Fault detection analysis of Boolean control networks

Ettore Fornasini and Maria Elena Valcher

***Abstract*—In this paper we address the fault detection problem for Boolean control networks (BCNs). We first investigate completeness and T-completeness of the set of input/output trajectories. Next, we introduce the concept of meaningful fault, and prove necessary and sufficient conditions under which meaningful faults can be detected from the input/output trajectories of the BCN. Two fault detection algorithms are provided.**

I. Introduction

The renewed interest in Boolean control networks (BCNs) can be credited to two major reasons. On the one hand, BCNs have proved to be a convenient modeling tool to capture a number of phenomena whose variables display only two operation levels (on/off, high/low, 1/0...). In particular, BCNs have been employed to describe genetic regulation networks [11], [17]. On the other hand, the algebraic representation developed by D. Cheng and co-authors allows to cast BCNs into the framework of linear state models (operating on canonical vectors) [1], [2], [3]. This set-up opened new perspectives on the solution of many problems for this class of systems. And, indeed, within this setting, stability, stabilizability, controllability [13], observability [5] and optimal control [6], [12], have been successfully investigated.

Research on fault detection originated in the seventies and still represents a lively research area (see [8], [10] for two extended surveys). Fault detection of logic circuits, in particular, has received a lot of attention [9]. Recently, in [15], this problem has been investigated by resorting to the semi-tensor product method. However, the class of logic networks considered in the paper was not described by a BCN and the only faults were "stuck-at faults", resulting in the fact that one (or more) of the input or output variables remains stuck at a certain value. Also, the failure location problem in networks investigated in Chapter 4 of [3] pertains the problem of evaluating whether a route connecting two nodes is active or not. The fault detection problem for gene regulation networks, described by means of Boolean networks, has been investigated to address some biomedical problems. In [14] it is observed that "the study of diseases such as cancer requires the modeling of gene regulations and the loss of control associated with it. The genetic alterations in the system can be modeled using different fault models in the Boolean Network paradigm." Similarly, in [18], the Authors develop a Boolean network to describe the failure of the oxidative stress response.

Aiming to generalize the results obtained in [14], [18] to the broader context of Boolean control networks, in this paper we address the situation when, as a consequence of a fault, a BCN switches from its original model to a different one, thus generating output trajectories that are not compatible with its updating equations. The main question we want to answer is the following one. Assuming that the BCN equations are known, but the state is not accessible, how can we decide whether a fault has occurred, by evaluating the BCN output that corresponds to the applied (known, but otherwise arbitrary) control input?

Ettore Fornasini and Maria Elena Valcher are with the Dipartimento di Ingegneria dell'Informazione, Università di Padova, via Gradenigo 6/B, 35131 Padova, Italy, {fornasini,meme}@dei.unipd.it.

The paper is organized as follows: section II introduces the algebraic state representation of a BCN and investigates completeness and T-completeness properties. Section III introduces the concept of meaningful fault, and provides necessary and sufficient conditions for a meaningful fault to be detected from the input/output trajectories. Finally, in section IV two fault detection algorithms are proposed. A preliminary version of part of the first 3 sections appeared in [7].

Notation. Given $k,n \in \mathbb{Z}_+$, with $k \le n$, the symbol $[k,n]$ denotes the set $\{k,k+1,\dots,n\}$. Boolean vectors and matrices take values in $\mathcal{B} := \{0,1\}$, with the usual operations (sum $+$, product $\cdot$ and negation $\bar{\cdot}$). δ^i_k denotes the ith canonical vector of size k, $\mathcal{L}_k$ the set of k-dimensional canonical vectors, and $\mathcal{L}_{k\times n} \subset \mathcal{B}^{k\times n}$ the set of $k \times n$ matrices whose columns are canonical vectors of size k. Any matrix $L \in \mathcal{L}_{k\times n}$ can be represented as a row whose entries are canonical vectors in $\mathcal{L}_k$, namely $L = [\,\delta^{i_1}_k \quad \delta^{i_2}_k \quad \dots \quad \delta^{i_n}_k\,]$, for suitable indices $i_1, i_2, \dots, i_n \in [1,k]$. The ℓth entry of a vector $\mathbf{v}$ is $[\mathbf{v}]_\ell$.

Given a matrix $L \in \mathcal{B}^{k\times k}$ (in particular, $L \in \mathcal{L}_{k\times k}$), we associate with it a *digraph* $\mathcal{D}(L)$, with vertices $1,\dots,k$. There is an arc (j,ℓ) from j to ℓ if and only if the (ℓ,j)th entry of L is unitary. A sequence $j_1 \to j_2 \to \dots \to j_r \to j_{r+1}$ in $\mathcal{D}(L)$ is a *path* of length r from j_1 to j_{r+1} provided that $(j_1,j_2),\dots,(j_r,j_{r+1})$ are arcs of $\mathcal{D}(L)$. A closed path is a *cycle*. A cycle with no repeated vertices is called *elementary*.

There is a bijective correspondence between Boolean variables $X \in \mathcal{B}$ and vectors $\mathbf{x} \in \mathcal{L}_2$, defined by the relationship

$$\mathbf{x} = \begin{bmatrix} X \\ \bar{X} \end{bmatrix}. \qquad (1)$$

We introduce the *(left) semi-tensor product* $\ltimes$ between matrices (in particular, vectors) [3]: given $L_1 \in \mathbb{R}^{r_1\times c_1}$ and $L_2 \in \mathbb{R}^{r_2\times c_2}$ (in particular, $L_1 \in \mathcal{L}_{r_1\times c_1}$ and $L_2 \in \mathcal{L}_{r_2\times c_2}$), we set $L_1 \ltimes L_2 := (L_1 \otimes I_{T/c_1})(L_2 \otimes I_{T/r_2}), T := \text{l.c.m.}\{c_1, r_2\}$, where l.c.m. denotes the least common multiple. If $c_1 = r_2$ then $L_1 \ltimes L_2 = L_1 L_2$. So, the semi-tensor product extends the standard matrix product. By resorting to it, we extend (1) into a bijective correspondence between $\mathcal{B}^n$ and $\mathcal{L}_{2^n}$: given $X = [\,X_1 \quad X_2 \quad \dots \quad X_n\,]^\top \in \mathcal{B}^n$, we set

$$\mathbf{x} := \begin{bmatrix} X_1 \\ \bar{X}_1 \end{bmatrix} \ltimes \begin{bmatrix} X_2 \\ \bar{X}_2 \end{bmatrix} \ltimes \dots \ltimes \begin{bmatrix} X_n \\ \bar{X}_n \end{bmatrix}.$$

Given a sequence $(\mathbf{w}(t))_{t\in\mathbb{Z}_+}$, we denote by $(\mathbf{w}(t))|_{[k,n]}$ its restriction to the "discrete window" $[k,n], k,n \in \mathbb{Z}_+, k \le n$. Similarly, given a set of sequences $\mathfrak{B}$, we denote by $\mathfrak{B}|_{[k,n]} := \{(\mathbf{w}(t))|_{[k,n]} : \exists\ (\mathbf{w}(t))_{t\in\mathbb{Z}_+} \in \mathfrak{B}\}$, the restriction of $\mathfrak{B}$ to $[k,n]$. The length of the window $[k,n]$ is $n-k+1$.

