Fault detection of Boolean control networks

M.E. Valcher

University of Padova, Italy

CCC & SICE 2015, Hangzhou, China

Motivations (1)

The interest in Boolean Control Networks (BCNs) is motivated by the large number of physical systems whose describing variables take only 2 values:

Motivations (1)

The interest in Boolean Control Networks (BCNs) is motivated by the large number of physical systems whose describing variables take only 2 values: 1 or 0,

Motivations (1)

The interest in Boolean Control Networks (BCNs) is motivated by the large number of physical systems whose describing variables take only 2 values: 1 or 0, on or off,

Motivations (1)

The interest in Boolean Control Networks (BCNs) is motivated by the large number of physical systems whose describing variables take only 2 values: 1 or 0, on or off, high or low ....

Motivations (1)

The interest in Boolean Control Networks (BCNs) is motivated by the large number of physical systems whose describing variables take only 2 values: 1 or 0, on or off, high or low ....

BCNs have been successfully used to model

Motivations (1)

The interest in Boolean Control Networks (BCNs) is motivated by the large number of physical systems whose describing variables take only 2 values: 1 or 0, on or off, high or low ....

BCNs have been successfully used to model
• the interactions among agents and hence to investigate consensus problems (Wang et al., Automatica 2012)

Motivations (1)

The interest in Boolean Control Networks (BCNs) is motivated by the large number of physical systems whose describing variables take only 2 values: 1 or 0, on or off, high or low ....

BCNs have been successfully used to model
• the interactions among agents and hence to investigate consensus problems (Wang et al., Automatica 2012)
• pursuit evasion problems in polygonal environments (Thunberg et al., ICRA 2011)

Motivations (1)

The interest in Boolean Control Networks (BCNs) is motivated by the large number of physical systems whose describing variables take only 2 values: 1 or 0, on or off, high or low ....

BCNs have been successfully used to model
• the interactions among agents and hence to investigate consensus problems (Wang et al., Automatica 2012)
• pursuit evasion problems in polygonal environments (Thunberg et al., ICRA 2011)
• context-aware systems in smart homes (Kabir et al., ISCE 2014)

Motivations (1)

The interest in Boolean Control Networks (BCNs) is motivated by the large number of physical systems whose describing variables take only 2 values: 1 or 0, on or off, high or low ....

BCNs have been successfully used to model

- the interactions among agents and hence to investigate consensus problems (Wang et al., Automatica 2012)
- pursuit evasion problems in polygonal environments (Thunberg et al., ICRA 2011)
- context-aware systems in smart homes (Kabir et al., ISCE 2014)
- potential games (Cheng, Automatica 2014).

Motivations (2)

The most successful area of application of BCNs however is represented by gene regulatory networks, and in fact, Boolean networks were first introduced in biology (Kauffman, J. Theoretical Biology 1969).

Motivations (2)

The most successful area of application of BCNs however is represented by gene regulatory networks, and in fact, Boolean networks were first introduced in biology (Kauffman, J. Theoretical Biology 1969).

Genes can be modeled as binary devices that exhibit two transcriptional states: active or inactive (“expressed" or not).

Motivations (2)

The most successful area of application of BCNs however is represented by gene regulatory networks, and in fact, Boolean networks were first introduced in biology (Kauffman, J. Theoretical Biology 1969).

Genes can be modeled as binary devices that exhibit two transcriptional states: active or inactive ("expressed" or not).

Even more, the state of a gene depends on the activation status of other genes, and such an interaction can be described by means of logical function.

Motivations (3)

An additional motivation for the interest in these logical networks is represented by the possibility of representing them by means of linear state space models whose state, input and output vectors are canonical vectors:

Motivations (3)

An additional motivation for the interest in these logical networks is represented by the possibility of representing them by means of linear state space models whose state, input and output vectors are canonical vectors: the algebraic representation of BCNs introduced by Daizhan Cheng.

Outline of the talk

- Motivational examples

Outline of the talk

- Motivational examples
- Notation and intro to the algebraic representation of Boolean Control Networks

Outline of the talk

- Motivational examples
- Notation and intro to the algebraic representation of Boolean Control Networks
- Fault detection: the scenario and the possible problems

Outline of the talk

- Motivational examples
- Notation and intro to the algebraic representation of Boolean Control Networks
- Fault detection: the scenario and the possible problems
- On-line fault detection of BCNs

Outline of the talk

- Motivational examples
- Notation and intro to the algebraic representation of Boolean Control Networks
- Fault detection: the scenario and the possible problems
- On-line fault detection of BCNs
- Off-line fault detection of BCNs

Outline of the talk

- Motivational examples
- Notation and intro to the algebraic representation of Boolean Control Networks
- Fault detection: the scenario and the possible problems
- On-line fault detection of BCNs
- Off-line fault detection of BCNs
- Conclusions

Sequential logic circuits (1)

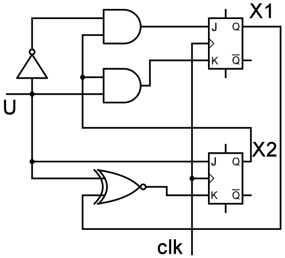
In circuit theory, a sequential logic circuit is a logic circuit whose output depends not only on the present value of its input signal(s) but also on the past value of the input(s).

Sequential logic circuits (1)

In circuit theory, a sequential logic circuit is a logic circuit whose output depends not only on the present value of its input signal(s) but also on the past value of the input(s). So, it is a logic circuit with memory elements (typically realized with flip-flops).

