Online Observability of Boolean Control Networks*

Guisen Wu[1] and Liyun Dai[2]

***Abstract*— Four types of observability of Boolean control networks (BCNs) have been proposed to solve different problems, but all of the four types of observability of BCNs is offline observability that we can't adjust the input sequence by observing the output sequence, when we determining the initial state of BCNs. So we proposed online observability that we can determine the initial state of BCNs by deciding input sequence and observing out sequence step by step in this paper. We define the concepts of deduce function and K steps deterministic for online observability of BCNs. And then, define the super tree and directed graph to determine online observability of BCNs. Finally, we introduce the applications of online observability of BCNs and talk about some future works.**

***Index Terms*— Boolean control networks, online observability, supertree, directed graph, find shortest path, avoid entering critical states.**

I. INTRODUCTION

In 1960s, Jacob and Monod found that "Any cell contains a number of 'regulatory' genes that act as switches and can turn one another on and off. If genes can turn one another on and off, then you can have genetic circuits." Inspired by these Boolean-type actions in genetic circuits, the Boolean networks (**BNs**) was firstly proposed by Kauffman for modeling nonlinear and complex biological systems. Some general descriptions of the **BNs** and its applications to biological systems can be found in Kauffman. Since then research interestes in **BNs** have been motivated by the large number of natural and artificial systems whose describing variables display only two distinct configurations, and hence take only two values.

When extenal regualtion or perturbation is considered, **BNs** are naturally extended to Boolean control networks (**BCNs**). The study on control-theoretic problems of **BCNs** can date back to 2007 TL: should be much much earlier :TL in which the problem of determining the controllability of **BCNs** is proved **NP**-hard in the number of nodes, and point out that "One of the major goals of systems biology is to develop a control theory for complex biological systems." Since then, the study on control-theoretic problems in the areas of **BNs** and **BCNs** has drawn great attention. Controllability and observability are basic control-theoretic problems. In 2009 Cheng et al. have developed an algebraic framework to deal with both **BNs** and **BCNs** by using new tool, called *semi-tensor product* (STP) of matrices proposed in 2001, and give equivalent conditions for controllability of **BCNs** and observability of controllable **BCNs**. This type of observability means that every initial state can be determined by an input sequence. Important results on other types of observability of **BCNs** came up hereafter. To date, there are four types of observability have been proposed.

1) This type of observability means that every initial state can be determined by an input sequence.
2) Another observability, proposed in 2010, stands for that for every two distinct initial states, there exists an input sequence which can distinguish them.
3) A third observability stating that there is an input sequence that determines the initial state, is proposed to study the identifiability problem of **BCNs**.
4) A fourth observability is determined in 2013, which is essentially the observability of linear control systems, i.e., every sufficient long input sequence can determine the initial state.

TL: can you state the four types observability clearly and formally here? :TL

But all the four types of observability of **BCNs** is offline observability that you can't adjust the input sequence by observing the output sequence when you determining the initial state of **BCNs**. So we proposed the online observability that we can determine the initial state of **BCNs** by deciding input sequence and observing out sequence step by step. To this end, we firstly define the deduce function and K steps deterministic for **BCNs**, which is the basis of definition of online observability of **BCNs**. Secondly, we use the supertree and directed graph to determine the online observability of **BCNs**, which are built by the definition of online observability. Finally, we use the online observability of **BCNs** to find the shortes path and avoid entering critical states when we determining the initial state of **BCNs**.

The remainder of this paper is organized as follows. **Section II** introduces necessary preliminaries about **BCNs**, algebraic forms of **BCNs** and the online observability of **BCNs**. **Section III** presents how to determine the online observability of **BCNs** by super tree and directed graph. **Section IV** introduces some applications of the online observability of **BCNs**. **Section V** end up with the introduction of some future works.

TL: I will try to rewrite the intro. :TL

II. PRELIMINARIES

In this section we introduce the **BCNs** and its algebraic form, then define the online observability of **BCNs**.

*This work was not supported by any organization

[1] Albert Author is with Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, 7500 AE Enschede, The Netherlands albert.author@papercept.net

[2] Bernard D. Researcheris with the Department of Electrical Engineering, Wright State University, Dayton, OH 45435, USA b.d.researcher@ieee.org

A. Boolean Control Networks

A Boolean control network can be described as a network graph TL: network graph? is this from the literature? :TL which is a directed graph, and the logical equations to describe the updating rules of the nodes in the network graph.