II. Completeness and T-completeness

A *Boolean Control Network* (BCN) is described by the following equations

$$\begin{aligned} X(t+1) &= f(X(t),U(t)), \\ Y(t) &= h(X(t)), \qquad t \in \mathbb{Z}_+, \end{aligned} \qquad (2)$$

where $X(t)$, $U(t)$ and $Y(t)$ denote the state variable, the input and the output at time t, taking values in $\mathcal{B}^n, \mathcal{B}^m$ and $\mathcal{B}^p$, respectively. f and h are logic functions, i.e. $f : \mathcal{B}^n \times \mathcal{B}^m \to \mathcal{B}^n$ and $h : \mathcal{B}^n \to \mathcal{B}^p$. By resorting to the semi-tensor product $\ltimes$, the BCN (2) can be described as [3]

$$\begin{array}{rcl} \mathbf{x}(t+1) &=& L \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \\ \mathbf{y}(t) &=& H \ltimes \mathbf{x}(t) = H\mathbf{x}(t), \qquad t \in \mathbb{Z}_+, \end{array} \tag{3}$$

where $\mathbf{x}(t) \in \mathcal{L}_N, \mathbf{u}(t) \in \mathcal{L}_M$ and $\mathbf{y}(t) \in \mathcal{L}_P$, with $N := 2^n, M := 2^m$ and $P := 2^p$. $L \in \mathcal{L}_{N\times NM}$ and $H \in \mathcal{L}_{P\times N}$ are matrices whose columns are canonical vectors. For every $\mathbf{u}(t) = \delta_M^j$, we set $L_j := L \ltimes \mathbf{u}(t) \in \mathcal{L}_{N\times N}$.

The *set* $\mathfrak{B}_{uy}$ *of the input/output trajectories* of the BCN (3) is the set of all pairs $(\mathbf{u}(t), \mathbf{y}(t))_{t\in\mathbb{Z}_+}$ such that $(\mathbf{y}(t))_{t\in\mathbb{Z}_+}$ is the output trajectory generated by (3) corresponding to some initial state $\mathbf{x}(0) = \mathbf{x}_0 \in \mathcal{L}_N$ and to the input $(\mathbf{u}(t))_{t\in\mathbb{Z}_+}$.

Definition 1: [19] The set $\mathfrak{B}_{uy}$ is

- *left-shift invariant* if $(\mathbf{u}(t), \mathbf{y}(t))_{t\in\mathbb{Z}_+} \in \mathfrak{B}_{uy}$ implies $(\mathbf{u}(t+k), \mathbf{y}(t+k))_{t\in\mathbb{Z}_+} \in \mathfrak{B}_{uy}$ for every $k \in \mathbb{Z}_+$;
- *complete* if $(\mathbf{u}(t), \mathbf{y}(t))|_{[\tau,\tau+T]} \in \mathfrak{B}_{uy}|_{[\tau,\tau+T]}$ for every $\tau, T \in \mathbb{Z}_+$ implies $(\mathbf{u}(t), \mathbf{y}(t))_{t\in\mathbb{Z}_+} \in \mathfrak{B}_{uy}$;
- *T-complete* if $(\mathbf{u}(t), \mathbf{y}(t))|_{[\tau,\tau+T]} \in \mathfrak{B}_{uy}|_{[\tau,\tau+T]}$ for every $\tau \in \mathbb{Z}_+$ implies $(\mathbf{u}(t), \mathbf{y}(t))_{t\in\mathbb{Z}_+} \in \mathfrak{B}_{uy}$.

Proposition 1: $\mathfrak{B}_{uy}$ is left shift-invariant and complete.

Proof: Left shift-invariance is obvious, so we only need to prove completeness. Given a sequence $(\mathbf{u}(t), \mathbf{y}(t))_{t\in\mathbb{Z}_+} \in (\mathcal{L}_M \times \mathcal{L}_P)^{\mathbb{Z}_+}$, we consider its restrictions to all finite windows $[0, T]$, with $T \in \mathbb{Z}_+$. If $(\mathbf{u}(t), \mathbf{y}(t))|_{[0,T]} \in \mathfrak{B}_{uy}|_{[0,T]}$, the set

$$\begin{aligned} \mathcal{X}_0^T := \{\mathbf{x}_0 \in \mathcal{L}_N : H\mathbf{x}_0 = \mathbf{y}(0), H(L \ltimes \mathbf{u}(0) \ltimes \mathbf{x}_0) = \mathbf{y}(1), \\ H(L \ltimes \mathbf{u}(1) \ltimes L \ltimes \mathbf{u}(0) \ltimes \mathbf{x}_0) = \mathbf{y}(2), ..., \\ H(L \ltimes \mathbf{u}(T-1) \ltimes \dots L \ltimes \mathbf{u}(0) \ltimes \mathbf{x}_0) = \mathbf{y}(T)\} \end{aligned}$$

of initial states that are compatible with this portion of trajectory is not empty. Also, $\mathcal{X}_0^0 \supseteq \mathcal{X}_0^1 \supseteq \mathcal{X}_0^2 \supseteq \dots$. As these sets have finite cardinality, if $\lim_{T\to+\infty} \mathcal{X}_0^T = \emptyset$, i.e., $(\mathbf{u}(t), \mathbf{y}(t))_{t\in\mathbb{Z}_+} \notin \mathfrak{B}_{uy}$, then there must be a finite $\tilde{T}$ (depending on the specific pair) such that $\mathcal{X}_0^{\tilde{T}} = \emptyset$. So, by performing a check on the windows $[0, T], T \in \mathbb{Z}_+$ (and hence on all finite windows $[\tau, \tau + T], \tau, T \in \mathbb{Z}_+$), we can decide whether a given sequence is a trajectory of $\mathfrak{B}_{uy}$ or not. ■

As clarified in the previous proof, for every sequence $(\mathbf{u}(t), \mathbf{y}(t))_{t\in\mathbb{Z}_+} \in (\mathcal{L}_M \times \mathcal{L}_P)^{\mathbb{Z}_+}$ that is not in $\mathfrak{B}_{uy}$ there exists $\tilde{T} \in \mathbb{Z}_+$ such that $(\mathbf{u}(t), \mathbf{y}(t))|_{[0,\tilde{T}]} \notin \mathfrak{B}_{uy}|_{[0,\tilde{T}]}$. However, in general, we cannot put an upper bound on the length $\tilde{T}$, since it may depend on the specific sequence. This amounts to saying that the $\mathfrak{B}_{uy}$ may be complete but not necessarily T-complete (see Example 1 in [7]).

T-completeness of $\mathfrak{B}_{uy}$ is equivalent to the fact that by suitably concatenating two input/output trajectories in $\mathfrak{B}_{uy}$, that coincide on T consecutive time instants, we obtain another trajectory in $\mathfrak{B}_{uy}$. This fact, whose proof follows the same lines as the one given by J.C. Willems in [19], is essential in order to provide the graph theoretic characterization of T-completeness given in Proposition 2, below.

