Observer design and fault detection of Boolean control networks

Prof. Dr. Ping Zhang

Institute of Automatic Control
University of Kaiserslautern-Landau
Germany

Introduction

Boolean control networks (BCN):

$$\begin{cases} X_1(k+1) = f_1\big(X_1(k), \cdots, X_n(k), U_1(k), \cdots, U_m(k)\big) \\ \vdots \\ X_n(k+1) = f_n\big(X_1(k), \cdots, X_n(k), U_1(k), \cdots, U_m(k)\big) \\ Y_1(k) = h_1\big(X_1(k), \cdots, X_n(k)\big) \\ \vdots \\ Y_p(k) = h_p\big(X_1(k), \cdots, X_n(k)\big) \end{cases}$$

$$\left.\begin{array}{ll} X_1(k), \cdots, X_n(k): & \text{States} \\ U_1(k), \cdots, U_m(k): & \text{Control inputs} \\ Y_1(k), \cdots, Y_p(k): & \text{Measured outputs} \end{array}\right\} \quad \text{Binary signals (0/1)}$$

$f_1(k), \cdots, f_n(k), h_1(k), \cdots, h_p(k)$: Boolean functions

Matrix expression of Boolean control networks

$$x(t+1) = L \ltimes x(t) \ltimes u(t)$$
$$y(t) = H \ltimes x(t)$$

L, H: constant matrices (contain all information about Boolean functions)

$x(t) \in \Delta_{2^n}, u(t) \in \Delta_{2^m}, y(t) \in \Delta_{2^p}$

$$\Delta_{2^n} = \left\{ \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{1 \times 2^n}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \right\} = \{ \delta_{2^n}^1, \delta_{2^n}^2, \cdots, \delta_{2^n}^{2^n} \}$$

Motivation

- Our work is motivated by one course "logic control systems" with focus on automaton and Petri nets. **Graphical tool → analytical approach?**
- The **semi-tensor product (STP)** introduced by Cheng and Co-workers provides a powerful and convenient tool for the analysis and design of BCN.
- **Key idea**: Matrix and vector expression of logical operation and logical variables through STP
- **Focus of this talk**: **Observer design** and **fault detection** for Boolean control networks

Main content

- Not all state variables can be measured → extension of the idea of Kalman filter to **observer design** for BCN
- Handling of unknown disturbances → Design of **unknown input observer**
- How to handle large-scale BCN? → **Distributed observer** design
- How to reduce computational efforts? → **Reduced-order observer**
- How to apply observer for fault detection purpose? → **observer-based fault detection for BCN and probabilistic BCN**
- Design control input signal to improve fault detection performance? → **observer-based active fault detection**

Main content in this talk

- Not all state variables can be measured → extension of the idea of Kalman filter to **observer design** for BCN
- Handling of unknown disturbances → Design of **unknown input observer**
- How to handle large-scale BCN? → **Distributed observer** design
- How to reduce computational efforts? → **Reduced-order observer**
- How to apply observer for fault detection purpose? → **observer-based fault detection for BCN and probabilistic BCN**
- Design control input signal to improve fault detection performance? → **observer-based active fault detection**

Especially related STP properties

Properties (Cheng et al., World Scientific, 2012)

- $(A \ltimes B)^{\mathrm{T}} = B^{\mathrm{T}} \ltimes A^{\mathrm{T}}$
- Power-reducing matrix: $\Phi_n = [\delta_{2^n}^1 \otimes \delta_{2^n}^1 \quad \delta_{2^n}^2 \otimes \delta_{2^n}^2 \quad \cdots \quad \delta_{2^n}^{2^n} \otimes \delta_{2^n}^{2^n}]$:

$$X \ltimes X = \Phi_n \ltimes X$$

- $X \ltimes A = (I_{2^n} \otimes A) \ltimes X$
- Swap matrix $W_{[2^m,2^n]}$: $X \ltimes Y = W_{[2^m,2^n]} \ltimes Y \ltimes X$

Lemma 1

The element-wise multiplication of vectors $X \in \mathbb{R}^{2^n \times 1}$ and $Y \in \mathbb{R}^{2^n \times 1}$ can be described by using power-reducing matrix Φ_n:

$$X \odot Y = \Phi_n^{\mathrm{T}} \ltimes X \ltimes Y, \text{ with } \Phi_n = [\delta_{2^n}^1 \otimes \delta_{2^n}^1 \quad \delta_{2^n}^2 \otimes \delta_{2^n}^2 \quad \cdots \quad \delta_{2^n}^{2^n} \otimes \delta_{2^n}^{2^n}]$$

Observer design

BCN:

$$x(t+1) = L \ltimes x(t) \ltimes u(t)$$
$$y(t) = H \ltimes x(t)$$

Main purpose of observer design:

Get a good estimate $\hat{x}(t)$ based on the available information of $u(t)$ and $y(t)$.

Observer design for Boolean control networks (Zhang et al., CDC 2016)

$$\hat{x}(0) = H^T y(0)$$

$$\hat{x}(t) = \underbrace{L\hat{x}(t-1)u(t-1)}_{\substack{Propogation\ of \\ state\ estimate\ \ \hat{x}(t-1)}} \quad \underbrace{\odot}_{\substack{elementweis \\ multiplication}} \quad \underbrace{H^T y(t)}_{\substack{Estimation\ of\ x(t) \\ directly\ from\ y(t)}}$$

$$= \Phi_n^T L\hat{x}(t-1)u(t-1)H^T y(1)$$

$$= \Phi_n^T (I_{2^n} \otimes H^T) L\hat{x}(t-1)u(t-1)y(t)$$

Example

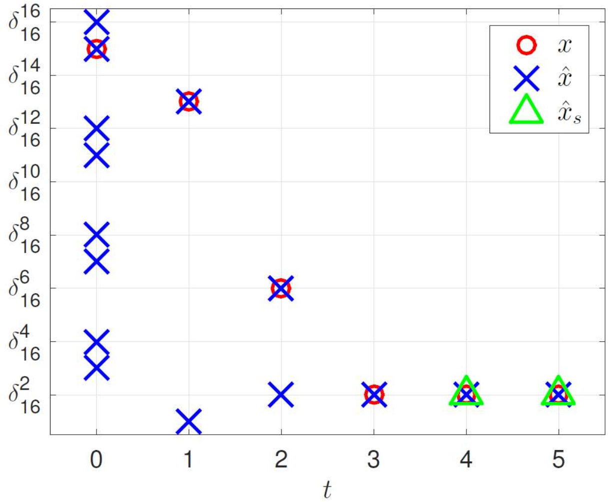
BCN:

$$\begin{cases} x_1(t+1) & = x_1(t) \vee x_3(t) \\ x_2(t+1) & = x_1(t) \\ x_3(t+1) & = u(t) \vee x_2(t) \vee x_4(t) \\ x_4(t+1) & = \neg x_3(t) \\ y(t) & = x_3(t) \end{cases}$$

Input and output sequence (logic value):

t	0	1	2	3	4	5
u	0	1	1	0	0	
y	0	1	1	1	1	1

State estimate:

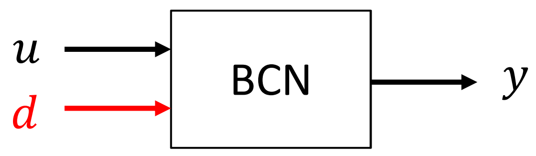


x: true state
$\hat{x}$: Luenberger-observer
$\hat{x}_s$: Shift-register observer

Unknown input observer design

Consider a BCN with unknown inputs

$$x(t+1) = Lx(t)u(t)d(t) \qquad u(t)\text{: control inputs}$$
$$y(t) = Hx(t) \qquad d(t)\text{: unknown inputs}$$



The BCN is said to be **unknown input reconstructible**, if for any possible input $d(t)$ a non-negative integer r can be found, so that with the knowledge of every admissable control input and output trajectory $\{u(0), y(0), u(1), y(1), \cdots, u(t), y(t)\}, t \geq r,$ the final state $x(t)$ can be unique determined.

