(0)$ are

$$
\begin{cases}
Y_s^6(1) = \delta_8(6, 6), \ s \in S^c, \\
Y_s^6(1) = \delta_8(\overline{6, 3}), \ s \in S, \\
Y_{28+s}^6(2) = \overline{Y_{56+s}^6}(2) = Y_{84+s}^6(2) = Y_{112+s}^3(2) = \delta_8(6, 6, 6), \ s \in S^c, \\
Y_{28+s}^6(2) = Y_{56+s}^6(2) = Y_{84+s}^6(2) = Y_{112+s}^3(2) = \delta_8(6, 6, 7), \ s \in S, \\
Y_{140+s}^6(2) = Y_{168+s}^6(2) = \delta_8(6, 3, 6), \ s \in S^c, \\
Y_{140+s}^6(2) = Y_{168+s}^6(2) = \delta_8(6, 3, 8), \ s \in S, \\
Y_{196+s}^6(2) = Y_{224+s}^6(2) = \delta_8(6, 6, 6), \ s \in S^c, \\
Y_{196+s}^6(2) = Y_{224+s}^6(2) = \delta_8(6, 6, 5), \ s \in S.
\end{cases}
$$

The output sequences stemming from states $x_1^7(0), x_2^7(0), \ldots, x_r^7(0)$ are

$$
\begin{cases}
Y_s^7(1) = \delta_8(7, 6), \ s \in S^c, \\
Y_s^7(1) = \delta_8(7, 3), \ s \in S, \\
Y_{28+s}^7(2) = Y_{56+s}^7(2) = Y_{84+s}^7(2) = Y_{112+s}^7(2) = \delta_8(7, 6, 6), \ s \in S^c, \\
Y_{28+s}^7(2) = Y_{56+s}^7(2) = Y_{84+s}^7(2) = Y_{112+s}^7(2) = \delta_8(7, 6, 7), \ s \in S, \\
Y_{140+s}^7(2) = Y_{168+s}^7(2) = \delta_8(7, 3, 6), \ s \in S^c, \\
Y_{140+s}^7(2) = Y_{168+s}^7(2) = \delta_8(7, 3, 8), \ s \in S, \\
Y_{196+s}^7(2) = Y_{224+s}^7(2) = \delta_8(7, 6, 6), \ s \in S^c, \\
Y_{196+s}^7(2) = Y_{224+s}^7(2) = \delta_8(7, 6, 5), \ s \in S.
\end{cases}
$$

The output sequences stemming from states $x_1^8(0), x_2^8(0), \ldots, x_r^8(0)$ are

$$
\begin{cases}
Y_s^8(1) = \delta_8(6, 6), \ s \in S^c, \\
Y_s^8(1) = \delta_8(6, 7), \ s \in S, \\
Y_{28+s}^8(2) = Y_{56+s}^8(2) = Y_{84+s}^8(2) = Y_{112+s}^8(2) \\
\qquad = Y_{196+s}^8(2) = Y_{224+s}^8(2) = \delta_8(6, 6, 6), \ s \in S^c, \\
Y_{28+s}^8(2) = Y_{56+s}^8(2) = Y_{84+s}^8(2) = Y_{112+s}^8(2) \\
\qquad = Y_{196+s}^8(2) = Y_{224+s}^8(2) = \delta_8(6, 6, 7), \ s \in S, \\
Y_{140+s}^8(2) = Y_{168+s}^8(2) = \delta_8(6, 7, 6), \ s \in S^c, \\
Y_{140+s}^8(2) = Y_{168+s}^8(2) = \delta_8(6, 7, 3), \ s \in S.
\end{cases}
$$

REFERENCES

[1] T. AKUTSU, S. MIYANO, AND S. KUHARA, *Algorithms for identifying Boolean networks and related biological networks based on matric multiplication and fingerprint function*, Journal of Computational Biology, 7 (2000), pp. 331–343.

[2] H. CHEN AND J. LIANG, *Local synchronization fo interconnected Boolean networks with stochastic disturbances*, IEEE Transactions on Neural Networks and Learning Systems, 31 (2020), pp. 452–463.

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

[4] D. Cheng and H. Qi, *Controllability and observability of Boolean control networks*, Automatica, 45 (2009), pp. 1659–1667.

[5] D. Cheng and H. Qi, *A linear representation of dynamics of Boolean networks*, IEEE Transactions on Automatic Control, 55 (2010), pp. 2251–2258.

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

[7] D. Cheng, H. Qi, and Z. Li, *Model construction of Boolean networks via observed data*, IEEE Transactions on Neural Networks, 22 (2011), pp. 525–536.

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

[9] D. Cheng, H. Qi, and Y. Zhao, *An Introduction to Semi-tensor Product of Matrices and its Applications*, World Scientific, 2012.

[10] D. Cheng and Y. Zhao, *Identification of Boolean control networks*, Automatica, 47 (2011), pp. 702–710.