Lemma 1: The set $\mathfrak{B}_{uy}$ is T-complete if and only if, for every choice of the trajectories $(\mathbf{u}_1(t), \mathbf{y}_1(t))_{t\in\mathbb{Z}_+}$, $(\mathbf{u}_2(t), \mathbf{y}_2(t))_{t\in\mathbb{Z}_+}$ in $\mathfrak{B}_{uy}$ and every $\tau \in \mathbb{Z}_+$, condition

$$(\mathbf{u}_1(t), \mathbf{y}_1(t)) = (\mathbf{u}_2(t), \mathbf{y}_2(t)), \ \forall\ t \in [\tau, \tau + T - 1], \tag{4}$$

implies that $\mathfrak{B}_{uy}$ includes the pair

$$(\mathbf{u}(t), \mathbf{y}(t)) := \begin{cases} (\mathbf{u}_1(t), \mathbf{y}_1(t)), & 0 \le t \le \tau - 1; \\ (\mathbf{u}_2(t), \mathbf{y}_2(t)), & t \ge \tau; \end{cases} \quad t \in \mathbb{Z}_+,$$

Proposition 2: The set $\mathfrak{B}_{uy}$ is not T-complete for any choice of T if and only if two distinct states $\mathbf{x}'$ and $\mathbf{x}'' \in \mathcal{L}_N$ can be found such that the following three conditions hold:

i) there exists a periodic input $(\mathbf{u}(t))_{t\in\mathbb{Z}_+}$ such that the state trajectories $(\mathbf{x}_1(t))_{t\in\mathbb{Z}_+}$, with $\mathbf{x}_1(0) = \mathbf{x}'$, and $(\mathbf{x}_2(t))_{t\in\mathbb{Z}_+}$, with $\mathbf{x}_2(0) = \mathbf{x}''$, are periodic, and the corresponding output sequences, $(\mathbf{y}_1(t))_{t\in\mathbb{Z}_+}$ and $(\mathbf{y}_2(t))_{t\in\mathbb{Z}_+}$ coincide;

ii) there exist $\mathbf{x}_{in} \in \mathcal{L}_N, r \in \mathbb{Z}_+, r > 0$, and an input sequence $\tilde{\mathbf{u}}(0), \tilde{\mathbf{u}}(1), \dots, \tilde{\mathbf{u}}(r-1)$ that drives the state from $\mathbf{x}(0) = \mathbf{x}_{in}$ to $\mathbf{x}(r) = \mathbf{x}'$, meanwhile generating the output sequence $\tilde{\mathbf{y}}(0), \tilde{\mathbf{y}}(1), \dots, \tilde{\mathbf{y}}(r-1)$, but $\mathbf{x}''$ cannot be reached at $t = r$, by making use of the same input sequence and generating the same output sequence;

iii) there exists an input sequence $(\bar{\mathbf{u}}(t))_{t\in\mathbb{Z}_+}$ that if applied starting from $\mathbf{x}(0) = \mathbf{x}''$ generates an output sequence $(\bar{\mathbf{y}}(t))_{t\in\mathbb{Z}_+}$ that cannot be generated by applying the same input starting from $\mathbf{x}(0) = \mathbf{x}'$.

Proof: Suppose that conditions i)-iii) hold and let k be the common period of the input/state trajectories $(\mathbf{x}_1(t), \mathbf{u}(t))_{\mathbb{Z}_+}$ and $(\mathbf{x}_2(t), \mathbf{u}(t))_{\mathbb{Z}_+}$. Note that, for every choice of T, there exist $c, d \in \mathbb{Z}_+$ and $q \in [0, k-1]$ such that $ck \ge T$ and $r = dk - q$, and consider the two input/output trajectories described as follows:

$$(\hat{\mathbf{u}}_1(t), \hat{\mathbf{y}}_1(t)) = \begin{cases} (\tilde{\mathbf{u}}(t), \tilde{\mathbf{y}}(t)), & 0 \le t \le r-1; \\ (\mathbf{u}(t-r), \mathbf{y}_1(t-r)), & t \ge r; \end{cases}$$

$$(\hat{\mathbf{u}}_2(t), \hat{\mathbf{y}}_2(t)) = \begin{cases} (\mathbf{u}(t+q), \mathbf{y}_2(t+q)), & 0 \le t \le ck + r - 1; \\ (\bar{\mathbf{u}}(t-ck-r), \bar{\mathbf{y}}(t-ck-r)), & t \ge ck + r. \end{cases}$$

By the assumptions i)-iii) both these trajectories belong to $\mathfrak{B}_{uy}$ but, even if they coincide in the time interval $[r, r+ck-1] \supseteq [r, r+T-1]$, they cannot be concatenated together thus getting a new trajectory in $\mathfrak{B}_{uy}$. Indeed, the trajectory

$$(\hat{\mathbf{u}}(t), \hat{\mathbf{y}}(t)) = \begin{cases} (\hat{\mathbf{u}}_1(t), \hat{\mathbf{y}}_1(t)), & 0 \le t \le r-1; \\ (\hat{\mathbf{u}}_2(t), \hat{\mathbf{y}}_2(t)), & t \ge r; \end{cases}$$

is not in $\mathfrak{B}_{uy}$. Therefore, by Lemma 1, $\mathfrak{B}_{uy}$ is not T-complete.

Conversely, suppose that $\mathfrak{B}_{uy}$ is not T-complete for any choice of T and introduce the following notation. Given a sequence $(\mathbf{u}(t), \mathbf{y}(t))_{t\in\mathbb{Z}_+} \in (\mathcal{L}_M \times \mathcal{L}_P)^{\mathbb{Z}_+}$, let $\mathcal{X}(\mathbf{u}(t), \mathbf{y}(t))$ denote the (possibly empty) set of state trajectories compatible with it, namely the set of all state trajectories $(\mathbf{x}(t))_{t\in\mathbb{Z}_+} \in (\mathcal{L}_N)^{\mathbb{Z}_+}$ such that, by choosing $\mathbf{x}(0)$ as initial state and $(\mathbf{u}(t))_{t\in\mathbb{Z}_+}$ as input, one gets $(\mathbf{x}(t))_{t\in\mathbb{Z}_+}$ and $(\mathbf{y}(t))_{t\in\mathbb{Z}_+}$ as state and output trajectories, respectively.
Assume, without loss of generality, that $T \ge MPN^2 + 1$. By Lemma 1, there exist two input/output trajectories

$(\mathbf{u}_1(t), \mathbf{y}_1(t))_{t\in\mathbb{Z}_+}, (\mathbf{u}_2(t), \mathbf{y}_2(t))_{t\in\mathbb{Z}_+} \in \mathfrak{B}_{uy}$ and $\tau \in \mathbb{Z}_+$ such that condition (4) holds, but the pair

$$(\mathbf{u}(t), \mathbf{y}(t)) := \begin{cases} (\mathbf{u}_1(t), \mathbf{y}_1(t)), & 0 \le t \le \tau - 1; \\ (\mathbf{u}_2(t), \mathbf{y}_2(t)), & t \ge \tau; \end{cases} \quad t \in \mathbb{Z}_+,$$

does not belong to $\mathfrak{B}_{uy}$. Consider two state trajectories: $(\mathbf{x}_1(t))_{t\in\mathbb{Z}_+} \in \mathcal{X}(\mathbf{u}_1, \mathbf{y}_1(t))$ and $(\mathbf{x}_2(t))_{t\in\mathbb{Z}_+} \in \mathcal{X}(\mathbf{u}_2, \mathbf{y}_2(t))$. By the assumption on T and condition (4), there exist $t_a, t_b \in [\tau, \tau + T - 1], t_a < t_b$, such that $(\mathbf{u}_1(t_a), \mathbf{y}_1(t_a)) = (\mathbf{u}_2(t_a), \mathbf{y}_2(t_a)) = (\mathbf{u}_2(t_b), \mathbf{y}_2(t_b)) = (\mathbf{u}_1(t_b), \mathbf{y}_1(t_b))$, and $(\mathbf{x}_1(t_a), \mathbf{x}_2(t_a)) = (\mathbf{x}_1(t_b), \mathbf{x}_2(t_b))$.