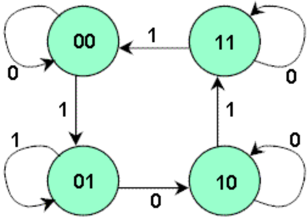
Sequential logic circuits (1)

In circuit theory, a sequential logic circuit is a logic circuit whose output depends not only on the present value of its input signal(s) but also on the past value of the input(s). So, it is a logic circuit with memory elements (typically realized with flip-flops). Consider the following example (M. M. Mano, Digital Design, Prentice Hall, 1984):



Sequential logic circuits (2)

If we denote by $X_1(t)$ and $X_2(t)$ the logical states of the upper and lower flip-flops at time t, and by $U(t)$ the Boolean input at time t, then the update of the pair (X_1, X_2) depending on the value of U follows the state diagram:



Sequential logic circuits (3)

This corresponds to the following Boolean control network:

$$\begin{aligned} X_1(t+1) &= [(X_1(t) \vee X_2(t)) \wedge \bar{U}(t)] \vee [X_1(t) \wedge \bar{X}_2(t) \wedge U(t)], \\ X_2(t+1) &= [(\bar{X}_1(t) \vee \bar{X}_2(t)) \wedge U(t)] \vee [X_1(t) \wedge X_2(t) \wedge \bar{U}(t)]. \end{aligned}$$

Assume that this circuit is part of a complex logical network whose output value Y depends at every time t from the variables $X_1(t)$ and $X_2(t)$ and from other logical state variables.

Sequential logic circuits (3)

This corresponds to the following Boolean control network:

$$\begin{aligned} X_1(t+1) &= [(X_1(t) \vee X_2(t)) \wedge \bar{U}(t)] \vee [X_1(t) \wedge \bar{X}_2(t) \wedge U(t)], \\ X_2(t+1) &= [(\bar{X}_1(t) \vee \bar{X}_2(t)) \wedge U(t)] \vee [X_1(t) \wedge X_2(t) \wedge \bar{U}(t)]. \end{aligned}$$

Assume that this circuit is part of a complex logical network whose output value Y depends at every time t from the variables $X_1(t)$ and $X_2(t)$ and from other logical state variables. A typical problem arising in this kind of circuits is the so-called stuck-in faults: one of logical variables is stuck at 0 or 1, independently of the input sequence.

Sequential logic circuits (3)

This corresponds to the following Boolean control network:

$$\begin{aligned} X_1(t+1) &= [(X_1(t) \vee X_2(t)) \wedge \bar{U}(t)] \vee [X_1(t) \wedge \bar{X}_2(t) \wedge U(t)], \\ X_2(t+1) &= [(\bar{X}_1(t) \vee \bar{X}_2(t)) \wedge U(t)] \vee [X_1(t) \wedge X_2(t) \wedge \bar{U}(t)]. \end{aligned}$$

Assume that this circuit is part of a complex logical network whose output value Y depends at every time t from the variables $X_1(t)$ and $X_2(t)$ and from other logical state variables. A typical problem arising in this kind of circuits is the so-called stuck-in faults: one of logical variables is stuck at 0 or 1, independently of the input sequence.

How can we model these faults?

Sequential logic circuits (3)

This corresponds to the following Boolean control network:

$$\begin{aligned} X_1(t+1) &= [(X_1(t) \vee X_2(t)) \wedge \bar{U}(t)] \vee [X_1(t) \wedge \bar{X}_2(t) \wedge U(t)], \\ X_2(t+1) &= [(\bar{X}_1(t) \vee \bar{X}_2(t)) \wedge U(t)] \vee [X_1(t) \wedge X_2(t) \wedge \bar{U}(t)]. \end{aligned}$$

Assume that this circuit is part of a complex logical network whose output value Y depends at every time t from the variables $X_1(t)$ and $X_2(t)$ and from other logical state variables. A typical problem arising in this kind of circuits is the so-called stuck-in faults: one of logical variables is stuck at 0 or 1, independently of the input sequence.

How can we model these faults?
How can we detect such faults?

Oxidative stress response (1)

Oxidative stress is caused by exposure to reactive oxygen species (ROS) (electrophiles and oxidants).

Oxidative stress response (1)

Oxidative stress is caused by exposure to reactive oxygen species (ROS) (electrophiles and oxidants). Since oxidative stress contributes to aging and age-related diseases (cancer, cardiovascular disease, chronic inflammation, and neurodegenerative disorders), the body has developed a number of counteractive measures to counterbalance it.

Oxidative stress response (1)

Oxidative stress is caused by exposure to reactive oxygen species (ROS) (electrophiles and oxidants). Since oxidative stress contributes to aging and age-related diseases (cancer, cardiovascular disease, chronic inflammation, and neurodegenerative disorders), the body has developed a number of counteractive measures to counterbalance it.

Indeed, when the concentration of electrophiles is high, the complex Keap1-Nrf2 (Keap1 is a sensor, while Nrf2 is a transcription factor) is broken and Nrf2 is liberated and transported into the nucleus.

Oxidative stress response (1)

Oxidative stress is caused by exposure to reactive oxygen species (ROS) (electrophiles and oxidants). Since oxidative stress contributes to aging and age-related diseases (cancer, cardiovascular disease, chronic inflammation, and neurodegenerative disorders), the body has developed a number of counteractive measures to counterbalance it.

Indeed, when the concentration of electrophiles is high, the complex Keap1-Nrf2 (Keap1 is a sensor, while Nrf2 is a transcription factor) is broken and Nrf2 is liberated and transported into the nucleus.

As a countermeasure, various antioxidant response elements (ARE) are activated.

Oxidative stress response (2)

Inside the nucleus, Nrf2 forms heterodimers with small Maf proteins (SMP) which then bind to the ARE and lead to the production of detoxifying enzymes.

Oxidative stress response (2)

Inside the nucleus, Nrf2 forms heterodimers with small Maf proteins (SMP) which then bind to the ARE and lead to the production of detoxifying enzymes.
Once the electrophiles have been eliminated, these protein complexes and other proteins named PKC and Bach1 bind to the ARE, and Nrf2 is transported back to the cytoplasm, where it binds with Keap1.