Definition 1: A **BCN** consists of input nodes, state nodes, output nodes, and directed edges which connect nodes. A node in **BCN** can take a logic value from $\{0, 1\}$ at a discrete time 0, 1, 2,.... For one directed edge from node v_i to node v_j means that the logic value of v_j at time step $t+1$ is affected by the value of v_i at time step t.

Note that one can only know whether or not a node is affected by another node from the network graph. Different **BCNs** may have the same structure, in order to determine a **BCN** uniquely, logical equations are also needed to describe the specific updating rules of **BCN**.

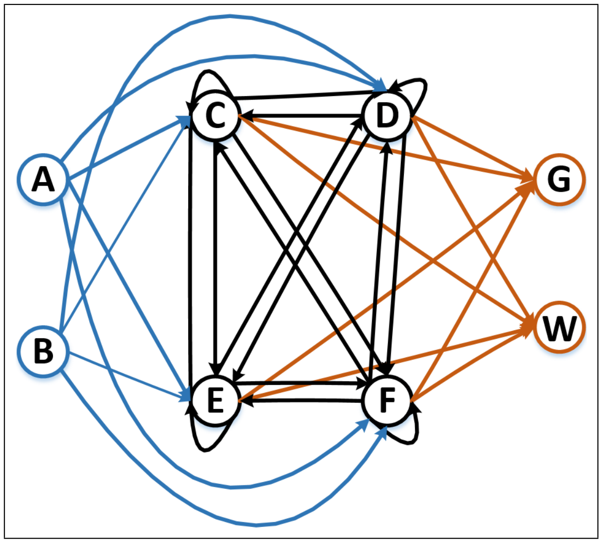


Fig. 1. A Boolean control network with two input nodes A and B, four state nodes C, D, E and F, and two output nodes G, W. We use blue, black and orange, to distinguish three kinds of nodes and three kinds of edges.

We give a simple example to describe **BCN**.

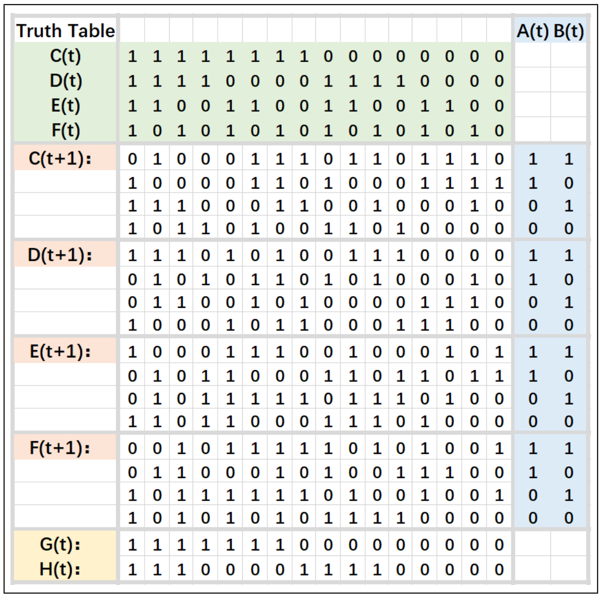
Example 1: In **Fig.1** we have a Boolean control network with two input nodes A and B, four state nodes C, D, E and F, and two output nodes G, W. The dynamics of the **BCN** shown in **Fig.1** is described as truth table (**Fig.2**). For convenience, we will use this example in the whole paper to explain various concepts we introduce.

B. The Algebraic Forms of BCNs

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

$$\begin{aligned} s(t+1) =& f(i(t), s(t)) \\ o(t) =& h(s(t)) \end{aligned} \tag{1}$$

where D : the set $\{0, 1\}$; $t = 0, 1, ...$ represents the discrete time; $s \in D^n$; $i \in D^m$; $o \in D^q$; $f : D^{n+m} \mapsto D^n$ and $h : D^n \mapsto D^q$ are logical functions. So in the previously mentioned example, we have $C(t), D(t), E(t), F(t) \in s(t)$; $A(t), B(t) \in i(t)$ and $G(t), W(t) \in o(t)$; $n = 4$, $m = 2$ and $q = 2$; f and h are described in the truth table (**Fig.2**). Then using the *semi-tensor product* (STP) of matrices to define the algebraic form of **BCNs**.