Clearly, $\mathbf{x}' := \mathbf{x}_1(t_a) \neq \mathbf{x}_2(t_a) =: \mathbf{x}''$, otherwise the two state trajectories $(\mathbf{x}_1(t))_{t\in\mathbb{Z}_+} \in \mathcal{X}(\mathbf{u}_1, \mathbf{y}_1(t))$ and $(\mathbf{x}_2(t))_{t\in\mathbb{Z}_+} \in \mathcal{X}(\mathbf{u}_2, \mathbf{y}_2(t))$ would coincide till $t = \tau + T - 1$ and hence the two input/output trajectories could be concatenated. So, the two distinct periodic state/input sequences of period $k := t_b - t_a$ obtained as

$$(\mathbf{x}_{p1}(t), \mathbf{u}(t)) := \begin{cases} (\mathbf{x}_1(t + t_a), \mathbf{u}_1(t + t_a)), & 0 \le t \le k - 1; \\ (\mathbf{x}_{p1}(t - k), \mathbf{u}(t - k)), & t \ge k \end{cases}$$

$$(\mathbf{x}_{p2}(t), \mathbf{u}(t)) := \begin{cases} (\mathbf{x}_2(t + t_a), \mathbf{u}_2(t + t_a)), & 0 \le t \le k - 1; \\ (\mathbf{x}_{p2}(t - k), \mathbf{u}(t - k)), & t \ge k \end{cases}$$

and hence corresponding to the same input, generate the same output sequence, thus proving i). By the way these trajectories have been obtained from $(\mathbf{u}_1(t), \mathbf{y}_1(t))_{t\in\mathbb{Z}_+}, (\mathbf{u}_2(t), \mathbf{y}_2(t))_{t\in\mathbb{Z}_+} \in \mathfrak{B}_{uy}$, conditions ii) and iii) are satisfied, too. ■

Proposition 2 shows that T-completeness is related to the way a BCN can "reach" and "leave" its periodic trajectories. Fig. 1 illustrates the idea underlying Proposition 2. The two cycles represent the two periodic state trajectories obtained starting from $\mathbf{x}'$ and $\mathbf{x}''$, respectively, corresponding to the input $\mathbf{u}$ and generating the same output $\mathbf{y}_1 = \mathbf{y}_2$ (condition i)). In the upper trajectory, the state $\mathbf{x}_{in}$ can be connected to $\mathbf{x}'$ through the input $\tilde{\mathbf{u}}$, while generating the output $\tilde{\mathbf{y}}$, but no state can be connected to $\mathbf{x}''$, by resorting to $\tilde{\mathbf{u}}$, in the lower trajectory, while generating the output $\tilde{\mathbf{y}}$ (condition ii)). The input output pair $(\bar{\mathbf{u}}, \bar{\mathbf{y}})$, appearing in the lower trajectory but prohibited in the upper one, illustrates condition iii).

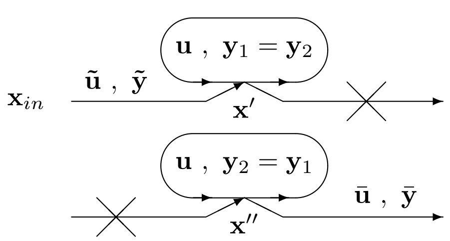

Fig. 1: Intuitive description of lack of T-completeness

III. Fault detection

Given a BCN, we want to investigate the problem of determining, from the measurement of its input and output trajectories, whether a fault has affected the BCN functioning. The first step toward this direction is to define what do we mean by a fault and what may be the outcome of a fault. Here we consider the basic set-up of a BCN with two possible configurations: a non-faulty (NF) and a faulty (F) one, and the fault affects only the state-update, not the output measurements. We henceforth represent the non-faulty BCN as in (3) and the faulty one as

$$\begin{aligned} \mathbf{x}(t+1) &= L^{(F)} \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \\ \mathbf{y}(t) &= H \ltimes \mathbf{x}(t) = H\mathbf{x}(t), \qquad t \in \mathbb{Z}_+, \end{aligned} \tag{5}$$

and we set $L_j^{(F)} := L^{(F)} \ltimes \delta_M^j$, $j \in [1, M]$. If we introduce the fault signal $(\mathbf{f}(t))_{t\in\mathbb{Z}_+}$, taking values in $\mathcal{L}_2$, and we assume that $\mathbf{f}(t) = \delta_2^1$ corresponds to the non-faulty BCN and $\mathbf{f}(t) = \delta_2^2$ to the faulty one, the overall BCN dynamics becomes

$$\begin{aligned} \mathbf{x}(t+1) &= \tilde{L} \ltimes \mathbf{f}(t) \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \\ \mathbf{y}(t) &= H \ltimes \mathbf{x}(t) = H\mathbf{x}(t), \qquad t \in \mathbb{Z}_+. \end{aligned} \tag{6}$$

where $\tilde{L} := [L \quad L^{(F)}] \in \mathcal{L}_{N\times 2NM}$. We assume that the BCN cannot autonomously recover from a fault; so once the fault signal switches from δ_2^1 to δ_2^2, it cannot switch back to δ_2^1, and the fault sequence is described by a step function

$$\mathbf{f}(t) = \begin{cases} \delta_2^1, & \text{for } 0 \le t < \bar{t}; \\ \delta_2^2, & \text{for } t \ge \bar{t}, \end{cases} \tag{7}$$

where $\bar{t} = +\infty$ in case no fault affects the BCN. We want to investigate under what conditions we can detect the fault occurrence from the measurement of the input and output sequences generated by the BCN (6). To better formalize this problem and its solution, we denote by $\mathbf{x}(t; \mathbf{x}_0, \mathbf{u}(\cdot), \mathbf{f}(\cdot))$ and $\mathbf{y}(t; \mathbf{x}_0, \mathbf{u}(\cdot), \mathbf{f}(\cdot))$, the state and output vectors of the BCN (6) at time t, when it starts from $\mathbf{x}(0) = \mathbf{x}_0$ and the input and fault sequences are $\mathbf{u}(\cdot)$ and $\mathbf{f}(\cdot)$, respectively.

As a preliminary remark, we notice that a fault taking place at time $\bar{t}$, for certain values of $\bar{\mathbf{x}} := \mathbf{x}(\bar{t}) \in \mathcal{L}_N$ and $\mathbf{u}(t), t \ge \bar{t}$, may not reveal itself. Indeed, it is possible that the state trajectory generated by the faulty BCN (5) starting from $\bar{\mathbf{x}}$ at $t = \bar{t}$, under the effect of $\mathbf{u}$, coincides with the state trajectory that the non-faulty BCN (3) generates in the same conditions. This is not an unreasonable situation, since it corresponds to the case when the faulty part of the system is not involved in the dynamic evolution and hence the fault cannot be detected. Under this perspective, it is convenient to introduce the concept of *meaningful fault*.

Definition 2: Given an initial state $\mathbf{x}_0 \in \mathcal{L}_N$ and an input sequence $(\mathbf{u}(t))_{t\in\mathbb{Z}_+}$, a fault sequence $(\mathbf{f}(t))_{t\in\mathbb{Z}_+}$ induces a *meaningful fault* for the BCN (6) if the state trajectory $(\mathbf{x}(t; \mathbf{x}_0, \mathbf{u}(\cdot), \mathbf{f}(\cdot)))_{t\in\mathbb{Z}_+}$, generated by (6) corresponding to $\mathbf{x}_0$, $\mathbf{u}$ and $\mathbf{f}$, is different from the state trajectory $(\mathbf{x}(t; \mathbf{x}_0, \mathbf{u}(\cdot), \delta_2^1))_{t\in\mathbb{Z}_+}$ (which coincides with the one) generated by the non-faulty system (3) corresponding to the same initial condition and input.

Meaningful fault sequences are the only ones we may hope to detect, by making use of the input and output trajectories, and hence we will restrict our attention to them. Moreover, we will move our attention from the time $\bar{t}$ to the first time $t_f \ge \bar{t}$ a meaningful fault modifies the state trajectory We can now formalize the concept of detectable fault.