Oxidative stress response (2)

Inside the nucleus, Nrf2 forms heterodimers with small Maf proteins (SMP) which then bind to the ARE and lead to the production of detoxifying enzymes.
Once the electrophiles have been eliminated, these protein complexes and other proteins named PKC and Bach1 bind to the ARE, and Nrf2 is transported back to the cytoplasm, where it binds with Keap1.
In [Sridharan et al., BMC Genomics 2012] the following Boolean control network has been proposed for the oxidative stress response:

Oxidative stress response (3)

$$
\begin{aligned}
X_1(t+1) &= \bar{X}_6(t) \wedge U(t), \\
X_2(t+1) &= \bar{X}_1(t) \wedge [X_2(t) \vee X_4(t)], \\
X_3(t+1) &= X_1(t) \wedge \bar{X}_6(t), \\
X_4(t+1) &= \bar{X}_2(t) \vee X_3(t), \\
X_5(t+1) &= \bar{X}_1(t), \\
X_6(t+1) &= X_4(t) \wedge [\bar{X}_5(t) \vee \bar{X}_6(t)],
\end{aligned}
$$

Oxidative stress response (3)

$$
\begin{aligned}
X_1(t+1) &= \bar{X}_6(t) \wedge U(t), \\
X_2(t+1) &= \bar{X}_1(t) \wedge [X_2(t) \vee X_4(t)], \\
X_3(t+1) &= X_1(t) \wedge \bar{X}_6(t), \\
X_4(t+1) &= \bar{X}_2(t) \vee X_3(t), \\
X_5(t+1) &= \bar{X}_1(t), \\
X_6(t+1) &= X_4(t) \wedge [\bar{X}_5(t) \vee \bar{X}_6(t)],
\end{aligned}
$$

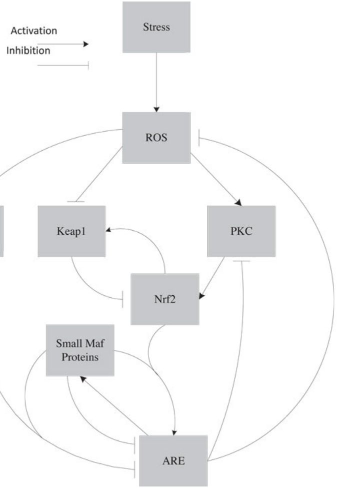
where $X_1 = ROS,\ X_2 = Keap1,\ X_3 = PKC,\ X_4 = Nrf2,\ X_5 = Bach1,\ X_6 = ARE,\ U = Stress$, and small Maf proteins are always assumed to be expressed (SMP=1).

Oxidative stress response (3)

$$
\begin{aligned}
X_1(t+1) &= \bar{X}_6(t) \wedge U(t), \\
X_2(t+1) &= \bar{X}_1(t) \wedge [X_2(t) \vee X_4(t)], \\
X_3(t+1) &= X_1(t) \wedge \bar{X}_6(t), \\
X_4(t+1) &= \bar{X}_2(t) \vee X_3(t), \\
X_5(t+1) &= \bar{X}_1(t), \\
X_6(t+1) &= X_4(t) \wedge [\bar{X}_5(t) \vee \bar{X}_6(t)],
\end{aligned}
$$

where $X_1 = ROS,\ X_2 = Keap1,\ X_3 = PKC,\ X_4 = Nrf2,\ X_5 = Bach1,\ X_6 = ARE,\ U = Stress$, and small Maf proteins are always assumed to be expressed (SMP=1).

Oxidative stress response (4)



Oxidative stress response (5)

The oxidative response pathway just illustrated interacts with the PI3k/Akt Pathway. A complete model of how the two pathways interact can be found in [Sridharan et al., BMC Genomics 2012].

Oxidative stress response (5)

The oxidative response pathway just illustrated interacts with the PI3k/Akt Pathway. A complete model of how the two pathways interact can be found in [Sridharan et al., BMC Genomics 2012]. The important thing to remark is that a fault in the oxidative response pathway may lead to a fault in the PI3k/Akt Pathway, whose malfunctioning has been related to various forms of cancer.

Oxidative stress response (5)

The oxidative response pathway just illustrated interacts with the PI3k/Akt Pathway. A complete model of how the two pathways interact can be found in [Sridharan et al., BMC Genomics 2012]. The important thing to remark is that a fault in the oxidative response pathway may lead to a fault in the PI3k/Akt Pathway, whose malfunctioning has been related to various forms of cancer.

For these reasons it is important to understand:

Oxidative stress response (5)

The oxidative response pathway just illustrated interacts with the PI3k/Akt Pathway. A complete model of how the two pathways interact can be found in [Sridharan et al., BMC Genomics 2012]. The important thing to remark is that a fault in the oxidative response pathway may lead to a fault in the PI3k/Akt Pathway, whose malfunctioning has been related to various forms of cancer.

For these reasons it is important to understand:
How can one detect a fault in the oxidative stress response pathway?

Oxidative stress response (5)

The oxidative response pathway just illustrated interacts with the PI3k/Akt Pathway. A complete model of how the two pathways interact can be found in [Sridharan et al., BMC Genomics 2012]. The important thing to remark is that a fault in the oxidative response pathway may lead to a fault in the PI3k/Akt Pathway, whose malfunctioning has been related to various forms of cancer.

For these reasons it is important to understand:

How can one detect a fault in the oxidative stress response pathway?

What are the output (measures) we can take, and how can we use them to detect a fault?

Notation (1)

- $\mathcal{B} := \{0, 1\}$ is the set where Boolean variables take values.

Notation (1)

• $\mathcal{B} := \{0, 1\}$ is the set where Boolean variables take values. On $\mathcal{B}$ we define the usual operations: sum $\vee$, product $\wedge$ and negation $\bar{\cdot}$

Notation (1)

- $\mathcal{B} := \{0, 1\}$ is the set where Boolean variables take values. On $\mathcal{B}$ we define the usual operations: sum $\vee$, product $\wedge$ and negation $\bar{\cdot}$

- δ_k^i is the ith canonical vector of size k

Notation (1)

- $\mathcal{B} := \{0, 1\}$ is the set where Boolean variables take values. On $\mathcal{B}$ we define the usual operations: sum $\vee$, product $\wedge$ and negation $\bar{\cdot}$

- δ_k^i is the ith canonical vector of size k

- $\mathcal{L}_{k\times n}$ is the set of $k \times n$ logical matrices whose n columns are canonical vectors of size k.