Unknown input observer design

The set of states that can be reached at time $t+1$ from a given state $x(t)$:

$$
\begin{aligned}
x(t+1) &= Lx(t)u(t)d(t) \\
&= LW_{[2^{m_d},2^{n+m}]}d(t)x(t)u(t) \\
&= LW_{[2^{m_d},2^{n+m}]}\left(\begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix}+\begin{bmatrix}0\\1\\\vdots\\0\end{bmatrix}+\cdots+\begin{bmatrix}0\\0\\\vdots\\1\end{bmatrix}\right)x(t)u(t) \\
&= LW_{[2^{m_d},2^{n+m}]}\underbrace{\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}}_{\mathbf{1}_{2^{m_d}}}x(t)u(t) \\
&= \tilde{L}x(t)u(t)
\end{aligned}
$$

Unknown input observer design

Introduce an **auxiliary system**

$$x(t+1) = \tilde{L}x(t)u(t)$$
$$y(t) = Hx(t)$$

The state candidates in $\hat{x}(t)$ that can generate the output $y(t)$ are

$$\hat{x}(0) = H^T y(0)$$

$$\hat{x}(t) = \tilde{L}\hat{x}(t)u(t) \odot H^T y(t)$$
$$= \Phi_n^T \tilde{L}\hat{x}(t-1)u(t-1)H^T y(t)$$
$$= \Phi_n^T (I_{2^n} \otimes H^T)\tilde{L}\hat{x}(t-1)u(t-1)y(t)$$

Unknown input observer design

Theorem for checking Unknown Input Reconstructibility

The BCN is unknown input reconstructible, if and only if the following two conditions are satisfied:

(1) The matrix $\Phi_n^T(I_{2^n} \otimes H^T)\tilde{L}$ contains only zero columns or columns with only one non-zero entry.

(2) The auxiliary system is reconstructible.

Example

BCN with unknown inputs ξ and ω:

$$\begin{cases} x_1(t+1) &= u(t) \wedge \neg x_6(t) \\ x_2(t+1) &= \neg x_1(t) \\ x_3(t+1) &= \neg x_1(t) \wedge (x_5(t) \vee x_3(t)) \\ x_4(t+1) &= x_2(t) \wedge \neg x_6(t) \\ x_5(t+1) &= (x_4(t) \vee \neg x_3(t)) \wedge \xi(t) \\ x_6(t+1) &= x_5(t) \wedge (\neg x_6(t) \vee \neg x_2(t)) \end{cases}$$

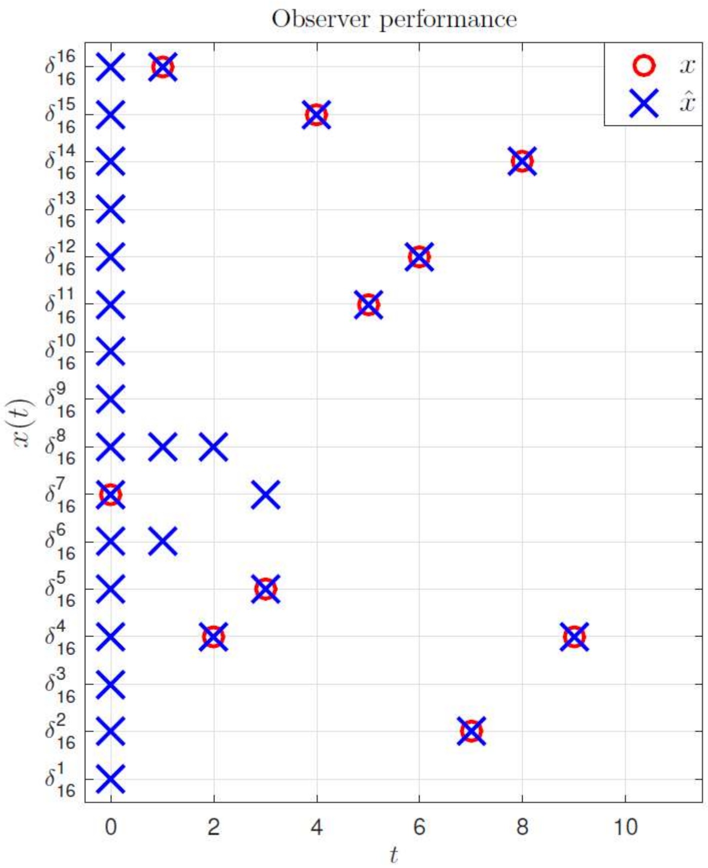
$$y_1(t) = x_2(t) \vee \omega(t)$$

$$y_2(t) = x_3(t), \; y_3(t) = x_5(t)$$

Input and output sequence (logic value):

t	0	1	2	3	4	5	6	7	8	9
u	1	1	1	1	1	0	1	0	1	
ξ	1	1	1	1	0	1	1	0	1	
ω	1	1	0	0	0	0	0	0	0	0
y_1	1	1	1	0	0	1	1	1	0	1
y_2	1	0	0	0	0	1	1	1	0	0
y_3	0	0	1	1	1	0	0	0	0	1

State estimate:



x: true state $(x_1 x_2 x_4 x_6)$

$\hat{x}$: unknown input observer

Observer design for large-scale BCN

Main Challenges:

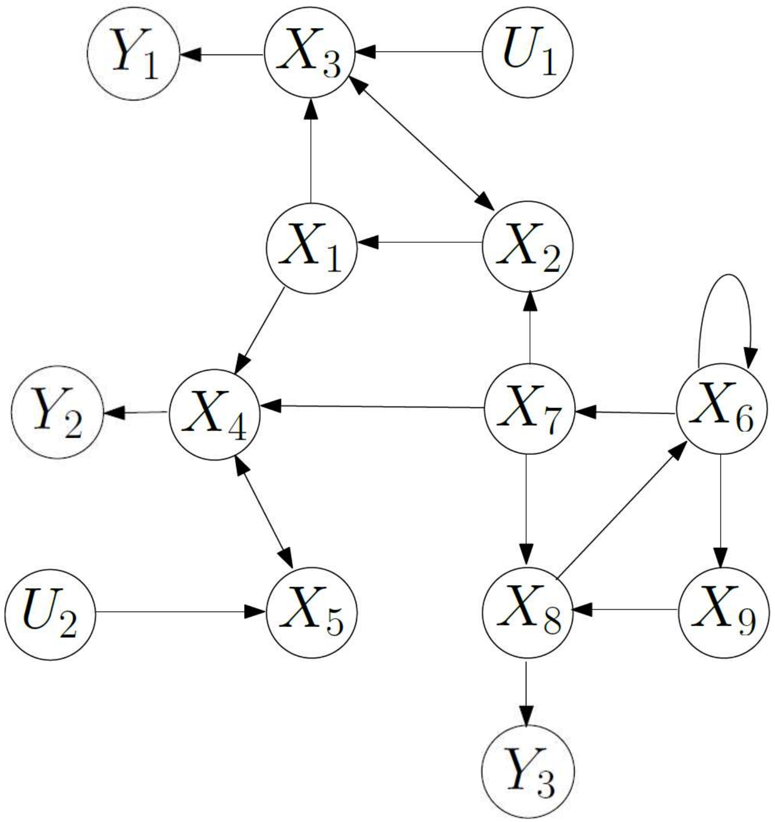
- High dimensional matrices L and H
- difficulty in the implementation of the observer due to memory and computational efforts