Definition 3: Given a BCN (6), an initial state $\mathbf{x}_0 \in \mathcal{L}_N$, an input sequence $(\mathbf{u}(t))_{t\in\mathbb{Z}_+}$, and a (meaningful) fault sequence $(\mathbf{f}(t))_{t\in\mathbb{Z}_+}$, we say that the *(meaningful) fault is detectable* if the input/output pair $(\mathbf{u}(t), \mathbf{y}(t; \mathbf{x}_0, \mathbf{u}(\cdot), \mathbf{f}(\cdot))_{t\in\mathbb{Z}_+}$ generated by the BCN (6) does not belong to $\mathfrak{B}_{uy}$.

To answer the problem we posed, it is convenient to define

$$X^* := \{\mathbf{x}^* \in \mathcal{L}_N : \exists\, \mathbf{u}^* \in \mathcal{L}_M \text{ s.t. } L \ltimes \mathbf{u}^* \ltimes \mathbf{x}^* \neq L^{(F)} \ltimes \mathbf{u}^* \ltimes \mathbf{x}^*\}, \qquad (8)$$

and, for every $\mathbf{x}^* \in X^*$,

$$U^*(\mathbf{x}^*) := \{\mathbf{u}^* \in \mathcal{L}_M : L \ltimes \mathbf{u}^* \ltimes \mathbf{x}^* \neq L^{(F)} \ltimes \mathbf{u}^* \ltimes \mathbf{x}^*\}. \quad (9)$$

Also, $\mathcal{U}Y^*$ denotes the set of input/output trajectories $(\mathbf{u}(t), \mathbf{y}^{(F)}(t)) \in (\mathcal{L}_M \times \mathcal{L}_P)^{\mathbb{Z}_+}$ generated by the faulty BCN (5) corresponding to some $\mathbf{x}(0) = \mathbf{x}^* \in X^*$ and to some input $(\mathbf{u}(t))_{t\in\mathbb{Z}_+}$ with $\mathbf{u}(0) \in U^*(\mathbf{x}^*)$. In other words, we focus on input/output trajectories for which a fault located at $t = 0$ is meaningful and modifies that state evolution starting at $t = 1$.

Proposition 3: For the BCN (6) the following facts are equivalent:

i) for every initial condition $\mathbf{x}_0 \in \mathcal{L}_N$ and every input sequence $(\mathbf{u}(t))_{t\in\mathbb{Z}_+}$, every fault that is meaningful (for the specific choice of $\mathbf{x}_0$ and $\mathbf{u}$) is also detectable;

ii) $$\mathcal{U}Y^* \cap \mathfrak{B}_{uy} = \emptyset. \qquad (10)$$

Proof: If condition (10) does not hold, there exist $\mathbf{x}_0 \in X^*$ and $(\mathbf{u}(t))_{t\in\mathbb{Z}_+}$, with $\mathbf{u}(0) \in U^*(\mathbf{x}_0)$, such that $\mathbf{f}(t) = \delta_2^2, \forall\, t \in \mathbb{Z}_+$, is a meaningful fault sequence, but the fault cannot be detected. Conversely, suppose there exist $\mathbf{x}_0 \in X^*$ and an input sequence $(\mathbf{u}(t))_{t\in\mathbb{Z}_+}$, for which some fault sequence $(\mathbf{f}(t))_{t\in\mathbb{Z}_+}$, described as in (7), is meaningful but not detectable. This means that the corresponding pair $(\mathbf{u}(t), \mathbf{y}(t))_{t\in\mathbb{Z}_+}$ belongs to $\mathfrak{B}_{uy}$. If $t_f \geq \bar{t}$ is the first time instant such that $L \ltimes \mathbf{u}(t_f) \ltimes \mathbf{x}(t_f) \neq L^{(F)} \ltimes \mathbf{u}(t_f) \ltimes \mathbf{x}(t_f)$, then $\mathbf{x}(t_f) \in X^*$, $\mathbf{u}(t_f) \in U^*(\mathbf{x}(t_f))$ and the portion of input/output trajectories $(\mathbf{u}(t), \mathbf{y}(t))|_{[t_f,+\infty)}$ belongs to both $\mathcal{U}Y^*$ and $\mathfrak{B}_{uy}$, thus contradicting (10). ■

Condition (10) can be checked by resorting to a graph theoretic approach. The idea is to introduce a graph that is able to keep in parallel the state-transitions in the non-faulty BCN and in the faulty one, starting from any pair of states and corresponding to any input sequence. We introduce the *NF-F (non-faulty-faulty) directed graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where

- the vertex set $\mathcal{V}$ is the set of all pairs of states, namely $\{(\delta_N^i, \delta_N^j) \in \mathcal{L}_N \times \mathcal{L}_N\}$;
- the labeled edge set $\mathcal{E}$ is defined as follows: there is an edge labeled by $\mathbf{u} \in \mathcal{L}_M$ from the pair (δ_N^i, δ_N^j) to the pair (δ_N^h, δ_N^k) if and only if $\delta_N^h = L \ltimes \mathbf{u} \ltimes \delta_N^i$ and $\delta_N^k = L^{(F)} \ltimes \mathbf{u} \ltimes \delta_N^j$. Note that from every pair (δ_N^i, δ_N^j) there are M outgoing arcs, one for each value of the input $\mathbf{u}$.

The vertex set is partitioned into 2 classes: C_0 and C_1. A pair (δ_N^i, δ_N^j) belongs to C_1 if $H\delta_N^i = H\delta_N^j$, while it belongs to C_0 if $H\delta_N^i \neq H\delta_N^j$.

Proposition 4: Given the BCN (6), let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the associated NF-F directed graph. All meaningful faults affecting the BCN are detectable if and only if each path in $\mathcal{G}$ endowed with the properties:

P1) it starts from some vertex pair $(\mathbf{x}_0, \mathbf{x}^*) \in \mathcal{L}_N \times X^*$;

P2) the first arc of the path (outgoing from $(\mathbf{x}_0, \mathbf{x}^*)$) is labeled by some $\mathbf{u}^* \in U^*(\mathbf{x}^*)$;

eventually enters the class C_0.

Proof: Condition (10) holds if and only if for every $\mathbf{x}^* \in X^*$, every $\mathbf{u}^* \in U^*(\mathbf{x}^*)$, and every input sequence $(\mathbf{u}(t))_{t\in\mathbb{Z}_+}$, with $\mathbf{u}(0) = \mathbf{u}^*$, no state $\mathbf{x}_0 \in \mathcal{L}_N$ can be found such that $\mathbf{y}(t; \mathbf{x}_0, \mathbf{u}(\cdot), \delta_2^1) = \mathbf{y}(t; \mathbf{x}^*, \mathbf{u}(\cdot), \delta_2^2)$, for every $t \in \mathbb{Z}_+$. This amounts to saying that for every $\mathbf{x}^* \in X^*$, every $\mathbf{u}^* \in U^*(\mathbf{x}^*)$, every input sequence $(\mathbf{u}(t))_{t\in\mathbb{Z}_+}$, with $\mathbf{u}(0) = \mathbf{u}^*$, and every state $\mathbf{x}_0 \in \mathcal{L}_N$ there exists a time instant $\hat{t} \geq 0$ such that $\mathbf{x}_1 := \mathbf{x}(\hat{t}; \mathbf{x}_0, \mathbf{u}(\cdot), \delta_2^1)$ generates an output that is different from the output generated by $\mathbf{x}_2 := \mathbf{x}(\hat{t}; \mathbf{x}^*, \mathbf{u}(\cdot), \delta_2^2)$. This simply means that $(\mathbf{x}_1, \mathbf{x}_2) \in C_0$. ■

Remark 1: Proposition 4 provides a necessary and sufficient condition for all meaningful faults to be detectable. The existence of a not detectable meaningful fault corresponds, henceforth, to the case when a path can be found, satisfying P1) and P2) but never leaving the class C_1. This ensures the existence in C_1 of a cycle that can be reached from the pair $(\mathbf{x}_0, \mathbf{x}^*)$. Note that the existence of a cycle in C_1 is equivalent to the existence of a not detectable fault. However, this fault is not necessarily meaningful, unless it can be reached starting from a pair $(\mathbf{x}_0, \mathbf{x}^*)$ satisfying P1) and P2).