Notation (1)

- $\mathcal{B} := \{0, 1\}$ is the set where Boolean variables take values. On $\mathcal{B}$ we define the usual operations: sum $\vee$, product $\wedge$ and negation $\bar{\cdot}$

- δ_k^i is the ith canonical vector of size k

- $\mathcal{L}_{k\times n}$ is the set of $k \times n$ logical matrices whose n columns are canonical vectors of size k. (We set $\mathcal{L}_k := \mathcal{L}_{k\times 1}$)

Notation (2)

- The semi-tensor product $\ltimes$ between matrices $L_1 \in \mathcal{B}_{r_1 \times c_1}$ and $L_2 \in \mathcal{B}_{r_2 \times c_2}$ is defined as

Notation (2)

- The semi-tensor product $\ltimes$ between matrices $L_1 \in \mathcal{B}_{r_1 \times c_1}$ and $L_2 \in \mathcal{B}_{r_2 \times c_2}$ is defined as

$$L_1 \ltimes L_2 := (L_1 \otimes I_{T/c_1})(L_2 \otimes I_{T/r_2}), \qquad T := \mathrm{l.c.m.}\{c_1, r_2\}.$$

Notation (2)

- The semi-tensor product $\ltimes$ between matrices $L_1 \in \mathcal{B}_{r_1 \times c_1}$ and $L_2 \in \mathcal{B}_{r_2 \times c_2}$ is defined as

$$L_1 \ltimes L_2 := (L_1 \otimes I_{T/c_1})(L_2 \otimes I_{T/r_2}), \qquad T := \mathrm{l.c.m.}\{c_1, r_2\}.$$

- There is a bijective correspondence between $\mathcal{B}$ and $\mathcal{L}_2$:

Notation (2)

- The semi-tensor product $\ltimes$ between matrices $L_1 \in \mathcal{B}_{r_1 \times c_1}$ and $L_2 \in \mathcal{B}_{r_2 \times c_2}$ is defined as

$$L_1 \ltimes L_2 := (L_1 \otimes I_{T/c_1})(L_2 \otimes I_{T/r_2}), \qquad T := \text{l.c.m.}\{c_1, r_2\}.$$

- There is a bijective correspondence between $\mathcal{B}$ and $\mathcal{L}_2$:

$$X = 1 \longleftrightarrow \mathbf{x} = X^v = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad X = 0 \longleftrightarrow \mathbf{x} = X^v = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Notation (2)

- The semi-tensor product $\ltimes$ between matrices $L_1 \in \mathcal{B}_{r_1 \times c_1}$ and $L_2 \in \mathcal{B}_{r_2 \times c_2}$ is defined as

$$L_1 \ltimes L_2 := (L_1 \otimes I_{T/c_1})(L_2 \otimes I_{T/r_2}), \qquad T := \text{l.c.m.}\{c_1, r_2\}.$$

- There is a bijective correspondence between $\mathcal{B}$ and $\mathcal{L}_2$:

$$X = 1 \longleftrightarrow \mathbf{x} = X^v = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad X = 0 \longleftrightarrow \mathbf{x} = X^v = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

It extends to a bijective correspondence between $\mathcal{B}^n$ and $\mathcal{L}_{2^n}$ through:

$$X = \begin{bmatrix} X_1 & X_2 & \dots & X_n \end{bmatrix}^\top \longleftrightarrow \mathbf{x} = X_1^v \ltimes X_2^v \ltimes \dots \ltimes X_n^v.$$

BCNs: from logic to algebraic representations (1)

A Boolean control network (BCN) is described by the following equations

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

BCNs: from logic to algebraic representations (1)

A Boolean control network (BCN) is described by the following equations

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

$X(\cdot)$, $U(\cdot)$ and $Y(\cdot)$ are the Boolean state (dim = n), input (dim = m), and output (dim = p).

BCNs: from logic to algebraic representations (1)

A Boolean control network (BCN) is described by the following equations

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

$X(\cdot)$, $U(\cdot)$ and $Y(\cdot)$ are the Boolean state (dim = n), input (dim = m), and output (dim = p). f and h are logic functions.

BCNs: from logic to algebraic representations (1)

A Boolean control network (BCN) is described by the following equations

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

$X(\cdot)$, $U(\cdot)$ and $Y(\cdot)$ are the Boolean state (dim = n), input (dim = m), and output (dim = p). f and h are logic functions.

Algebraic representation [D. Cheng]: if we represent the Boolean vectors by means of their “canonical equivalent", the BCN (1) can be described as

$$\begin{aligned} \mathbf{x}(t+1) &= L \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \qquad t \in \mathbb{Z}_+, \\ \mathbf{y}(t) &= H\mathbf{x}(t) \end{aligned} \tag{2}$$

where $L \in \mathcal{L}_{2^n \times 2^{(n+m)}}$ and $H \in \mathcal{L}_{2^p \times 2^n}$.

BCNs: from logic to algebraic representations (1)

A Boolean control network (BCN) is described by the following equations

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

$X(\cdot)$, $U(\cdot)$ and $Y(\cdot)$ are the Boolean state (dim = n), input (dim = m), and output (dim = p). f and h are logic functions.

Algebraic representation [D. Cheng]: if we represent the Boolean vectors by means of their “canonical equivalent", the BCN (1) can be described as

$$\begin{aligned} \mathbf{x}(t+1) &= L \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \qquad t \in \mathbb{Z}_+, \\ \mathbf{y}(t) &= H\mathbf{x}(t) \end{aligned} \tag{2}$$

where $L \in \mathcal{L}_{2^n \times 2^{(n+m)}}$ and $H \in \mathcal{L}_{2^p \times 2^n}$. In the following

$$N := 2^n, \qquad M := 2^m, \qquad P := 2^p.$$

The scenario (1)

The general problem: Given a BCN, we want to investigate the problem of determining, from its input and output trajectories (but no access to the state), whether a fault has affected the BCN functioning or not.

The scenario (1)

The general problem: Given a BCN, we want to investigate the problem of determining, from its input and output trajectories (but no access to the state), whether a fault has affected the BCN functioning or not.