➡ **Distributed observer design**

Basic idea of distributed observer design

Step 1: Express the large-scale Boolean network as a directed graph

$$\begin{cases} X_1(t+1) &= X_2(t) \\ X_2(t+1) &= X_3(t) \wedge X_7(t) \\ X_3(t+1) &= X_1(t) \wedge X_2(t) \wedge U_1(t) \\ X_4(t+1) &= \neg X_1(t) \wedge X_5(t) \wedge X_7(t) \\ X_5(t+1) &= X_4(t) \wedge U_2(t) \\ X_6(t+1) &= X_6(t) \vee X_8(t) \\ X_7(t+1) &= X_6(t) \\ X_8(t+1) &= X_5(t) \vee X_7(t) \vee X_9(t) \\ X_9(t+1) &= \neg X_8(t) \\ Y_1(t) = X_3(t), \quad Y_2(t) = X_4(t), \quad Y_3(t) = X_8(t) \end{cases}$$



Basic idea of distributed observer design

Step 2: Partition the large-scale boolean networks into subnetworks

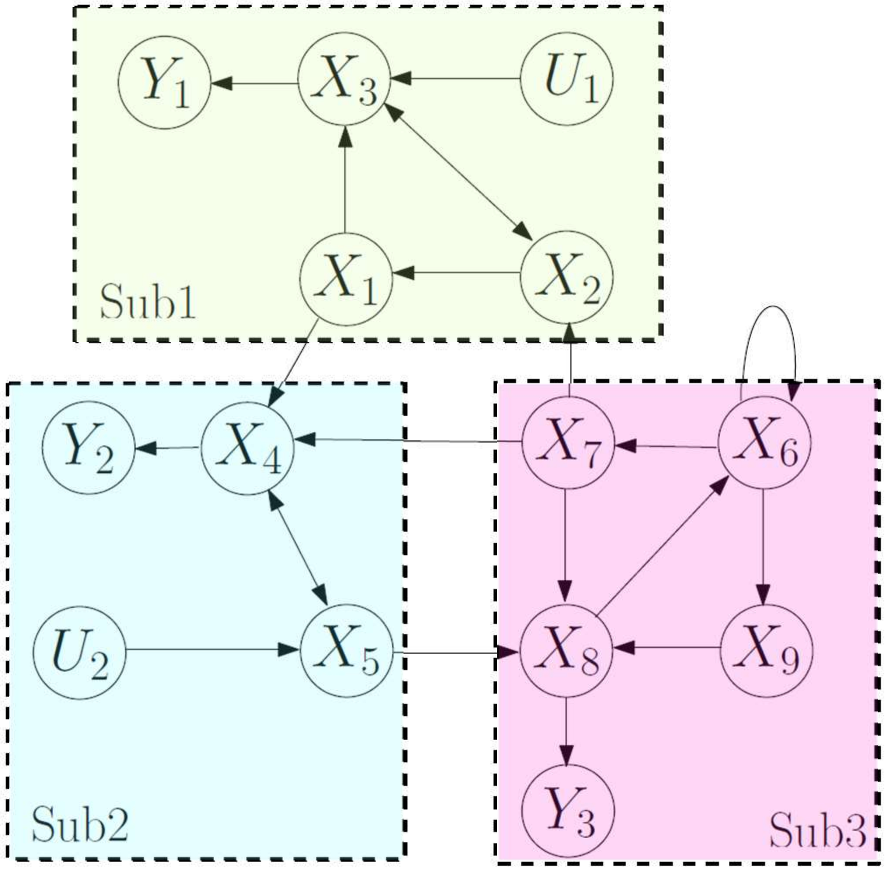
$$\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2 \cup \cdots \cup \mathcal{X}_\alpha,$$
$$\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2 \cup \cdots \cup \mathcal{U}_\alpha,$$
$$\mathcal{Y} = \mathcal{Y}_1 \cup \mathcal{Y}_2 \cup \cdots \cup \mathcal{Y}_\alpha,$$

i-th subnetwork:

$$\mathcal{N}_i = \mathcal{X}_i \cup \mathcal{U}_i \cup \mathcal{Y}_i$$

In-neighbors to the i-th subnetwork: the incoming edges from other subnetworks.

Each state and output belongs to only one subnetwork.