Remark 2: Proposition 4 provides a way (at least when N, M and P are not too large) to check whether meaningful faults are always detectable. Indeed, one simply needs to explore in the NF-F graph all the paths endowed with properties P1) and P2) and see after how many steps they enter C_0. One may wonder how long these paths may be, in the worst case, and hence how heavy is this test from a computational viewpoint. If every path satisfying P1) and P2) eventually enters C_0, it cannot encounter the same vertex pair in C_1 twice. So its length is upper bounded by the cardinality of C_1, and, in the worst case, the maximum number of distinct path (of length $|C_1|$) we have to evaluate in the NF-F digraph is upper-bounded by $|X^*|R(\max_{\mathbf{x}^*\in X^*} |U^*(\mathbf{x}^*)|)M^{|C_1|-1} \ll N^2 M^{|C_1|}$, where R is the cardinality of the largest set of states that correspond to the same output value.

Remark 3: If all meaningful faults are detectable, when any such fault $\mathbf{f}$ affects the BCN and t_f denotes the smallest $t \geq 0$ such that $\mathbf{x}(t, \mathbf{x}(0), \mathbf{u}(\cdot), \mathbf{f}(\cdot)) \neq \mathbf{x}(t, \mathbf{x}(0), \mathbf{u}(\cdot), \delta_2^1)$, we have $(\mathbf{u}(t), \mathbf{y}(t))|_{[0,t_f+D]} \notin \mathfrak{B}_{uy}|_{[0,t_f+D]}$, and $\mathbf{f}$ can be detected within $D \leq |C_1|$ time instants from t_f. As there are no upper bounds on t_f, the idea of storing the input/output data from $t = 0$ to $t = t_f + D$ is not feasible, unless we assume the T-completeness of $\mathfrak{B}_{uy}$ for some $T \geq 0$. If so, we can store and update the samples on a sliding window of length $T + 1$. As we will see in the next section, in order to detect faults it is more convenient to exploit the knowledge of the internal structure of the BCN, thus performing detection algorithms that do not require the T-completeness of $\mathfrak{B}_{uy}$.

Example 1: Consider a BCN (3) with $N = 4$, $M = 2$, $P = 4$ and

$$\begin{aligned} L_1 &:= L \ltimes \delta_2^1 = [\,\delta_4^2 \quad \delta_4^3 \quad \delta_4^2 \quad \delta_4^4\,], \\ L_2 &:= L \ltimes \delta_2^2 = [\,\delta_4^1 \quad \delta_4^3 \quad \delta_4^4 \quad \delta_4^4\,], \\ H &:= [\,\delta_4^1 \quad \delta_4^2 \quad \delta_4^2 \quad \delta_4^3\,]. \end{aligned}$$

The BCN is represented by the digraph of Fig. 2, obtained by overlapping the digraphs $\mathcal{D}(L_1)$ and $\mathcal{D}(L_2)$. (Blue) continuous arcs belong to $\mathcal{D}(L_1)$, while (red) dashed arcs belong to $\mathcal{D}(L_2)$. For each vertex j (corresponding to δ_4^j), in addition to the two outgoing arcs describing the state transitions associated with $\mathbf{u}=\delta_2^1$ and $\mathbf{u}=\delta_2^2$, there is an arrow describing the output value associated with it ($\mathbf{y}=\delta_4^j, j\in[1,4]$).

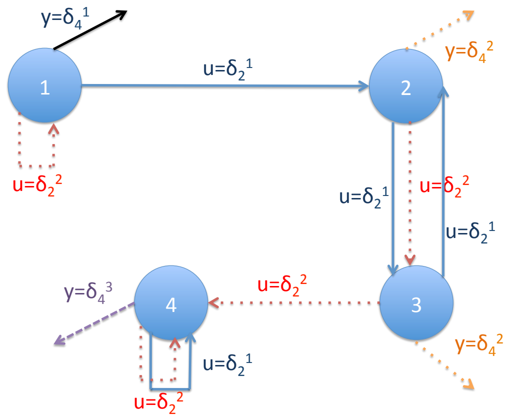

Fig. 2: Digraph corresponding to the (non-faulty) BCN of Example 1

Assume, now, that, as a consequence of a fault, the matrices describing the BCN become (see Fig. 3)

$$\begin{aligned} L_1^{(F)} &:= L^{(F)} \ltimes \delta_2^1 = [\,\delta_4^4 \quad \delta_4^3 \quad \delta_4^2 \quad \delta_4^4\,], \\ L_2^{(F)} &:= L^{(F)} \ltimes \delta_2^2 = [\,\delta_4^1 \quad \delta_4^3 \quad \delta_4^4 \quad \delta_4^1\,], \\ H &:= [\,\delta_4^1 \quad \delta_4^2 \quad \delta_4^2 \quad \delta_4^3\,]. \end{aligned}$$

It is easy to see that $L^{(F)} = [\,L_1^{(F)} \quad L_2^{(F)}\,]$ differs from $L=[\,L_1 \quad L_2\,]$ only in the first and last columns. $X^* = \{\delta_4^1,\delta_4^4\}$ and $U^*(\delta_4^1)=\{\delta_2^1\}$, while $U^*(\delta_4^4)=\{\delta_2^2\}$. In the NF-F directed graph, all paths starting from $(\mathbf{x}_0,\delta_4^1)$, with first outgoing arc labelled by δ_2^1, either start in C_0 (for $\mathbf{x}_0\neq\delta_4^1$) or reach C_0 in one step.

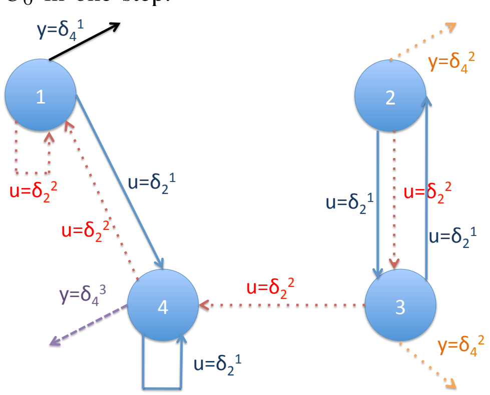

Fig. 3: Faulty version of the BCN in Example 1

All paths starting from $(\mathbf{x}_0,\delta_4^4)$, with first outgoing arc labelled by δ_2^2, either start in C_0 (for $\mathbf{x}_0\neq\delta_4^4$) or reach C_0 in one step. So, all meaningful faults are detectable in $D=1$ step. ♠

IV. Fault detection algorithms

In this section we propose two different solutions to the problem of testing whether a fault occurred or not, by assuming that all meaningful faults are detectable, but $\mathfrak{B}_{uy}$ is not necessarily T-complete. The ideas underlying the two algorithms are two classical ones, well known in the literature about model-based fault detection for linear systems since the eighties (see [8], [10]). The first algorithm relies on the idea of producing copies of the original systems that are not affected by faults. By comparing the outputs they produce with the output produced by the (potentially faulty) real system, one can detect fault occurrence. The second algorithm, on the other hand, exploits observer-based fault detection, one of the most popular techniques based on analytical redundancy. The idea is to use the BCN input and output to estimate the current BCN state. If none of the possible N states is compatible with the input and output values generated by the BCN up to some time, a fault has occurred. From a computational viewpoint, we have in both cases dynamical systems working in parallel with the original BCN. The first technique requires $r\le R<N$ (the meaning of the symbols will be explained below) copies of the original BCN, but the fault detection algorithm reduces to a simple comparison. The second technique, on the other hand, is based on a Boolean system of size N that at each steps performs two Boolean products and one AND operation.