Truth Table																	A(t)	B(t)
C(t)	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0		
D(t)	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0		
E(t)	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0		
F(t)	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0		
C(t+1):	0	1	0	0	0	1	1	1	0	1	1	0	1	1	1	0	1	1
	1	0	0	0	0	1	1	0	1	0	0	0	1	1	1	1	1	0
	1	1	1	0	0	0	1	1	0	0	1	0	0	0	1	0	0	1
	1	0	1	1	0	1	0	0	1	1	0	1	0	0	0	0	0	0
D(t+1):	1	1	1	0	1	0	1	0	0	1	1	1	0	0	0	0	1	1
	0	1	0	1	0	1	1	0	1	0	1	0	0	0	1	0	1	0
	0	1	1	0	0	1	0	1	0	0	0	0	1	1	1	0	0	1
	1	0	0	0	1	0	1	1	0	0	0	1	1	1	0	0	0	0
E(t+1):	1	0	0	0	1	1	1	0	0	1	0	0	0	1	0	1	1	1
	0	1	0	1	1	0	0	0	1	1	0	1	1	0	1	1	1	0
	0	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	0	1
	1	1	0	1	1	0	0	0	1	1	1	0	1	0	0	0	0	0
F(t+1):	0	0	1	0	1	1	1	1	1	0	1	0	1	0	0	1	1	1
	0	1	1	0	0	0	1	0	1	0	0	1	1	1	0	0	1	0
	1	0	1	1	1	1	1	1	0	1	0	0	1	0	0	1	0	1
	1	0	1	0	1	0	1	0	1	1	1	1	0	0	0	0	0	0
G(t):	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0		
H(t):	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0		

Fig. 2. The truth table which describe the updating rules of the **BCN** shown in **Fig.1**.

Definition 2 (STP): Let $A \in \mathbb{R}_{m \times n}$, $B \in \mathbb{R}_{p \times q}$ and $\alpha = lcm(n, p)$ be the least common multiple of n and p. The STP of A and B is defined as $A \ltimes B = (A \otimes I_{\alpha/n})(B \otimes I_{\alpha/p})$, where $\otimes$ denotes the Kronecker product.

Since STP keeps most properties of the conventional product, the associative law, the distributive law, etc., we usually omit the symbol "$\ltimes$" hereinafter. Before introducing algebraic form, we introduce some related notations at first. δ_n^i: the i-th column of the identity matrix I_n; Δ_n: the set $\{\delta_n^1, ..., \delta_n^n\}$; $\delta_n\,[i_1, ..., i_s]$: the logical matrix $\left[\delta_n^{i_1}, ..., \delta_n^{i_s}\right]$ $(i_1, ..., i_s \in \{1, 2, ..., n\})$; $L_{n \times s}$: the set of $n \times s$ logical matrices. So (1) can be quivalently represented in the following algebraic form:

$$\begin{aligned} s(t+1) =& Li(t)s(t) \\ o(t) =& Hs(t) \end{aligned} \tag{2}$$

where $s \in \Delta_N$, $i \in \Delta_M$, and $o \in \Delta_Q$ denote the states, inputs and outputs, respectively; $L \in L_{N \times (NM)}$; $H \in L_{Q \times N}$; and $N := 2^n$, $M := 2^m$, and $Q := 2^q$.

The **BCN** whose structure is depicted in **Fig.1** and updating rules is described in **Fig.2** can be represented with the algebraic form:

$$\begin{aligned} s(t+1) =& \delta_{16}[\alpha]i(t)s(t) \\ o(t) =& \delta_4[\beta]s(t) \end{aligned} \tag{3}$$

where $\alpha = \{10, 4, 11, 16, 9, 5, 1, 7, 15, 2, 3, 12, 7, 6, 8, 13, 8, 9, 15, 10, 14, 4, 3, 16, 1, 14, 12, 13, 5, 7, 2, 6, 7, 2, 3, 13, 13, 9, 5, 1, 16, 13, 6, 14, 11, 10, 4, 15, 1, 14, 7, 6, 9, 8, 11, 12, 5, 5, 13, 3, 10, 12, 16, 16\}$ and $\beta = \{1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4\}$; $t \in \mathbb{N}$; $s \in \Delta_{16}$; $i \in \Delta_4$; $o \in \Delta_4$.

III. The Online Observability of BCNs

In this paper we propose the online observability:

Definition 3:

A **BCN** is called online observable, if for every initial state $x_0 \in \Delta_N$, there exists a way TL: be precise! :TL that the initial state x_0 can be determined by the deciding input sequence and observing output sequence step by step.