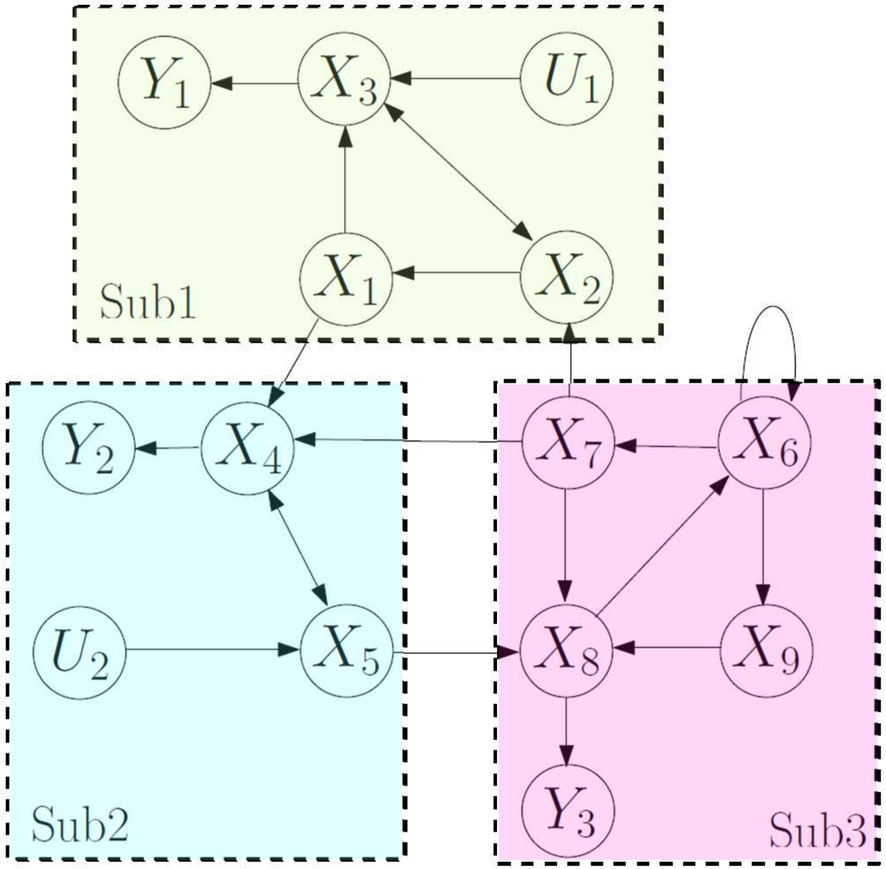


Basic idea of distributed observer design

Step 2: Partition the large-scale boolean networks into subnetworks

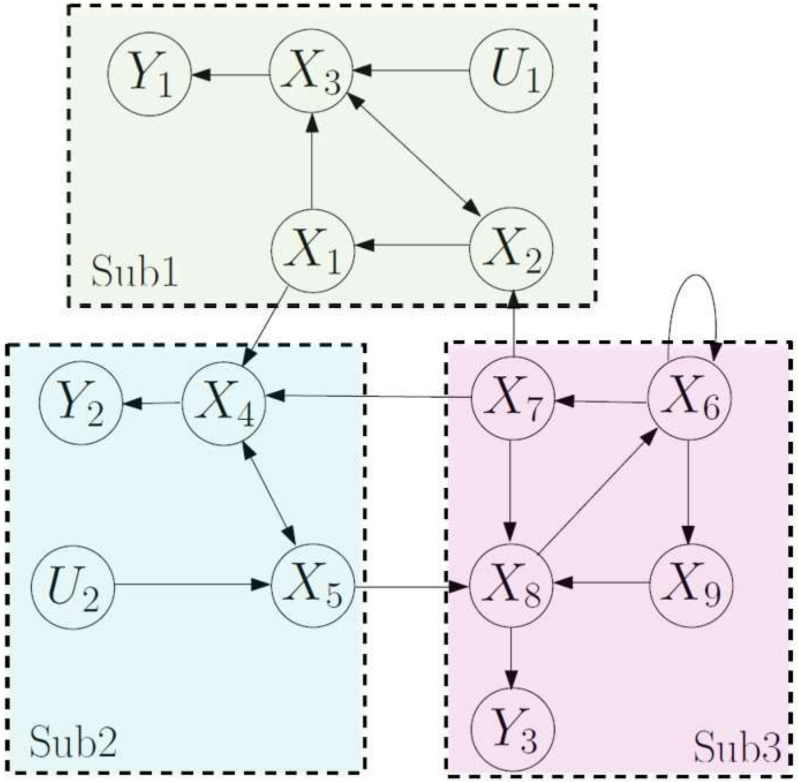
Basic requirement on partition:

- Each subnetwork should contain a small number of states → reduce memory and computational efforts
- Each subnetwork should connect to a small number of in-neighbors subsystems → reduce communication



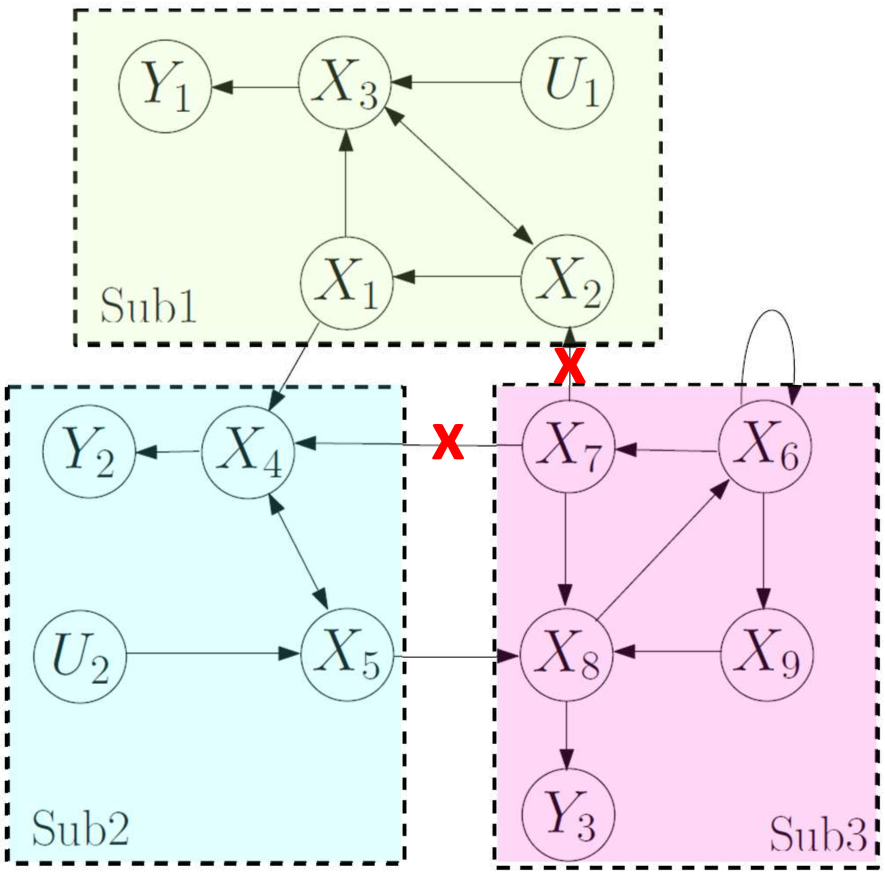
Basic idea of distributed observer design

Step 3: For each subnetwork, design a local observer to estimate the states. Aggregate the results to get the estimate of all states.



Distributed observer design

Consider at first a simple case:

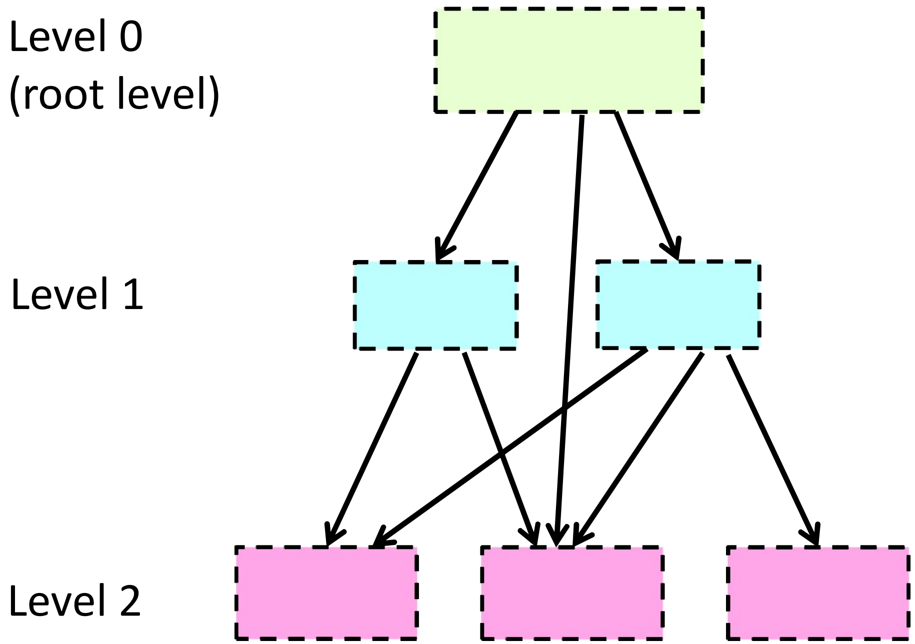


Estimation procedure:

- Estimate the states of Subnetwork 1 (who has no in-neighbors)
- Estimate the states of the Subnetwork 2
- Estimate the states of the Subnetwork 3