A. *Fault-detection based on physical redundancy*

Since the initial state of the BCN is a canonical vector in $\mathcal{L}_N$, for any input sequence $\mathbf{u}(t), t\in\mathbb{Z}_+$, there are N (not necessarily distinct) output trajectories. If the output trajectory we get from the BCN under observation is not one of them, the system is faulty. So, a simple idea is to compare the output trajectory of the BCN under observation with those of N non-faulty BCNs, starting from one of the possible states in $\mathcal{L}_N$ and stimulated by the same input. This idea can be further refined. First, we do not need N copies of the non-faulty BCN to make a comparison. Indeed, from the output at time $t=0$ we know that only the states belonging to $\mathbf{X}_0^{NF} := \{\delta_N^i : H\delta_N^i = \mathbf{y}(0)\}$ are admissible, and the others can be discarded at the first step. Therefore we keep at most R copies of the BCN (3), where R is the cardinality of the largest set of states that correspond to the same output value. Second, as soon as the output of any of these BCN copies differs from the output of the BCN under observation, the comparison does not need to be carried on.

Algorithm 1

[Initialization] Set $\tau := 0, \mathbf{X}_0^{NF} := \{\delta_N^i : H\delta_N^i = \mathbf{y}(0)\} = \{\delta_N^{i_1},\delta_N^{i_2},\dots,\delta_N^{i_r}\}$, where $r := |\mathbf{X}_0^{NF}| \le R$ denotes the cardinality of the set $\mathbf{X}_0^{NF}$. For $k=1,2,\dots,r$, we let $(\mathbf{x}^{(k)}(t))_{t\in\mathbb{Z}_+}$ and $(\mathbf{y}^{(k)}(\tau))_{t\in\mathbb{Z}_+}$ denote the state and the output trajectories of the kth copy of the non-faulty BCN starting from $\mathbf{x}^{(k)}(0) := \delta_N^{i_k}$, and corresponding to $(\mathbf{u}(t))_{t\in\mathbb{Z}_+}$.

We let $\mathbf{C}_\tau$ be the set of the indices of the non-faulty BCNs whose output trajectory coincides with the output trajectory $\mathbf{y}$ of the BCN under observation up to time τ, and initialize $\mathbf{C}_0$ as $\mathbf{C}_0 := [1,r]$.

[Recursive step] Set $\tau := \tau+1$. For every $k\in\mathbf{C}_{t-1}$, if $\mathbf{y}^{(k)}(\tau)\neq\mathbf{y}(\tau)$, then set $\mathbf{C}_\tau := \mathbf{C}_{\tau-1}\setminus\{k\}$. If $\mathbf{C}_\tau=\emptyset$, then a fault has occurred and the algorithm stops. Otherwise repeat the recursive step.

B. *Observer-based fault detection*

A way to test whether a fault occurred or not, consists in verifying, at every time τ, whether the set $\mathbf{X}_\tau^{NF}$ of the states that the non-faulty BCN (3) can reach at time τ, under the effect of the input sequence $\mathbf{u}(t), t\in[0,\tau-1]$, meanwhile generating the output $\mathbf{y}(t), t\in[0,\tau]$, is non-empty.

In other words, one starts at time $\tau=0$ by determining the set $\mathbf{X}_0^{NF}$ of the initial states compatible with $\mathbf{y}(0)$. At $\tau=1$, one evaluates $\mathbf{X}_1^{NF}$ of the states that are compatible with $\mathbf{y}(1)$ and can be obtained from the states in $\mathbf{X}_0^{NF}$ by applying $\mathbf{u}(0)$. By proceeding in this way, we obtain the sequence of sets $\mathbf{X}_\tau^{NF}$, whose cardinality decreases with τ. If for some τ we have $\mathbf{X}_\tau^{NF} = \emptyset$, a fault has occurred. On the other had, if $\mathbf{X}_\tau^{NF} \neq \emptyset$, the portion of trajectory $(\mathbf{u}(t), \mathbf{y}(t))_{[0,\tau]}$ belongs to $\mathfrak{B}_{uy}|_{[0,\tau]}$, but considering the delay in revealing the fault, we can only ensure that no meaningful fault had affected the system up to time $\tau - D + 1$. At every time τ the indices of the states that are compatible with a given output sample δ_P^j are the indices of the unitary entries of the Boolean vector $H^\top \delta_P^j$. In detail, the algorithm is the following one:

Algorithm 2

[Initialization] Set $\tau := 0, \mathbf{X}_0^{NF} := \{\delta_N^i : H\delta_N^i = \mathbf{y}(0)\} = \{\delta_N^i : [H^\top \mathbf{y}(0)]_i = 1\}$ and $\mathbf{v}_0 := \sum_{\delta_N^i \in \mathbf{X}_0^{NF}} \delta_N^i = H^\top \mathbf{y}(0)$.

[Recursive step] Set $\tau := \tau + 1$, and

$$\mathbf{X}_\tau^{NF} := \{\delta_N^i : [L \ltimes \mathbf{u}(\tau-1) \ltimes \mathbf{v}_{\tau-1}]_i = 1 \wedge [H^\top \mathbf{y}(\tau)]_i = 1\}.$$

If $\mathbf{X}_\tau^{NF} = \emptyset$ then a fault occurred: STOP. Otherwise set $\mathbf{v}_\tau := \sum_{\delta_N^i \in \mathbf{X}_\tau^{NF}} \delta_N^i$ and repeat the recursive step.

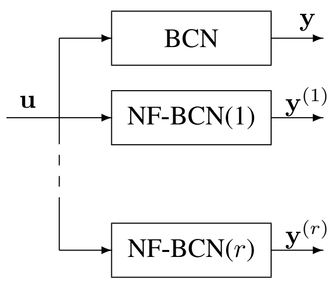

Fig. 4: Block scheme corresponding to Algorithm 1

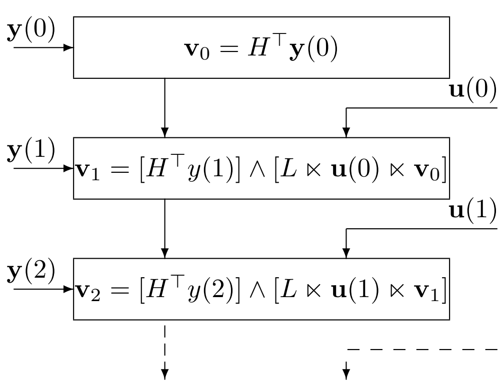

Fig. 5: Flowchart corresponding to Algorithm 2

Example 2: Consider the BCN (3) with matrices

$$\begin{aligned} L_1 &:= L \ltimes \delta_2^1 = [\,\delta_4^2 \quad \delta_4^3 \quad \delta_4^4 \quad \delta_4^1\,], \\ L_2 &:= L \ltimes \delta_2^2 = [\,\delta_4^1 \quad \delta_4^3 \quad \delta_4^3 \quad \delta_4^2\,], \\ H &:= [\,\delta_2^1 \quad \delta_2^1 \quad \delta_2^2 \quad \delta_2^2\,], \end{aligned}$$

initialized at $t = 0$ with $\mathbf{x}(0) = \delta_4^1$. Assume that at time $t = 1$ a fault occurs and the faulty BCN is described as follows

$$\begin{aligned} L_1^{(F)} &:= L^{(F)} \ltimes \delta_2^1 = [\,\delta_4^2 \quad \delta_4^3 \quad \delta_4^4 \quad \delta_4^1\,] = L_1, \\ L_2^{(F)} &:= L^{(F)} \ltimes \delta_2^2 = [\,\delta_4^1 \quad \delta_4^3 \quad \delta_4^3 \quad \delta_4^1\,], \end{aligned}$$