TL: maybe I did not understand this, but I think you are confusing two things: the observability and the algorithm (approach) to determine the initial state. It seems to me that you are describing a new approach (the online approach), but does this change the observability? if yes, how? Is this a stronger notion or a weaker notion or incomparable? :TL

Different from existed four types, the observability we proposed can determine the initial state online that every input in the input sequence is decided step by step as we observe the output sequence. At the beginning, we can observe the output of **BCNs**, so we can infer the possible values of state nodes and treat them as possible states set. Then as we can know the possible states set, we need to decide the input (i_0) that it will make sure any different possible states ($s_i, s_j \in \Delta_N$) will not turn into the same state ($Ls_i i_0 = Ls_j i_0$) after affected by input. After decided input, we can observe the new outputs, and then we can infer the new possible states set, the cardinality of possible states set after input will not lager than the cardinality of possible states set before input. If the cardinality of possible states set turn into be 1, we can determine the initial state of BCN. The mathematical definition of online observability is given below:

First, we define the deduce function which simulate the deduction process :

$$D\left(S, I, O\right)$$

where $S \in 2^{\Delta_N}$ is the possible states set before deduction; $I \in \Delta_M$ represents the input; $O \in \Delta_Q$ represents the output; $D\left(S, I, O\right)$ is the possible states set after deduction. Based on deduction process, we have for any $s_i(t+1) \in D\left(S, I, O\right)$, there exists $s_i \in S$ that $s_i(t+1) = LIs_i(t)$ and $O = Hs_i(t+1)$.

And then we have:

$$D\left(\varnothing, I_i, O_i\right) = D\left(\varnothing, \varepsilon, O_i\right) = D\left(\varnothing, \varepsilon, \varepsilon\right) = \varnothing \quad (4)$$

$$D\left(S_i, \varepsilon, \varepsilon\right) = S_i \quad (5)$$

$$D\left(\Delta_N, \varepsilon, \delta_4^1\right) = \{\delta_{16}^1, \delta_{16}^2, \delta_{16}^3\} \quad (6)$$

$$D\left(\{\delta_{16}^1, \delta_{16}^2, \delta_{16}^3\}, \delta_4^1, \varepsilon\right) = \{\delta_{16}^{10}, \delta_{16}^4, \delta_{16}^{11}\} \quad (7)$$

$$D\left(\{\delta_{16}^1, \delta_{16}^2, \delta_{16}^3\}, \delta_4^1, \delta_4^3\right) = \{\delta_{16}^{10}, \delta_{16}^{11}\} \quad (8)$$

$$D\left(\{\delta_{16}^4, \delta_{16}^5, \delta_{16}^6\}, \delta_4^3, \varepsilon\right) = \{\delta_{16}^9, \delta_{16}^{13}\} \quad (9)$$

(4) represents that if the possible states set is $\varnothing$, no matter what you do you can only deduce $\varnothing$; if the possible states set is $\S_i$ and we don't input anything and don't observe the output, we can only deduce that the possible states set is $\S_i$ which shown in (5); when the possible states set is Δ_N we observe that the outputs is δ_4^1 we can deduce that the possible states would be δ_{16}^1, δ_{16}^2 or δ_{16}^3 which shown in (6) ; and then we input δ_4^1, before observe the output we can only deduce the possible states would be δ_{16}^{10}, δ_{16}^4 or δ_{16}^{11} which shown in (7); but if we observe that the output is δ_4^3, we can deduce that the possible state values can be δ_{16}^{10} or δ_{16}^{11} which shown in (8); then if the set of states is $\{\delta_{16}^4, \delta_{16}^5, \delta_{16}^6\}$ and the inputs is δ_4^3, before observe we can infer that the possible state values can be δ_{16}^9 or δ_{16}^{19} shown in (9), as you can see the number of the possible state decreased, so we may can't deduce the initial state any more.

After difined the deduce function, we can introduce the mathematical definition of the observability of **BCNs**. It may easier to be difined by programming language recursively, but we can also define its mathematical form by natural induction. Before define the observability of **BCNs**, we need to difine the K ($K \geq 0$) **steps deterministic** of the states set of **BCNs**:

When $K = 0$:
If for a set of states S_i and $|S_i| = 1$, then S_i is 0 step deterministic. It means that we can determine the state without any input and observing output if we have the cardinality of possible states set is 1.

When $K > 0$:
If for a set of states S_i and $|S_i| \geq 1$, $\exists I_i \in \Delta_M$ implies $|D\left(S_i, I_i, \varepsilon\right)| = |S_i|$, and implies "$\forall O_i \in \Delta_Q$, $|D\left(S_i, I_i, O_i\right)| \neq 0$ implies $D\left(S_i, I_i, O_i\right)$ is K_i ($K_i < K$) stepes deterministic", then S_i is K ($K > 0$) steps deterministic.