Distributed observer design

Large-scale BCN with acyclic structure: No directed cycles exists in the graph

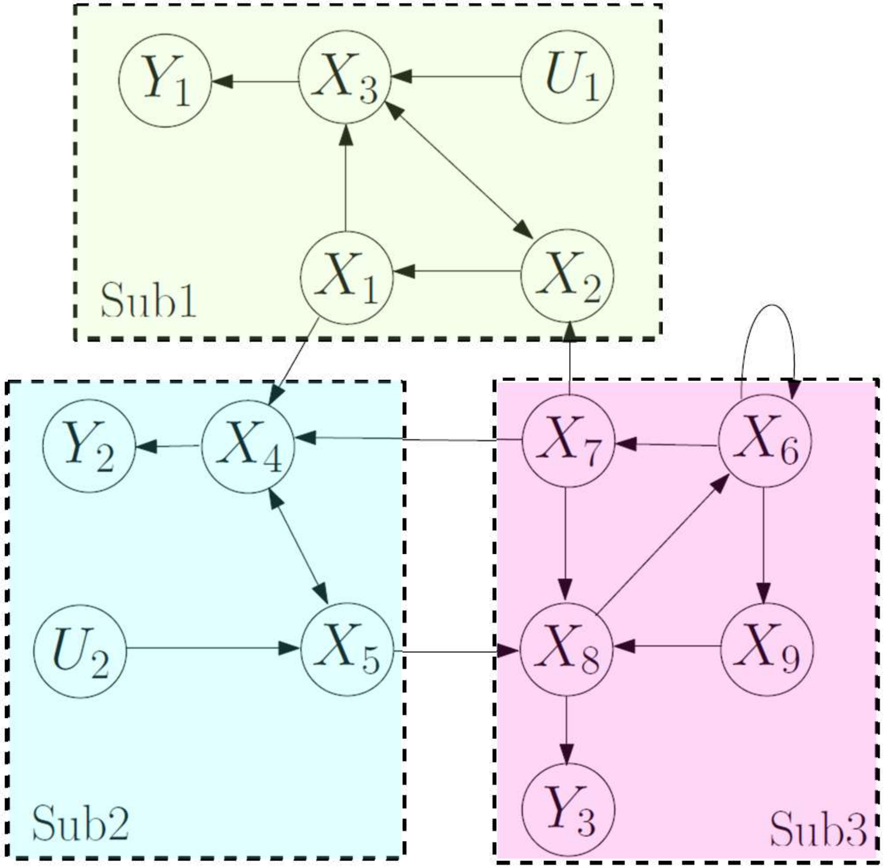


- Sort the subnetworks into β levels.
- A large-scale BCN with acyclic structure is reconstructible, if all subnetworks are reconstructible.
- The minimal reconstructibility index of the whole network can be estimated based on the minimal reconstructibility index of subnetworks.

How to handle large-scale BCNs which don’t have acyclic structure?

Distributed observer design

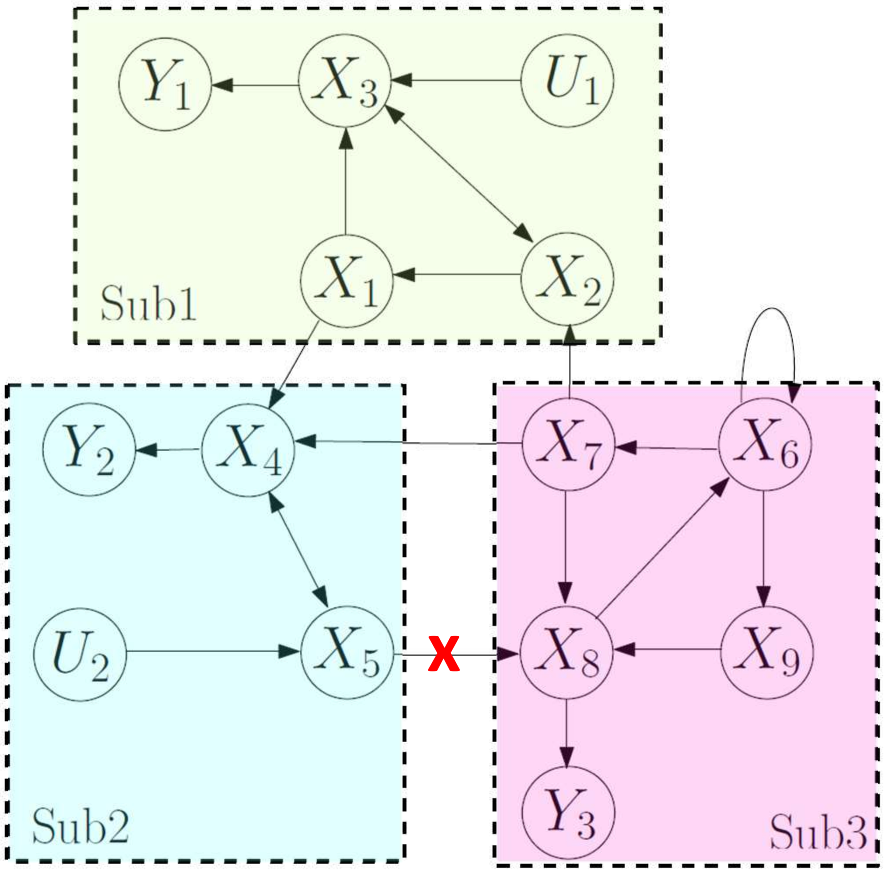
Consider our example before:



(1) Subsystem 1, 2, 3 are reconstructible.

(2) Subsystem 3 is unknown input reconstructible with respect to the incoming-Edge from X_5. → The incoming-edge from X_5 to X_8 can be cut off.

Example



(1) Subsystem 1, 2, 3 are reconstructible.

(2) Subsystem 3 is unknown input reconstructible with respect to the incoming-Edge from X_5. → The incoming-edge from X_5 to X_8 can be cutted off.

(3) Cutting off the incoming edge from X_5 to X_8 results in a directed acyclic graph.

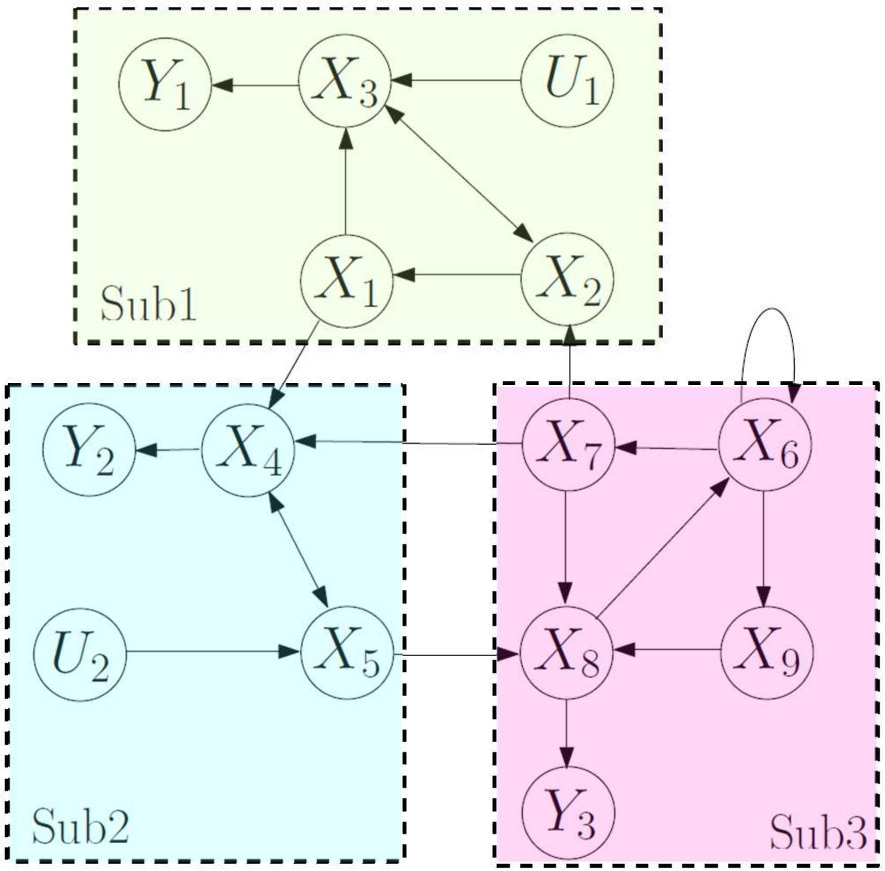
Level 0: Sub3

Level 1: Sub 1

Level 2: Sub2

Distributed observer design

Large-scale BCN with arbitrary structure



A large-scale BCN is reconstructible, if

(1) All subnetworks are reconstructible.