(while H is unaltered) We want to illustrate how the two algorithms work when the input sequence is $\mathbf{u}(0) = \delta_2^1, \mathbf{u}(1) = \delta_2^1, \mathbf{u}(2) = \delta_2^2, \mathbf{u}(3) = \delta_2^1, \mathbf{u}(4) = \delta_2^2, \mathbf{u}(5) = \delta_2^2, \ldots$ and the measured output sequence is $\mathbf{y}(0) = \delta_2^1, \mathbf{y}(1) = \delta_2^1, \mathbf{y}(2) = \delta_2^2, \mathbf{y}(3) = \delta_2^2, \mathbf{y}(4) = \delta_2^2, \mathbf{y}(5) = \delta_2^1, \mathbf{y}(6) = \delta_2^1, \ldots$

Algorithm 1:

For $\tau = 0$, one has $X_0^{NF} = \{\delta_4^1, \delta_4^2\}$. $\mathbf{C}_0 = [1, 2]$, $\mathbf{x}^{(1)}(0) = \delta_4^1 \to \mathbf{y}^{(1)}(0) = \delta_2^1$, $\mathbf{x}^{(2)}(0) = \delta_4^2 \to \mathbf{y}^{(2)}(0) = \delta_2^1$. For $\tau = 1$, one has $\mathbf{x}^{(1)}(1) = \delta_4^2 \to \mathbf{y}^{(1)}(1) = \delta_2^1$, $\mathbf{x}^{(2)}(1) = \delta_4^3 \to \mathbf{y}^{(2)}(1) = \delta_2^2$, and hence $\mathbf{C}_1 = [1]$. For $\tau = 2$, one has $\mathbf{x}^{(1)}(2) = \delta_4^3 \to \mathbf{y}^{(1)}(2) = \delta_2^2$ and hence $\mathbf{C}_2 = [1]$. For $\tau = 3$, one has $\mathbf{x}^{(1)}(3) = \delta_4^3 \to \mathbf{y}^{(1)}(3) = \delta_2^2$ and hence $\mathbf{C}_3 = [1]$. For $\tau = 4$, one has $\mathbf{x}^{(1)}(4) = \delta_4^4 \to \mathbf{y}^{(1)}(4) = \delta_2^2$ and hence $\mathbf{C}_4 = [1]$. For $\tau = 5$, one has $\mathbf{x}^{(1)}(5) = \delta_4^2 \to \mathbf{y}^{(1)}(5) = \delta_2^1$ and hence $\mathbf{C}_5 = [1]$. Finally, for $\tau = 6$, one has $\mathbf{x}^{(1)}(6) = \delta_4^3 \to \mathbf{y}^{(1)}(6) = \delta_2^2$ and hence $\mathbf{C}_6 = \emptyset$. A fault has occurred.

Algorithm 2:

$X_0^{NF} = \{\delta_4^1, \delta_4^2\}, \mathbf{v}_0 = [1 \quad 1 \quad 0 \quad 0]^\top$.
$X_1^{NF} = \{\delta_4^2\}, \mathbf{v}_1 = [0 \quad 1 \quad 0 \quad 0]^\top$.
$X_2^{NF} = \{\delta_4^3\}, \mathbf{v}_2 = [0 \quad 0 \quad 1 \quad 0]^\top$.
$X_3^{NF} = \{\delta_4^3\}, \mathbf{v}_3 = [0 \quad 0 \quad 1 \quad 0]^\top$.
$X_4^{NF} = \{\delta_4^4\}, \mathbf{v}_4 = [0 \quad 0 \quad 0 \quad 1]^\top$.
$X_5^{NF} = \{\delta_4^2\}, \mathbf{v}_5 = [0 \quad 1 \quad 0 \quad 0]^\top$.
$X_6^{NF} = \emptyset$. A fault has occurred.

References

[1] D. Cheng. Input-state approach to Boolean Networks. *IEEE Trans. Neural Networks*, 20 (3):512 – 521, 2009.
[2] D. Cheng and H. Qi. State-space analysis of Boolean Networks. *IEEE Trans. Neural Networks*, 21 (4):584 – 594, 2010.
[3] D. Cheng, H. Qi, and Z. Li. *Analysis and control of Boolean networks*. Springer-Verlag, London, 2011.
[4] X. Ding and P.M. Frank. Fault detection via factorization approach. *Systems & Control Letters*, 14:431–436, 1990.
[5] E. Fornasini and M. E. Valcher. Observability, reconstructibility and state observers of Boolean control networks. *IEEE Tran. Aut. Contr.*, 58 (6):1390 – 1401, 2013.
[6] E. Fornasini and M.E. Valcher. Optimal control of Boolean control networks. *IEEE Trans. Aut. Contr.*, 59 (5): 1258–1270, 2014.
[7] E. Fornasini and M.E. Valcher. Fault detection of Boolean control networks. *Proc. of the 53rd Conference on Decision and Control (CDC2014)*, pp. 6542–6547, Los Angeles (CA), 2014.
[8] P.M. Frank. Fault diagnosis in dynamic systems using analytical and knowledge based redundancy- A survey and some new results. *Automatica*, 26: 459–474, 1990.
[9] H. Fujiwara and S. Toida. The complexity of fault detection problems for combinational logic circuits. *IEEE Trans. Compu.*, 31(6):555–560, 1982.
[10] I. Hwang, S. Kim, Y. Kim, and C. Eng Seah A Survey of Fault Detection, Isolation, and Reconfiguration Methods. *IEEE Trans. Contr. Sys. Tech.*, 18 (3):636–653, 2010.
[11] S.A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theoretical Biology*, 22:437–467, 1969.
[12] D. Laschov and M. Margaliot. A maximum principle for single-input BCNs. *IEEE Trans. Aut. Contr.*, 56 (4):913–917, 2011.
[13] D. Laschov and M. Margaliot. Controllability of BCNs via the Perron-Frobenius theory. *Automatica*, 48:1218–1223, 2012.
[14] R. Layek and A. Datta. Fault detection and intervention in biological feedback networks. *J. Biol. Syst.*, 20:DOI: 10.1142/S0218339012400062, 2012.
[15] H. Li and Y. Wang. Boolean derivative calculation with application to fault detection of combinational circuits via the semi-tensor product method. *Automatica*, 48 (4):688–693, 2012.
[16] H. Li and Y. Wang. Output feedback stabilization control design for Boolean control networks. *Automatica*, 49:3641–3645, 2013.
[17] I. Shmulevich, E.R. Dougherty, S. Kim, and W. Zhang. From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. *Proc. IEEE*, 90 (11):1778–1792, 2002.
[18] S. Sridharan, R. Layek, A.Datta, and J. Venkatraj. Boolean modeling and fault diagnosis in oxidative stress response. *BMC Genomics*, 13, suppl. 6: S4:DOI:10.1186/1471–2164–13–S6–S4PMCID: PMC3481480, 2012.
[19] J.C. Willems. Models for dynamics. *Dynamics report.*, 2:171–269, 1988.