What do we mean by a fault and what may be the outcome of a fault?

The scenario (1)

The general problem: Given a BCN, we want to investigate the problem of determining, from its input and output trajectories (but no access to the state), whether a fault has affected the BCN functioning or not.

What do we mean by a fault and what may be the outcome of a fault?

Assumptions:
A1) The BCN exhibits only two possible configurations:

The scenario (1)

The general problem: Given a BCN, we want to investigate the problem of determining, from its input and output trajectories (but no access to the state), whether a fault has affected the BCN functioning or not.

What do we mean by a fault and what may be the outcome of a fault?

Assumptions:
A1) The BCN exhibits only two possible configurations:
a non-faulty (NF) and a faulty (F) one.

The scenario (1)

The general problem: Given a BCN, we want to investigate the problem of determining, from its input and output trajectories (but no access to the state), whether a fault has affected the BCN functioning or not.

What do we mean by a fault and what may be the outcome of a fault?

Assumptions:
A1) The BCN exhibits only two possible configurations:
a non-faulty (NF) and a faulty (F) one.
A2) The fault affects only the state-update, not the output measurements.

The scenario (2)

Accordingly, we represent the non-faulty BCN as in (2) and the faulty one as

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

with $L^{(F)} \in \mathcal{L}_{N \times NM}$.

The scenario (2)

Accordingly, we represent the non-faulty BCN as in (2) and the faulty one as

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

with $L^{(F)} \in \mathcal{L}_{N \times NM}$.

We introduce the fault signal $(\mathbf{f}(t))_{t\in\mathbb{Z}_+}$, taking values in $\mathcal{L}_2$, and 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 scenario (2)

Accordingly, we represent the non-faulty BCN as in (2) and the faulty one as

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

with $L^{(F)} \in \mathcal{L}_{N \times NM}$.

We introduce the fault signal $(\mathbf{f}(t))_{t \in \mathbb{Z}_+}$, taking values in $\mathcal{L}_2$, and 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. So, the overall BCN dynamics is

$$\begin{aligned} \mathbf{x}(t+1) &= \tilde{L} \ltimes \mathbf{f}(t) \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \\ \mathbf{y}(t) &= H \ltimes \mathbf{x}(t) = H\mathbf{x}(t), \qquad t \in \mathbb{Z}_+. \end{aligned} \tag{4}$$

where $\tilde{L} := \begin{bmatrix} L & L^{(F)} \end{bmatrix} \in \mathcal{L}_{N \times 2NM}$.

The scenario (3)

Another assumption:

The scenario (3)

Another assumption: A3) the BCN cannot autonomously recover from a fault:

The scenario (3)

Another assumption: A3) the BCN cannot autonomously recover from a fault: once the fault signal switches from δ_2^1 to δ_2^2, it cannot switch back to δ_2^1.

The scenario (3)

Another assumption: A3) the BCN cannot autonomously recover from a fault: once the fault signal switches from δ_2^1 to δ_2^2, it cannot switch back to δ_2^1. So, a fault acting at time $\bar{t}$ is described by a step function

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

where $\bar{t} = +\infty$ in case no fault affects the BCN.

Meaningful fault (1)

Let $\mathbf{x}(t; \mathbf{x}_0, \mathbf{u}(\cdot), \mathbf{f}(\cdot))$ and $\mathbf{y}(t; \mathbf{x}_0, \mathbf{u}(\cdot), \mathbf{f}(\cdot))$ denote the state and output vectors of the BCN (4) 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.

Meaningful fault (1)

Let $\mathbf{x}(t; \mathbf{x}_0, \mathbf{u}(\cdot), \mathbf{f}(\cdot))$ and $\mathbf{y}(t; \mathbf{x}_0, \mathbf{u}(\cdot), \mathbf{f}(\cdot))$ denote the state and output vectors of the BCN (4) 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.

A fault taking place at time $\bar{t}$, for certain values of $\bar{\mathbf{x}} := \mathbf{x}(\bar{t}) \in \mathcal{L}_N$ and $\mathbf{u}(t), t \geq \bar{t}$, may not reveal itself, independently of how we choose the output measurements.

Meaningful fault (1)

Let $\mathbf{x}(t; \mathbf{x}_0, \mathbf{u}(\cdot), \mathbf{f}(\cdot))$ and $\mathbf{y}(t; \mathbf{x}_0, \mathbf{u}(\cdot), \mathbf{f}(\cdot))$ denote the state and output vectors of the BCN (4) 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.

A fault taking place at time $\bar{t}$, for certain values of $\bar{\mathbf{x}} := \mathbf{x}(\bar{t}) \in \mathcal{L}_N$ and $\mathbf{u}(t), t \geq \bar{t}$, may not reveal itself, independently of how we choose the output measurements.

Indeed, the state trajectory generated by the faulty BCN (3) starting from $\bar{\mathbf{x}}$ at $t = \bar{t}$, under the effect of $\mathbf{u}(\cdot)$, may coincide with the state trajectory that the non-faulty BCN (2) generates in the same conditions.

Meaningful fault (2)

This is not unreasonable, as the faulty part of the system may not involved in the dynamic evolution and hence the fault cannot be detected.

Meaningful fault (2)

This is not unreasonable, as the faulty part of the system may not involved in the dynamic evolution and hence the fault cannot be detected.

Definition 1 Given a state $\mathbf{x}_0 \in \mathcal{L}_N$ and an input $(\mathbf{u}(t))_{t\in\mathbb{Z}_+}$,

Meaningful fault (2)

This is not unreasonable, as the faulty part of the system may not involved in the dynamic evolution and hence the fault cannot be detected.

Definition 1 Given a state $\mathbf{x}_0 \in \mathcal{L}_N$ and an input $(\mathbf{u}(t))_{t\in\mathbb{Z}_+}$, a fault sequence $(\mathbf{f}(t))_{t\in\mathbb{Z}_+}$ induces a meaningful fault for the BCN if the state trajectory $(\mathbf{x}(t;\mathbf{x}_0,\mathbf{u}(\cdot),\mathbf{f}(\cdot)))_{t\in\mathbb{Z}_+}$ is different from the state trajectory $(\mathbf{x}(t;\mathbf{x}_0,\mathbf{u}(\cdot),\,\delta_2^1))_{t\in\mathbb{Z}_+}$.