(2) There exist a set Ψ of subnetworks, which are unknown input reconstructible with respect to the states from their in-neighbors.

(3) Cutting off the incoming edges of all subnetworks in the set Ψ results in a directed acyclic graph.

Distributed observer design

Design procedure:

Step 1: After the partition of the large-scale BCN, we get a set of subnetworks

$$x_i(t+1) = L_i x_i(t) u_i(t) z_i(t)$$
$$y_i(t) = H_i x_i(t), \qquad i = 1, \cdots, \alpha$$

$z_i(t)$: the coupling states from the in-neighbors of the i-th subnetwork

Step 2: Check the reconstructibility and the unknown input reconstructibility of each subnetwork

Step 3: Design a distributed observer which contains α local observers

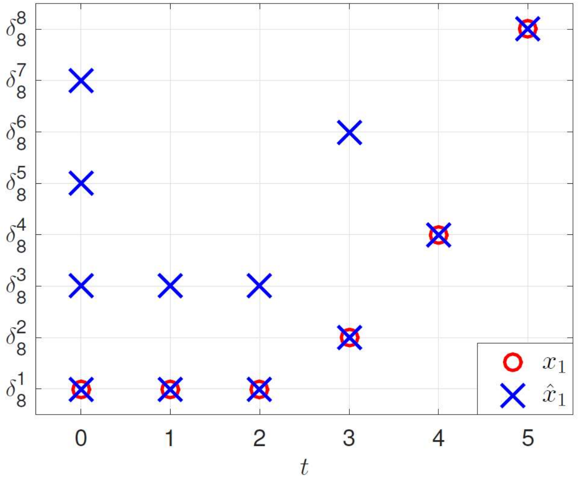
$$\hat{x}_i(0) = {H_i}^T y_i(0)$$
$$\hat{x}_i(t) \; = \Phi_{n_i}^T \left(I_{2^n} \otimes {H_i}^T\right) L_i \hat{x}_i(t-1) u_i(t-1) \hat{z}_i(t-1) y_i(t)$$

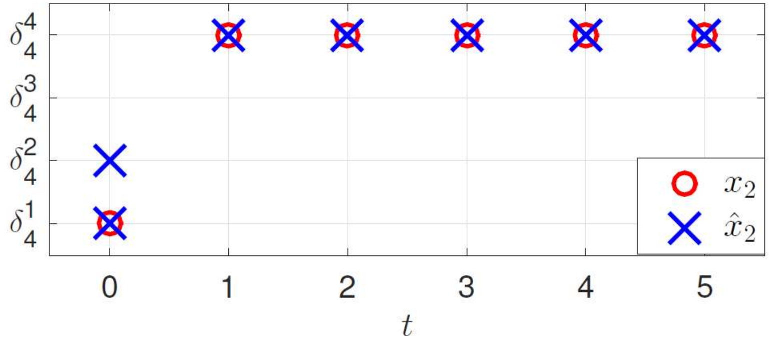
$i = 1, \cdots, \alpha$, $\hat{z}_i(t-1)$ are information delivered by the in-neighbors

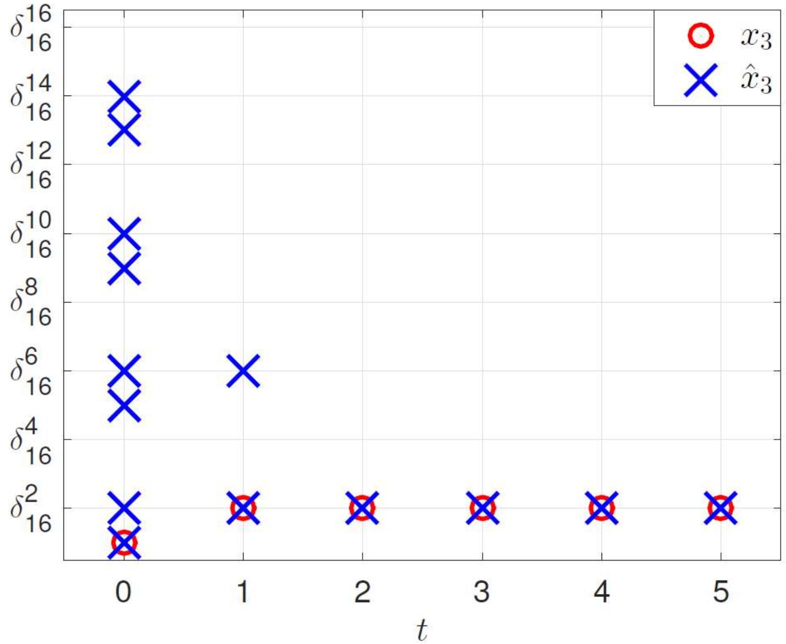
Example

Input and output trajectory (Logical value)