From the definition of deterministic, if S_i is K_1 steps deterministic and $K_1 \leq K_2$, then S_i is K_2 steps deterministic. But if S_i is K_1 steps deterministic and $K_1 \geq K_2$, we can not make sure that S_i is K_2 steps deterministic. So you can consider the mean of "S_i is K_i steps deterministic" as "whether we can determine the state of a **BCN** with possible states set S_i by deciding input sequence and observing out sequence step by step in K_i steps".

Definition 4 (Online Observability of BCNs): If $\forall O_i \in \Delta_Q$ and $|D\left(\Delta_N, \varepsilon, O_i\right)| \neq 0$, $\exists K_i \geq 0$ implies $D\left(\Delta_N, \varepsilon, O_i\right)$ is K_i stepes deterministic, then this **BCN** is online observable. Even simpler, if $\exists K_i \geq 0$ implies Δ_N is K_i stepes deterministic, then this **BCN** is online observable. The difference is that whether we observe the initial output of the **BCN** at first, for better performance the former definition would be better.

IV. Determining the Observability of BCNs

In this paper we proposed two ways to determine the observability of **BCNs**, the first one is by supertree, the second one is by directed graph. The construction process of supertree simulates deduction process mentioned before.

Then we check the tree based on the definition of online observability of **BCNs** depth first or breadth first. When we used the super tree to determine the observability of **BCNs**, we need to check the existence of loops, and many nodes in the tree are repeated, which will take a lot of time overhead and space overhead. So we proposed the second way which determine the online observability by directed graph, in this way we can avoid checking the existence of loop and avoid checking repeated nodes. There are also other advantages like determining observability earlier and select the input smarter, which will reduce time and space overhead.

A. Supertree

As we mentioned before when we want to determine the initial state of **BCNs**, we can use the deduce function. So the root node of the supertree should be the Δ_N, and then according to the definition of online observability we will alternately observing the output and deciding the input. So the son nodes of S_i will be $D\left(\Delta_N, \varepsilon, O_i\right)$ and $D\left(\Delta_N, I_i, \varepsilon\right)$ alternately. Finally, when the cardinal number of the states set is 1 we can determine the initial state, so the leave nodes should be the nodes which containt only one state.

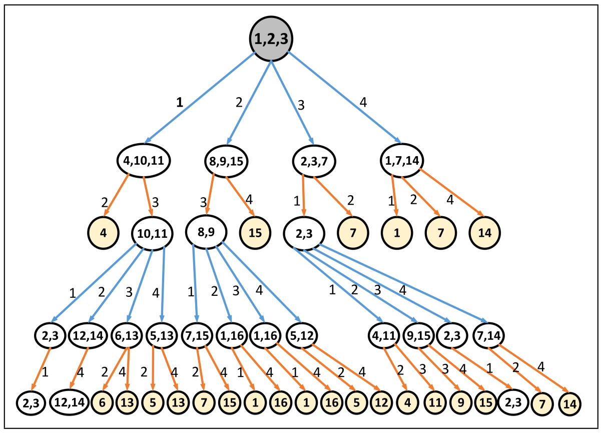


Fig. 3. Branch of the tree which represents $\{\delta_{16}^1, \delta_{16}^2, \delta_{16}^3\}$. The blue edges and orange edges show the observing output processes and deciding input processes respectively; the yellow nodes are leaf nodes.

The **Fig.3** show branch of the tree which represents $\{\delta_{16}^1, \delta_{16}^2, \delta_{16}^3\}$, the nodes represent the states sets, the blue edges represent the observing output processes, and the orange edges represent the deciding input processes. And only the yellow nodes are the leave nodes, so you can see that this branch is not completed. If we want to find all of the ways to determine the initial state, we have to build the complete tree, it will takes many time and space. Especially when there are loops in the tree like the $\{\delta_{16}^2, \delta_{16}^3\}$ in fourth layer and the $\{\delta_{16}^2, \delta_{16}^3\}$ in fifth layer that will form a loop. In this case you can never build the complete tree, so you need to check the existence of loops and omit it. There are also some nodes take the same states set, like that there two nodes which take $\{\delta_{16}^1, \delta_{16}^{16}\}$ in the fifth layer. But if you only need to find a ways to determine the initial state, when you find the leaf nodes δ_{16}^1, δ_{16}^7 and δ_{16}^{14} in third layer by breadth-first algorithm, you can make sure that the states set $\{\delta_{16}^1, \delta_{16}^2, \delta_{16}^3\}$ is 1 step deterministic.