Meaningful fault (2)

This is not unreasonable, as the faulty part of the system may not involved in the dynamic evolution and hence the fault cannot be detected.

Definition 1 Given a state $\mathbf{x}_0 \in \mathcal{L}_N$ and an input $(\mathbf{u}(t))_{t\in\mathbb{Z}_+}$, a fault sequence $(\mathbf{f}(t))_{t\in\mathbb{Z}_+}$ induces a meaningful fault for the BCN if the state trajectory $(\mathbf{x}(t;\mathbf{x}_0,\mathbf{u}(\cdot),\mathbf{f}(\cdot)))_{t\in\mathbb{Z}_+}$ is different from the state trajectory $(\mathbf{x}(t;\mathbf{x}_0,\mathbf{u}(\cdot),\,\delta_2^1))_{t\in\mathbb{Z}_+}$.

Meaningful fault sequences are the only ones we may hope to detect, by making use of the input and output trajectories.

Possible problems

On-line fault detection: the BCN, undergoing normal working conditions, is subject to an arbitrary input, and we want to understand, based on its input-output behavior, if a fault has occurred.

Possible problems

On-line fault detection: the BCN, undergoing normal working conditions, is subject to an arbitrary input, and we want to understand, based on its input-output behavior, if a fault has occurred.

Off-line fault detection: an ad hoc off-line test is performed on the BCN, to ascertain whether it is faulty or it is working correctly.

Possible problems

On-line fault detection: the BCN, undergoing normal working conditions, is subject to an arbitrary input, and we want to understand, based on its input-output behavior, if a fault has occurred.

Off-line fault detection: an ad hoc off-line test is performed on the BCN, to ascertain whether it is faulty or it is working correctly. In other words, we want to detect whether the BCN is faulty or not, by applying a fixed finite support input test sequence, independent of the initial condition.

Possible problems

On-line fault detection: the BCN, undergoing normal working conditions, is subject to an arbitrary input, and we want to understand, based on its input-output behavior, if a fault has occurred.

Off-line fault detection: an ad hoc off-line test is performed on the BCN, to ascertain whether it is faulty or it is working correctly. In other words, we want to detect whether the BCN is faulty or not, by applying a fixed finite support input test sequence, independent of the initial condition. In this latter case, we assume that the BCN is either working correctly or erroneously during the whole duration of the test, namely $\mathbf{f}(t), t \in \mathbb{Z}_+,$ is either identically equal to δ_2^1 or to δ_2^2.

On-line fault detection: Preliminaries

We introduce the set of the admissible input/output trajectories of the non-faulty BCN:

$$\begin{aligned} \mathfrak{B}_{uy} \;:=\; & \{(\mathbf{u}(t), \mathbf{y}(t))_{t\in\mathbb{Z}_+} : (\mathbf{u}(t))_{t\in\mathbb{Z}_+} \in (\mathcal{L}_M)^{\mathbb{Z}_+}, \text{and} \\ & \exists \mathbf{x}_0 \in \mathcal{L}_N \text{ s.t. } \mathbf{y}(t) = \mathbf{y}(t; \mathbf{x}_0, \mathbf{u}(\cdot), \delta_2^1)\}. \end{aligned} \tag{5}$$

On-line fault detection: Preliminaries

We introduce the set of the admissible input/output trajectories of the non-faulty BCN:

$$\begin{aligned} \mathfrak{B}_{uy} \;:=\; & \{(\mathbf{u}(t), \mathbf{y}(t))_{t\in\mathbb{Z}_+} : (\mathbf{u}(t))_{t\in\mathbb{Z}_+} \in (\mathcal{L}_M)^{\mathbb{Z}_+}, \text{and} \\ & \exists \mathbf{x}_0 \in \mathcal{L}_N \text{ s.t. } \mathbf{y}(t) = \mathbf{y}(t; \mathbf{x}_0, \mathbf{u}(\cdot), \delta_2^1)\}. \end{aligned} \tag{5}$$

$\mathfrak{B}_{uy}$ 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 (2) corresponding to some initial state $\mathbf{x}(0) = \mathbf{x}_0 \in \mathcal{L}_N$ and to the input $(\mathbf{u}(t))_{t\in\mathbb{Z}_+}$.

Detectability of a meaningful fault (1)

Definition 2 Given a BCN (4), a state $\mathbf{x}_0 \in \mathcal{L}_N$, an input $(\mathbf{u}(t))_{t\in\mathbb{Z}_+}$, and a (meaningful) fault sequence $(\mathbf{f}(t))_{t\in\mathbb{Z}_+}$,

Detectability of a meaningful fault (1)

Definition 2 Given a BCN (4), a state $\mathbf{x}_0 \in \mathcal{L}_N$, an input $(\mathbf{u}(t))_{t\in\mathbb{Z}_+}$, 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 (4) does not belong to $\mathfrak{B}_{uy}$.

Detectability of a meaningful fault (2)

We define:

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

Detectability of a meaningful fault (2)

We define:

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

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

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

Detectability of a meaningful fault (2)

We define:

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

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

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

Finally, $\mathcal{UY}^*$ 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 (3) corresponding to some $\mathbf{x}(0) = \mathbf{x}^* \in X^*$ and to some input $(\mathbf{u}(t))_{t\in\mathbb{Z}_+}$ with $\mathbf{u}(0) \in U^*(\mathbf{x}^*)$.

Detectability of a meaningful fault (3)

Proposition 1 The following facts are equivalent:

Detectability of a meaningful fault (3)

Proposition 1 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 (the on-line fault detection problem is solvable);

Detectability of a meaningful fault (3)

Proposition 1 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 (the on-line fault detection problem is solvable);
ii) $\mathcal{U}\mathcal{Y}^* \cap \mathfrak{B}_{uy} = \emptyset$.