[11] A. Datta, A. Choudhary, M. Bittner, and E. Dougherty, *External control in Markovian genetic regulatory networks: the imperfect information case*, Bioinformatics, 20 (2004), pp. 924–930.

[12] E. Fornasini and M. E. Valcher, *Observability, reconstructibility and state observers of Boolean control networks*, IEEE Transactions on Automatic Control, 58 (2013), pp. 1390–1401.

[13] Y. Guo, *Observability of Boolean control networks*, in Proccedings of the 37th Chinese Control Conference, Dalian,, 2017, pp. 7743–7748.

[14] Y. Guo, *Observability of Boolean control networks using paralled extension and set reachability*, IEEE Transactions on Neural Networks and Learning Systems, 29 (2018), pp. 6402–6408.

[15] S. Huang and D. Ingber, *Shape-dependent control of cell growth, differentiation, and apoptosis: Switching between attractors in cell regulatory networks*, Experimental Cell Research, 261 (2000), pp. 91–103.

[16] S. A. Kauffman, *Metabolic stability and epigenesis in randomly construcuted genetic net*, Journal of Theoretical Biology, 22 (1969), pp. 437–467.

[17] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, New York, 1993.

[18] D. Laschov, M. Margaliot, and G. Even, *Observability of Boolean networks: a graphtheoretic approach*, Automatica, 49 (2013), pp. 2351–2362.

[19] F. Li and J. Sun, *Stability and stabilization of Boolean networks with impulsive effects*, Systems & Control Letters, 61 (2012), pp. 1–5.

[20] H. Li and X. Ding, *A control Lyapunov function approach to feedback stabilization of logical control networks*, SIAM Journal on Control and Optimization, 57 (2019), pp. 810–831.

[21] R. Li and T. Chu, *Complete synchronization of Boolean networks*, IEEE Transactions on Neural Networks and Learning Systems, 23 (2012), pp. 840–846.

[22] S. Liang, S. Fuhrman, and R. Somogyi, *Reveal, a general reverse engineering algorithm for inference of genetic network architectures*, in Proccedings of Pacific Symposium on Biocomputing, vol. 3, 1998, pp. 18–19.

[23] J. Lu, L. Sun, Y. Liu, D. W. C. Ho, and J. Cao, *Stabilization of Boolean control networks*, SIAM Journal on Control and Optimization, 56 (2018), pp. 4385–4404.

[24] M. Meng, J. Lam, J. Feng, and K. C. Cheung, *Stability and guaranteed cost analysis of time-triggered Boolean networks*, IEEE Transactions on Neural Networks and Learning Systems, 29 (2018), pp. 3893–3899.

[25] D. Nam, S. Seo, and S. Kim, *An efficient top-down search algorithm for learning Boolean networks of gene expression*, Machine Learning, 65 (2006), pp. 229–245.

[26] R. Pal, I. Ivanov, A. Datta, M. Bittner, and E. Dougherty, *Generating Boolean networks with a prescribed attractor structure*, Bioinformatics, 21 (2005), pp. 4021–4025.

[27] A. Veliz-Cuba and B. Stigler, *Boolean models can explain bistability in the lac operon*, Journal of Computational Biology, 18 (2011), pp. 783–794.

[28] B. Wang, J. Feng, H. Li, and Y. Yu, *On detectability of Boolean control networks*, Nonlinear Analysis: Hybrid Systems, 36 (2020), p. 100859.

[29] Y. Wu, X. Sun, X. Zhao, and T. Shen, *Optimal control of Boolean control networks with average cost: A policy iteration approach*, Automatica, 100 (2018), pp. 378–387.

[30] Y. Yu, J. Feng, J. Pan, and D. Cheng, *Block decoupling of Boolean control networks*, IEEE Transactions on Automatic Control, 64 (2019), pp. 3129–3140.

[31] K. Zhang and L. Zhang, *Observability of Boolean control networks: a unified approach ap-*

proach based on finite automata, IEEE Transactions on Automatic Control, 61 (2016), pp. 2733–2738.

[32] Y. ZHAO, H. QI, AND D. CHENG, *Input-state incidence matrix of Boolean control networks and its application*, Systems & Control Letters, 59 (2010), pp. 767–774.

[33] J. ZHONG, D. HO, J. LU, AND Q. JIAO, *Pinning controllers for activation output tracking of Boolean networks under one-bit perturbation*, IEEE Transactions on Cybernetics, 49 (2019), pp. 3398–3408.

[34] Q. ZHU, Y. LIU, J. LU, AND J. CAO, *Further results on the controllability of Boolean control networks*, IEEE Transactions on Automatic Control, 64 (2019), pp. 440–442.

[35] Y. ZOU AND J. ZHU, *Cycles of periodically time-variant Boolean networks*, Automatica, 51 (2015), pp. 175–179.