B. Directed Graph

To improve the shortcomings of the way by supertree, we proposed derected graph wich will takes less time and space overhead. The most difference between supertree and derected graph is that supertree is built from the root node to leaf nodes, but the derected graph is built from smaller nodes (contain less states) to larger nodes (contain more states). The construction process of derected graph:

Step1: Use a variable k ($k \geq 0$) to represent the number of states contained in the nodes, and initialize k to 0.

Step2: Let k be $(k+1)$, then try to build all the nodes with k states whose corresponding outputs are the same, and classify them by their corresponding outputs. If we can't build any node with k states whose corresponding output are the same, stop building and we can make sure this **BCN** is online observable. For instance in **Fig.4**, the nodes in the second last layer is the nodes with 2 states, and the figure only show the nodes whose corresponding outputs is δ_4^1. For better performce, we sort the states inside the nodes from small to large, and then sort the nodes based on the values inside the nodes, like the nodes $\{\delta_{16}^1, \delta_{16}^2\}$, $\{\delta_{16}^1, \delta_{16}^3\}$ and $\{\delta_{16}^2, \delta_{16}^3\}$ shown in **Fig.4**. If $k = 1$ we go to **Step2** after we built the nodes with 1 state; else if $k = 2$ we go to **Step3** to find suitable inputs set for the nodes with 2 states; else if $k \geq 3$ we go to **Step4** to find suitable inputs set for the nodes with more than 2 states.

Step3: We make Δ_M as the suitable inputs set for every node which is built in the previous step. And then go to **Step2** to build edges.

Step4: We use two nodes with $(k-1)$ states and a node with two states to find the suitable inputs set for every node which is built in the previous step. Like that, we can search inputs sets which make $\{\delta_{16}^4, \delta_{16}^5, \delta_{16}^6\}$, $\{\delta_{16}^5, \delta_{16}^6, \delta_{16}^7\}$ and $\{\delta_{16}^4, \delta_{16}^7\}$ Z ($Z \geq 1$) steps deterministic, and then take the intersection of these sets to be the suitable inputs set of $\{\delta_{16}^4, \delta_{16}^5, \delta_{16}^6, \delta_{16}^7\}$. And then go to **Step5** to build edges.

Step5: According to the order determined in **Step2**, we input all suitable inputs for every node one by one. If for one input I_i and states set of a node S_i implies $\left|D\left(S_i, I_i, \varepsilon\right)\right| < \left|S_i\right|$, we can make sure the I_i is a wrong input; else if for all $O_i \in \Delta_Q$ and $\left|D\left(S_i, I_i, O_i\right)\right| > 0$, the $D\left(S_i, I_i, O_i\right)$ is Z ($Z \geq 0$) steps deterministic then I_i is a right input, then connect the node S_i to all nodes $D\left(S_i, I_i, O_i\right)$ with directed edges, the colour of directed edges represent the corresponding input; else if there exist $O_i \in \Delta_Q$ and we can not make sure whether $D\left(S_i, I_i, O_i\right)$ is Z steps deterministic, we check it in the next round; and then if the input is the last suitable input of this node we begin to build edges for next node, else input the next suitable input. If try to build edges for all nodes in one whole round, and no more new edge has been built, we stop building edge. If there exist one node without any edge connect it with other nodes, we can make sure that this node is not Z steps deterministic, so this **BCN** is not online observable, then we stop building nodes, else we go to **Step2**.

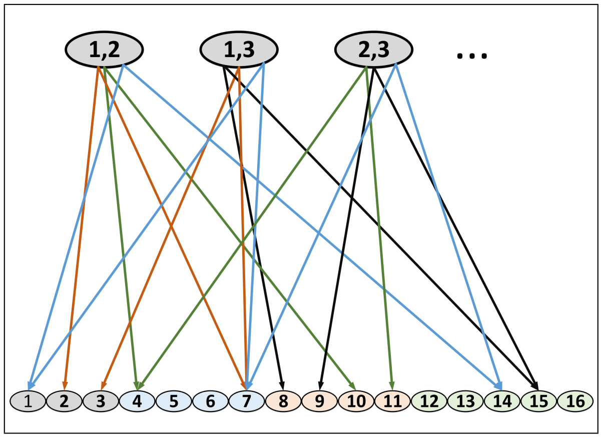


Fig. 4. Part of the directed graph which represents $\{\delta_{16}^1, \delta_{16}^2\}$, $\{\delta_{16}^1, \delta_{16}^3\}$ and $\{\delta_{16}^2, \delta_{16}^3\}$. The green, black, orange, blue edges show the inputs δ_4^1, δ_4^2, δ_4^3 and δ_4^4 respectively.