Graph-theoretic characterization (1)

The idea: 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:

Graph-theoretic characterization (1)

The idea: 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: the NF-F (non-faulty-faulty) directed graph.

Graph-theoretic characterization (2)

The NF-F (non-faulty-faulty) directed graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, C_0, C_1)$, where

Graph-theoretic characterization (2)

The NF-F (non-faulty-faulty) directed graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, C_0, C_1)$, where

- The vertex set $\mathcal{V}$ is the set of all pairs of states $\{(\delta_N^i, \delta_N^j) \in \mathcal{L}_N \times \mathcal{L}_N\}$.

Graph-theoretic characterization (2)

The NF-F (non-faulty-faulty) directed graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, C_0, C_1)$, where

- The vertex set $\mathcal{V}$ is the set of all pairs of states $\{(\delta_N^i, \delta_N^j) \in \mathcal{L}_N \times \mathcal{L}_N\}$.
- The labeled edge set $\mathcal{E}$ is defined as follows:

Graph-theoretic characterization (2)

The NF-F (non-faulty-faulty) directed graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, C_0, C_1)$, where

- The vertex set $\mathcal{V}$ is the set of all pairs of states $\{(\delta_N^i, \delta_N^j) \in \mathcal{L}_N \times \mathcal{L}_N\}$.
- The labeled edge set $\mathcal{E}$ is defined as follows: there is an edge labeled by $\mathbf{u} \in \mathcal{L}_M$ from (δ_N^i, δ_N^j) to (δ_N^h, δ_N^k) if and only if

$$\delta_N^h = L \ltimes \mathbf{u} \ltimes \delta_N^i \text{ and } \delta_N^k = L^{(F)} \ltimes \mathbf{u} \ltimes \delta_N^j.$$

Graph-theoretic characterization (2)

The NF-F (non-faulty-faulty) directed graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, C_0, C_1)$, where

- The vertex set $\mathcal{V}$ is the set of all pairs of states $\{(\delta_N^i, \delta_N^j) \in \mathcal{L}_N \times \mathcal{L}_N\}$.
- The labeled edge set $\mathcal{E}$ is defined as follows: there is an edge labeled by $\mathbf{u} \in \mathcal{L}_M$ from (δ_N^i, δ_N^j) to (δ_N^h, δ_N^k) if and only if

  $$\delta_N^h = L \ltimes \mathbf{u} \ltimes \delta_N^i \text{ and } \delta_N^k = L^{(F)} \ltimes \mathbf{u} \ltimes \delta_N^j.$$

  From every pair (δ_N^i, δ_N^j) there are M outgoing arcs, one for each value of the input $\mathbf{u}$.

Graph-theoretic characterization (2)

The NF-F (non-faulty-faulty) directed graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, C_0, C_1)$, where

- The vertex set $\mathcal{V}$ is the set of all pairs of states $\{(\delta_N^i, \delta_N^j) \in \mathcal{L}_N \times \mathcal{L}_N\}$.
- The labeled edge set $\mathcal{E}$ is defined as follows: there is an edge labeled by $\mathbf{u} \in \mathcal{L}_M$ from (δ_N^i, δ_N^j) to (δ_N^h, δ_N^k) if and only if
$$\delta_N^h = L \ltimes \mathbf{u} \ltimes \delta_N^i \text{ and } \delta_N^k = L^{(F)} \ltimes \mathbf{u} \ltimes \delta_N^j.$$
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.

Graph-theoretic characterization (2)

The NF-F (non-faulty-faulty) directed graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, C_0, C_1)$, where

- The vertex set $\mathcal{V}$ is the set of all pairs of states $\{(\delta_N^i, \delta_N^j) \in \mathcal{L}_N \times \mathcal{L}_N\}$.
- The labeled edge set $\mathcal{E}$ is defined as follows: there is an edge labeled by $\mathbf{u} \in \mathcal{L}_M$ from (δ_N^i, δ_N^j) to (δ_N^h, δ_N^k) if and only if

  $$\delta_N^h = L \ltimes \mathbf{u} \ltimes \delta_N^i \text{ and } \delta_N^k = L^{(F)} \ltimes \mathbf{u} \ltimes \delta_N^j.$$

  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. (δ_N^i, δ_N^j) belongs to C_1 if $H\delta_N^i = H\delta_N^j$,

Graph-theoretic characterization (2)

The NF-F (non-faulty-faulty) directed graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, C_0, C_1)$, where

- The vertex set $\mathcal{V}$ is the set of all pairs of states $\{(\delta_N^i, \delta_N^j) \in \mathcal{L}_N \times \mathcal{L}_N\}$.
- The labeled edge set $\mathcal{E}$ is defined as follows: there is an edge labeled by $\mathbf{u} \in \mathcal{L}_M$ from (δ_N^i, δ_N^j) to (δ_N^h, δ_N^k) if and only if

  $$\delta_N^h = L \ltimes \mathbf{u} \ltimes \delta_N^i \text{ and } \delta_N^k = L^{(F)} \ltimes \mathbf{u} \ltimes \delta_N^j.$$

  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. (δ_N^i, δ_N^j) belongs to C_1 if $H\delta_N^i = H\delta_N^j$, otherwise it belongs to C_0.

Graph-theoretic characterization (3)

Proposition 2 Given the BCN (4), let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, C_0, C_1)$ be the associated NF-F directed graph. The on-line fault detection problem is solvable if and only if

Graph-theoretic characterization (3)

Proposition 2 Given the BCN (4), let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, C_0, C_1)$ be the associated NF-F directed graph. The on-line fault detection problem is solvable 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 $\mathbf{u}^* \in U^*(\mathbf{x}^*)$;

Graph-theoretic characterization (3)

Proposition 2 Given the BCN (4), let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, C_0, C_1)$ be the associated NF-F directed graph. The on-line fault detection problem is solvable 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 $\mathbf{u}^* \in U^*(\mathbf{x}^*)$;

enters the class C_0 in a finite number of steps.