t	0	1	2	3	4	5
U_1	1	1	0	0	0	
U_2	0	1	1	0	0	
Y_1	1	1	1	0	0	0
Y_2	1	0	0	0	0	0
Y_3	1	1	1	1	1	1

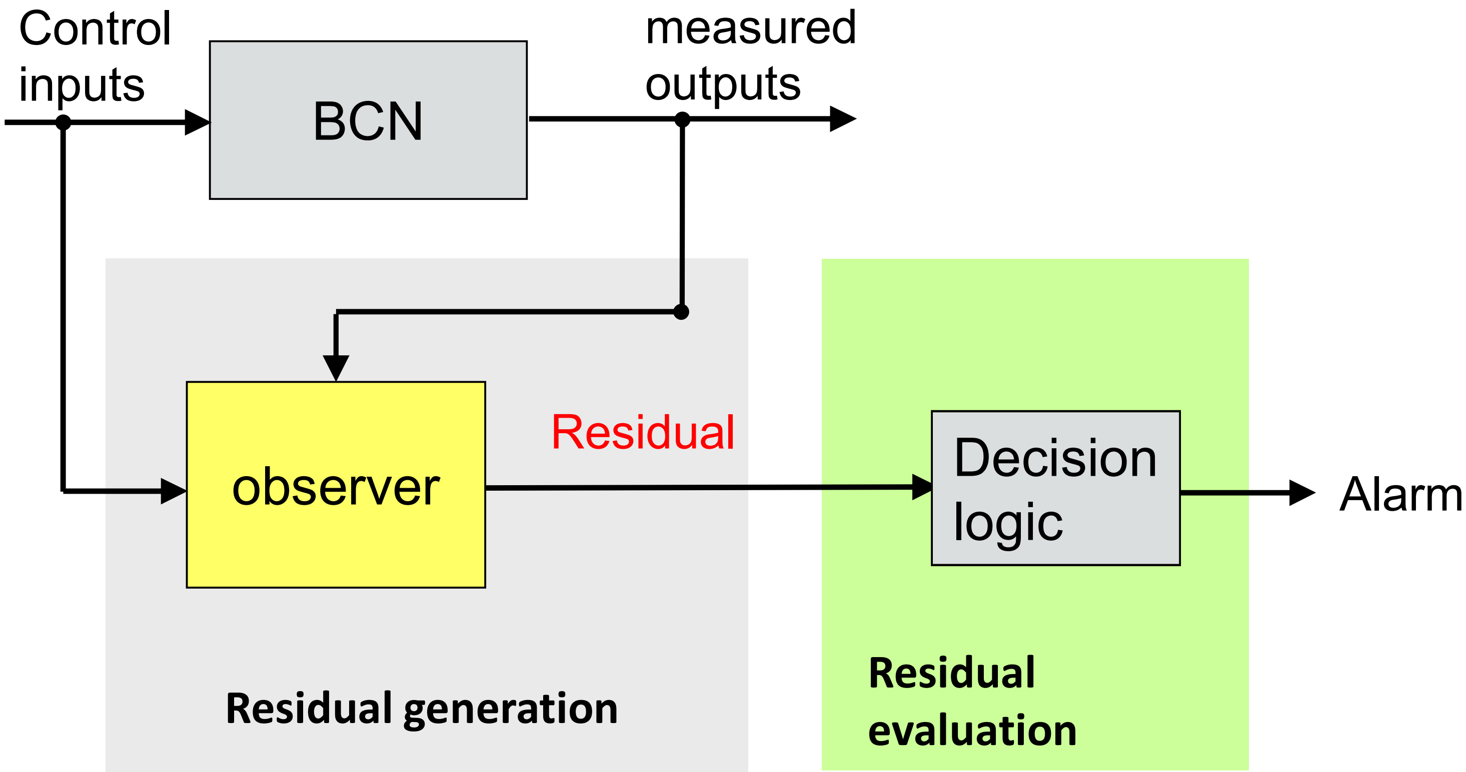






Fault detection of BCN

Goal: detect the abnormality in the system behaviour based on online measurement data and system model



Fault detection of BCN

Fault-free system

$$x(t+1) = Lu(t)x(t)$$
$$y(t) = Hx(t)$$

Observer-based fault detector:

$$\hat{x}(t) = \Phi_n^T (I_{2^n} \otimes H^T) L W_{[2^n,2^m]} \hat{x}(t-1) u(t-1) y(t)$$
$$r(t) = 1 - \mathbf{1}_{2^n}^T \hat{x}(t)$$

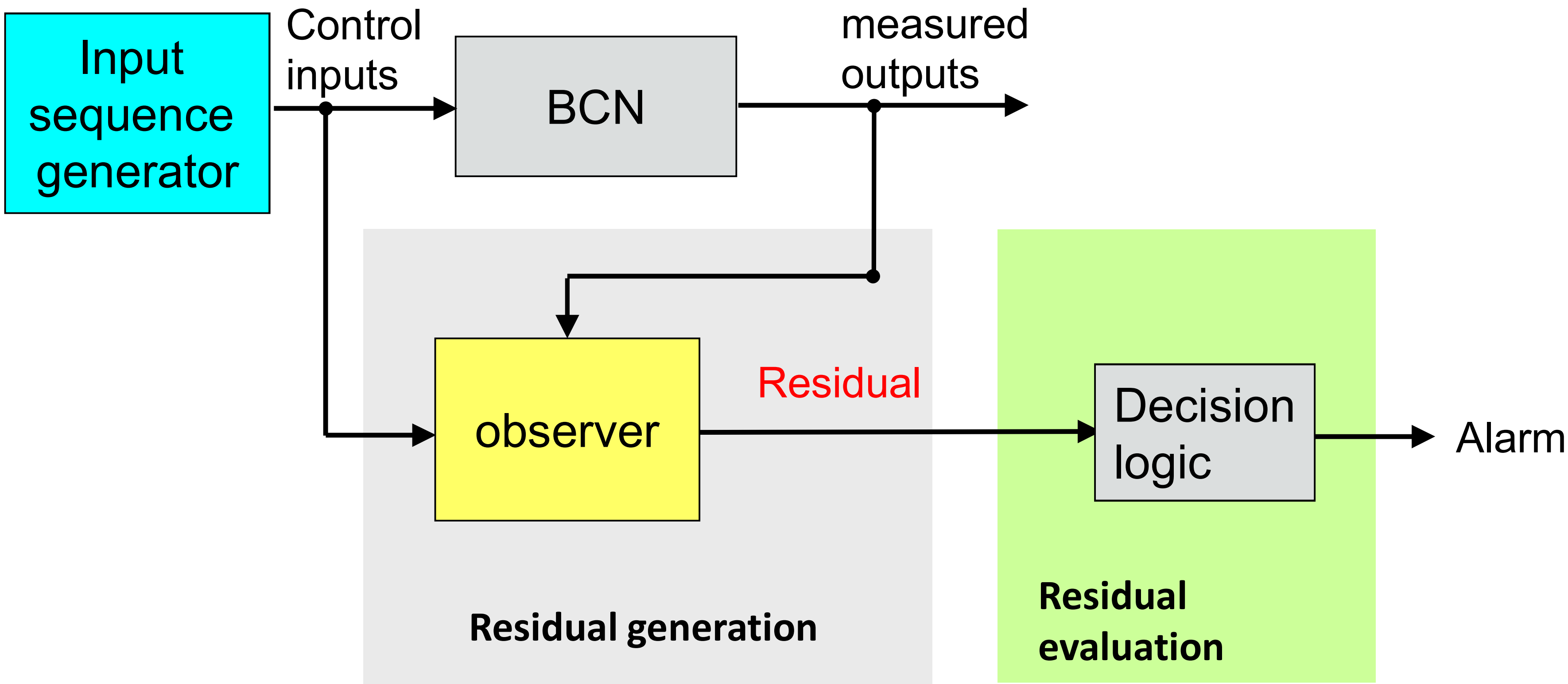
$r(t)$: residual signal

Decision logic:

$$\begin{cases} r(t) = 1 & \Longleftrightarrow \quad \text{A fault has happened} \\ r(t) < 1 & \Longleftrightarrow \quad \text{The BCN is fault-free} \end{cases}$$

Active fault detection

Research question: Could we design the input sequence, so that the fault better detectable?



Active fault detection

Fault-free system

$$x(t+1) = Lu(t)x(t)$$
$$y(t) = Hx(t)$$

Faulty system

$$x_F(t+1) = L_F u(t) x_F(t)$$
$$y_F(t) = H_F x(t)$$

A fault is said to be **actively detectable**, if there exists an input sequence $\{u(0), u(1), \cdots, u(T)\}$, such that the output trajectories $\{y(0), y(1), \cdots, y(T+1)\}$ and $\{y_F(0), y_F(1), \cdots, y_F(T+1)\}$ generated by the fault-fee system and the faulty system differ at some time instant t with $t \in [0, T+1]$, no matter what the initial state is.