V. Applications

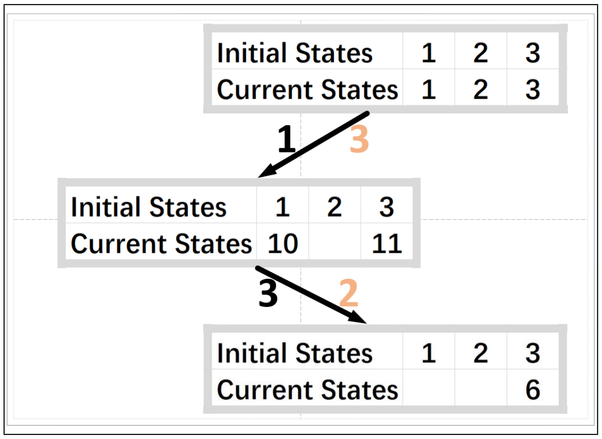
If the **BCN** we research is online observable, and we have built the directed graph for it, then we can use it to determine the initial state of **BCN**. And we can also use it to try to find the shortest path or avoid entering critical states when we determine the initial state of **BCN**. Because the output we observe is undeterminable, we use expected value and variance of the length of path and the times of entering critical states to help us to chose the input.

A. Determine the Initial State

After we build the directed graph of the **BCN**, we can use it to determine the initial state of **BCNs** as **Fig.5** shows. First, we observe the output of the **BCN** mentioned before, if we observe δ_4^1 that we can infer that the possible states set can be $\{\delta_{16}^1, \delta_{16}^2, \delta_{16}^3\}$, record them as initial states and current states in the table. And then, we input δ_4^1 and observe δ_4^3, we can infer that the possible states set can be $\{\delta_{16}^{10}, \delta_{16}^{11}\}$, record them as current states set in the corresponding position. Input and output again and again untill we can infer only one possible state, in that time we can determine the current state and the corresponding initial state of the **BCN**.

B. Find Shortest Path

When we need to determine the initial state of a **BCN**, an important aspect that we will consider is to find the shortest path. Maybe we can't find the shortest path definitely, but we can use the directed graph to make the best decision. We introduce two functions $SPE(S_i, I_i)$ and $SPV(S_i, I_i)$ to describe the shortest path expected value and shortest path variance, S_i is the possible states set and I_i is the input we chose, the definition of $SPE(S_i, I_i)$ is as follows:



Initial States	1	2	3
Current States	1	2	3

Initial States	1	2	3
Current States	10		11

Initial States	1	2	3
Current States			6

Fig. 5. The process of determing the initial state of BCNs, we change the values of current states by input and the output we observe.

When $|S_i| = 1$:
For all $I_i \in \Delta_M$, we have $SPE(S_i, I_i) = 0$, then the least shortest path expected value $LLCE(S_i) = 0$.
When $|S_i| > 1$:
$\{I_1, I_2, ...\}$: the right inputs set of S_i; $\forall I_i \in \{I_1, I_2, ...\}$, and $\{S_i^1, S_i^2, ...\}$ is corresponding to all possible output O_i after input I_i; $SPE(S_i, I_i) = 1 + ((LSPE(S_i^1)|S_i^1| + LSPE(S_i^2)|S_i^2| + ...)\,/\,|S_i|)$; $LSPE(S_i) = Min(SPE(S_i, I_1), SPE(S_i, I_2), ...)$.

C. Avoid Entering Critical States

Another important aspect that we will consider is to avoid entering critical states. We can also construct two functions $LCE(S_i, I_i)$ and $LCV(S_i, I_i)$ to describe least expected value and variance of the times of entering critical states, the definition of $LCE(S_i, I_i)$ is as follows:

CS: the critical states set of the **BCN** we research.
When $|S_i| = 1$:
For all $I_i \in \Delta_M$, we have $LCE(S_i, I_i) = |S_i \cap CS|$, then the least shortest path expected value $LLCE(S_i) = |S_i \cap CS|$.