Off-line fault detection: Preliminaries

Proposition 3 The following facts are equivalent:
i) the off-line fault detection problem is solvable;
ii) there exist $T \in \mathbb{Z}_+$ and an input $\hat{\mathbf{u}}(t), t \in [0, T-1]$, taking values in $\mathcal{L}_M$, such that the two sets of output trajectories

$$\begin{aligned}
\hat{\mathcal{Y}}|_{[0,T]} &:= \{(\mathbf{y}(t))|_{[0,t]} : \exists\, \mathbf{x}_0 \in \mathcal{L}_N \text{ s.t.} \\
&\qquad \mathbf{y}(t) = \mathbf{y}(t; \mathbf{x}_0, \hat{\mathbf{u}}(\cdot), \delta_2^1), \forall\, t \in [0, T]\} \\
\hat{\mathcal{Y}}^{(F)}|_{[0,T]} &:= \{(\mathbf{y}(t))|_{[0,t]} : \exists\, \mathbf{x}_0 \in \mathcal{L}_N \text{ s.t.} \\
&\qquad \mathbf{y}(t) = \mathbf{y}(t; \mathbf{x}_0, \hat{\mathbf{u}}(\cdot), \delta_2^2), \forall\, t \in [0, T]\}
\end{aligned}$$

are disjoint.

Graph-theoretic characterization

Proposition 4 The following facts are equivalent:
i) the off-line fault detection problem is solvable;

Graph-theoretic characterization

Proposition 4 The following facts are equivalent:
i) the off-line fault detection problem is solvable;
ii) for every vertex $v = (\delta_N^i, \delta_N^j) \in C_1$ there is path in the NF-F graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, C_0, C_1)$ from v to some vertex belonging to C_0.

Algorithm (1)

The previous analysis determined the conditions under which we can solve the two problems.

Algorithm (1)

The previous analysis determined the conditions under which we can solve the two problems.

But, assuming that such conditions hold, how do we practically detect whether a fault occurred in either one of the previous set-ups?

Algorithm (1)

The previous analysis determined the conditions under which we can solve the two problems.
But, assuming that such conditions hold, how do we practically detect whether a fault occurred in either one of the previous set-ups?

Let $\mathbf{X}_{\tau}^{NF}$ be the set of the states that the non-faulty BCN (2) can reach at time τ, under the effect of the input sequence $\mathbf{u}(t), t \in [0, \tau - 1]$, meanwhile generating the output $\mathbf{y}(t), t \in [0, \tau]$.

Algorithm (1)

The previous analysis determined the conditions under which we can solve the two problems.
But, assuming that such conditions hold, how do we practically detect whether a fault occurred in either one of the previous set-ups?

Let $\mathbf{X}_\tau^{NF}$ be the set of the states that the non-faulty BCN (2) can reach at time τ, under the effect of the input sequence $\mathbf{u}(t), t \in [0, \tau - 1]$, meanwhile generating the output $\mathbf{y}(t), t \in [0, \tau]$.

A fault has occurred if and only if there is τ such that $\mathbf{X}_\tau^{NF} = \emptyset$.

Algorithm (2)

In other words,

Algorithm (2)

In other words,
one starts at time $\tau = 0$ by determining the set $\mathbf{X}_0^{NF}$ of the initial states compatible with $\mathbf{y}(0)$.

Algorithm (2)

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 $\mathbf{u}(0)$.

Algorithm (2)

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.

Algorithm (2)

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.

Note: at every t, the state δ_N^i is compatible with a given output sample $\mathbf{y}(t) \in \mathcal{L}_P$ if and only if $[H^\top \mathbf{y}(t)]_i = 1$.

Algorithm (3)

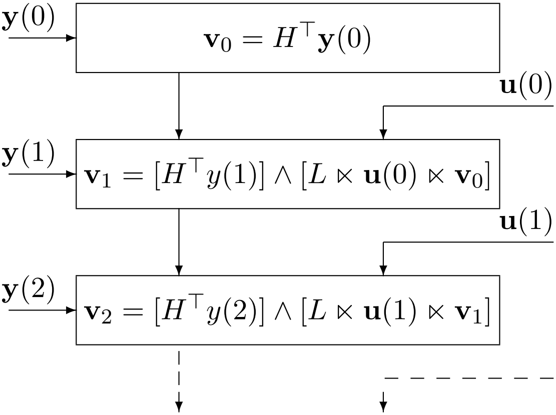


Fig. 1: Flowchart corresponding to the Algorithm.

Some open problems

- For the on-line fault detection problem:

Some open problems

- For the on-line fault detection problem:

* Assuming that there is only one kind of faults (a non-faulty and a faulty BCN), how can we estimate the time $\bar{t}$ at which the fault occurred?

Some open problems

- For the on-line fault detection problem:

* Assuming that there is only one kind of faults (a non-faulty and a faulty BCN), how can we estimate the time $\bar{t}$ at which the fault occurred?

* What if different types of faults may occur? How can we identify which fault affected the BCN?

Some open problems

- For the on-line fault detection problem:

* Assuming that there is only one kind of faults (a non-faulty and a faulty BCN), how can we estimate the time $\bar{t}$ at which the fault occurred?
* What if different types of faults may occur? How can we identify which fault affected the BCN?
* What if the fault is reversible?

Some open problems

- For the on-line fault detection problem:

* Assuming that there is only one kind of faults (a non-faulty and a faulty BCN), how can we estimate the time $\bar{t}$ at which the fault occurred?
* What if different types of faults may occur? How can we identify which fault affected the BCN?
* What if the fault is reversible?

- For the off-line fault detection problem:

Some open problems

- For the on-line fault detection problem:

* Assuming that there is only one kind of faults (a non-faulty and a faulty BCN), how can we estimate the time $\bar{t}$ at which the fault occurred?
* What if different types of faults may occur? How can we identify which fault affected the BCN?
* What if the fault is reversible?

- For the off-line fault detection problem:

* If solvable, how can we find the most efficient (= the shortest input test sequence)?

Thanks for your attention!

Questions?