Active fault detection

Basic idea: Introduce an auxiliary state variable

$$\tilde{x}(t) = x(t)x_F(t)$$

Dynamics of the joint system

$$\begin{aligned}
\tilde{x}(t+1) &= x(t+1)x_F(t+1) \\
&= Lu(t)x(t)L_F u(t)x_F(t) \\
&= L(I_{2^{m+n}} \otimes L_F)u(t)x(t)u(t)x_F(t) \\
&= L(I_{2^{m+n}} \otimes L_F)u(t)W_{[2^m,2^n]}u(t)x(t)x_F(t) \\
&= L(I_{2^{m+n}} \otimes L_F)\big(I_{2^{m+n}} \otimes W_{[2^m,2^n]}\big)u(t)u(t)x(t)x_F(t) \\
&= L(I_{2^{m+n}} \otimes L_F)\big(I_{2^{m+n}} \otimes W_{[2^m,2^n]}\big)\Phi_m u(t)x(t)x_F(t) \\
&= \underbrace{L(I_{2^{m+n}} \otimes L_F)\big(I_{2^{m+n}} \otimes W_{[2^m,2^n]}\big)\Phi_m}_{\tilde{L}} u(t)\tilde{x}(t) \\
&= \tilde{L}u(t)\tilde{x}(t)
\end{aligned}$$

Active fault detection

Compare the output of the fault-free system and the faulty system

$$\begin{cases} y(t) \odot y_F(t) = 0, & if\ y(t) \neq y_F(t) \\ y(t) \odot y_F(t) \in \Delta_{2^p}, & if\ y(t) = y_F(t) \end{cases}$$

Let

$$\tau = 1 - \mathbf{1}_{2^p}^T \left(y(t) \odot y_F(t) \right) = \begin{cases} 1, & if\ y(t) \neq y_F(t) \\ 0, & if\ y(t) = y_F(t) \end{cases}$$

Then

$$\begin{aligned} \tau &= 1 - \mathbf{1}_{2^p}^T \Phi_p^T y(t) y_F(t) \\ &= 1 - \mathbf{1}_{2^p}^T \Phi_p^T H x(t) H_F x_F(t) \\ &= 1 - \mathbf{1}_{2^p}^T \Phi_p^T H (I_{2^n} \otimes H_F) x(t) x_F(t) \\ &= \mathbf{1}_{2^{2n}}^T x(t) x_F(t) - \mathbf{1}_{2^p}^T \Phi_p^T H (I_{2^n} \otimes H_F) x(t) x_F(t) \\ &= \underbrace{\left(\mathbf{1}_{2^{2n}}^T - \mathbf{1}_{2^p}^T \Phi_p^T H (I_{2^n} \otimes H_F) \right)}_{\widetilde{H}} \tilde{x}(t) \\ &= \widetilde{H} \tilde{x}(t) \end{aligned}$$

Active fault detection

The auxiliary system

$$\tilde{x}(t+1) = \tilde{L}u(t)\tilde{x}(t)$$
$$\tau(t) = \widetilde{H}\tilde{x}(t) = \begin{cases} 1, & if\ y(t) \neq y_F(t) \\ 0, & if\ y(t) = y_F(t) \end{cases}$$

The active fault detection problem: find an input sequence, so that $\tau = 1$, no matter what the initial state is.

An equivalent problem: Could we find an input sequence, so that the output of the auxiliary system be stabilized to 1?

Active fault detection

The equivalent problem can be solved as a deadbeat stabilization problem of autonomous switched linear discrete-time system.

Observer-based fault detector:

$$\hat{x}(t) = \Phi_n^T(I_{2^n} \otimes H^T)LW_{[2^n,2^m]}\hat{x}(t-1)u(t-1)y(t)$$
$$r(t) = 1 - \mathbf{1}_{2^n}^T\hat{x}(t)$$

$r(t)$: residual signal

Decision logic:

$$\begin{cases} r(t) = 1 & \Leftrightarrow \quad \text{A fault has happened} \\ r(t) < 1 & \Leftrightarrow \quad \text{The BCN is fault-free} \end{cases}$$

Conclusion

- The STP is a useful and systematic tool that can be used to carry out observer design and fault detection in BCN
- Some algorithms can be further developed. For instance, we can find an analytical way to find the optimal partition of large-scale BCN.

Literature

- Z. Zhang, T. Leifeld, P. Zhang. Observer design for Boolean control networks. *Proceedings of the 55th IEEE Conference on Decision and Control (CDC)*, pp. 6272-6277, Las Vegas, USA, 2016.
- T. Leifeld, Z. Zhang, P. Zhang. Fault detection for probabilistic Boolean networks. *Proceedings of the 2016 European Control Conference (ECC)*, pp. 740-745, Aalborg, Denmark, 2016.
- Z. Zhang, T. Leifeld, P. Zhang. Unknown input decoupling and estimation in observer design for Boolean control networks. *Proceedings of the 20th IFAC World Congress (IFAC)*, pp. 2972-2977, Toulouse, France, 2017.
- Z. Zhang, T. Leifeld, P. Zhang. Distributed observer design for large-scale Boolean control networks. *Proceedings of the 2017 American Control Conference (ACC)*, pp. 2618-2623, Seattle, USA, 2017.
- Z. Zhang, T. Leifeld, P. Zhang. Active fault detection of Boolean control networks. *Proceedings of the 2018 American Control Conference (ACC)*, pp. 5001-5006, Milwaukee, Wisconsin, USA, 2018.
- Z. Zhang, T. Leifeld, P. Zhang. Reconstructibility analysis and observer design for Boolean control networks. *IEEE Transactions on Control of Network Systems*, Vol. 7, No. 1, pp. 516-528, 2020.
- Z. Zhang, T. Leifeld, P. Zhang. Reduced-order observer design for Boolean control networks. *IEEE Transactions on Automatic Control*, Vol. 65, No. 1, pp. 434-441, 2020.
- Z. Zhang, P. Zhang, T. Leifeld. Reduced-order Observer Design for Fault Diagnosis of Boolean Control Networks. *Automatica*, Vol. 146, 110618, 2022.

Literature

- T. Leifeld, J. Schlegel, P. Zhang. A new approach to the reliability analysis of safety instrumented systems (in German). *Automatisierungstechnik*, Vol. 64, No. 6, pp. 457-466, 2016.
- Z. Zhang, T. Leifeld, P. Zhang. Identification of Boolean control networks incorporating prior knowledge. *Proceedings of the 56th IEEE Conference on Decision and Control (CDC)*, pp. 5839-5844, Melbourne, Australia, 2017
- T. Leifeld, Z. Zhang, P. Zhang. Data-driven controller design for Boolean control networks. *Proceedings of the 2018 American Control Conference (ACC)*, pp. 3044-3049, Milwaukee, Wisconsin, USA, 2018.
- T. Leifeld, Z. Zhang, P. Zhang. Identification of Boolean network models from time series data incorporating prior knowledge. *Frontiers in Physiology*, Vol. 9, Article 695, 12 Pages, 2018.
- Z. Zhang, T. Leifeld, P. Zhang. Finite horizon tracking control of Boolean control networks. *IEEE Transactions on Automatic Control*, Vol. 63, No. 3, pp. 1798-1805, 2018.
- Z. Zhang, T. Leifeld, P. Zhang. An improved algorithm for stabilization of Boolean networks via pinning control. *Proceedings of the 2019 IEEE Conference on Decision and Control (CDC)*, pp. 114-119, Nice, France, 2019.
- T. Leifeld, Z. Zhang, P. Zhang. Overview and comparison of approaches towards an algebraic description of discrete event systems. *Annual Reviews in Control*, Vol. 48, pp. 80-88, 2019.

Thank you for your attention!