When $|S_i| > 1$:
$\{I_1, I_2, ...\}$: the right inputs set of S_i; $\forall I_i \in \{I_1, I_2, ...\}$, and $\{S_i^1, S_i^2, ...\}$ is corresponding to all possible output O_i after input I_i; $LCE(S_i, I_i) = (|S_i \cap CS| + (LLCE(S_i^1)|S_i^1| + LLCE(S_i^2)|S_i^2| + ...)\,/\,|S_i|)$; $LLCE(S_i) = Min(LCE(S_i, I_1), LCE(S_i, I_2), ...)$.

Then we can get the definitions of $SPV(S_i, I_i)$ and $LCV(S_i, I_i)$ in similar ways, and use them to make the best decision we like.

VI. Conclusions

In this paper, we proposed online observability of **BCNs** and define its mathematical form; then we use the super tree and directed graph to determine the online observability; after determined the online observability we use it to find the shortest path and avoid entering critical states when we determining the initial state of **BCNs**.

But even we use the directed graph, it is still hard to determine the the online observability of **BCNs** which with a large number of nodes. So, in the future we will try to separate the **BCNs**, and then determine their online observability; and try to use some knowledge about formal methods for better scalable of **BCNs**. In addition to the theoretical aspects, realistic application is also very important, we will also try to find some realistic example which can be modeled by **BCNs**, then determine their online observability for better performance.

APPENDIX

Appendixes should appear before the acknowledgment.

ACKNOWLEDGMENT

The preferred spelling of the word acknowledgment in America is without an e after the g. Avoid the stilted expression, One of us (R. B. G.) thanks . . . Instead, try R. B. G. thanks. Put sponsor acknowledgments in the unnumbered footnote on the first page.

References are important to the reader; therefore, each citation must be complete and correct. If at all possible, references should be commonly available publications.

REFERENCES

[1] G. O. Young, Synthetic structure of industrial plastics (Book style with paper title and editor), in Plastics, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 1564.

[2] W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123135.

[3] H. Poor, An Introduction to Signal Detection and Estimation. New York: Springer-Verlag, 1985, ch. 4.

[4] B. Smith, An approach to graphs of linear forms (Unpublished work style), unpublished.

[5] E. H. Miller, A note on reflector arrays (Periodical styleAccepted for publication), IEEE Trans. Antennas Propagat., to be publised.

[6] J. Wang, Fundamentals of erbium-doped fiber amplifiers arrays (Periodical styleSubmitted for publication), IEEE J. Quantum Electron., submitted for publication.

[7] C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.

[8] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces(Translation Journals style), IEEE Transl. J. Magn.Jpn., vol. 2, Aug. 1987, pp. 740741 [Dig. 9th Annu. Conf. Magnetics Japan, 1982, p. 301].

[9] M. Young, The Techincal Writers Handbook. Mill Valley, CA: University Science, 1989.

[10] J. U. Duncombe, Infrared navigationPart I: An assessment of feasibility (Periodical style), IEEE Trans. Electron Devices, vol. ED-11, pp. 3439, Jan. 1959.

[11] S. Chen, B. Mulgrew, and P. M. Grant, A clustering technique for digital communications channel equalization using radial basis function networks, IEEE Trans. Neural Networks, vol. 4, pp. 570578, July 1993.

[12] R. W. Lucky, Automatic equalization for digital communication, Bell Syst. Tech. J., vol. 44, no. 4, pp. 547588, Apr. 1965.

[13] S. P. Bingulac, On the compatibility of adaptive controllers (Published Conference Proceedings style), in Proc. 4th Annu. Allerton Conf. Circuits and Systems Theory, New York, 1994, pp. 816.

[14] G. R. Faulhaber, Design of service systems with priority reservation, in Conf. Rec. 1995 IEEE Int. Conf. Communications, pp. 38.

[15] W. D. Doyle, Magnetization reversal in films with biaxial anisotropy, in 1987 Proc. INTERMAG Conf., pp. 2.2-12.2-6.

[16] G. W. Juette and L. E. Zeffanella, Radio noise currents n short sections on bundle conductors (Presented Conference Paper style), presented at the IEEE Summer power Meeting, Dallas, TX, June 2227, 1990, Paper 90 SM 690-0 PWRS.

[17] J. G. Kreifeldt, An analysis of surface-detected EMG as an amplitude-modulated noise, presented at the 1989 Int. Conf. Medicine and Biological Engineering, Chicago, IL.

[18] J. Williams, Narrow-band analyzer (Thesis or Dissertation style), Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.

[19] N. Kawasaki, Parametric study of thermal and chemical nonequilibrium nozzle flow, M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.

[20] J. P. Wilkinson, Nonlinear resonant circuit devices (Patent style), U.S. Patent 3 624 12, July 16, 1990.