Diagnosability of composite automata based on semi-tensor product

Zengqiang Chen, Yingrui Zhou, Zhipeng Zhang and Yongyi Yan

ABSTRACT
Fault diagnosis is an important issue of partially observed discrete event systems (DESs). In this problem, fault detection and isolation are two associated tasks, where a fault is diagnosable if it can be detected certainly with a finite delay occurrence and a system is diagnosable if any type of faults can be distinguished with the observed information. Existing researches for fault diagnosis tend to regard faults as unobservable faulty events or states. In this paper, there exists a key conversion, denoting faults by accessible changes of state transition, and then obtains a normative and a faulty construction of given systems. Specially, we establish algebraic structures of composite automata with the help of semi-tensor product and propose a definition of diagnosability based on algebraic state space. Notice that the definition is so different from existing results that provide a new opportunity to fault diagnosis when existing criteria fail. Besides, we construct matrix observers to summarize the requisite information in the evolutionary process and present corresponding theorems to the verification of diagnosability. We apply the exhaust gas recirculation system and the heating system to illustrate the matrix approach in this paper is feasible.

ARTICLE HISTORY
Received 31 July 2020
Accepted 9 December 2020

KEYWORDS
Discrete event systems; composite automata; fault detection; diagnosability; semi-tensor product; matrix approach

1. Introduction
Property verification of discrete event systems (DESs) is a crucial subject and there has been considerable excellent results (Cassandras & Lafortune, 2009; Yin, 2019), containing detectability (Han, Chen, Zhao, 2017; Shu & Lin, 2013; Shu et al., 2007), opacity (Saboori & Hadjicostis, 2012; Yin & Lafortune, 2017) and diagnosability (HashtrudiZad et al., 2003; Lin, 1994; Reshmila & Rajagopalan, 2019; Sampath et al., 1995; Yin & Li, 2016b; Zaytoon & Lafortune, 2013; Zhu et al., 2017). Today, with increasing systems’ security requirements, diagnostic problem research has always made sense. In the history of diagnosability analysis for DESs, scholars tend to regard faults as unobservable faulty events or states. Typically, the work in Lin (1994) used states to model faults and studied both on-line and off-line diagnostics. The language-based definition of diagnosability was first proposed in Sampath et al. (1995), where a language can be diagnosed if faults are detectable and isolable with a finite delay occurrence. Besides, the authors proposed a valid approach for detection and isolation of faulty events by diagnosers. From then on, many expanded problems of fault diagnosis have been studied. For example, Contant et al. (2004) considered the problem of ‘intermittent’ faults diagnosis for some fault events followed by corresponding ‘reset’ events, whereas Genc and Lafortune (2007) studied online fault diagnosis of modular dynamic systems modelled as place-bordered Petri nets and Yin and Li (2016a, 2018) investigated decentralized fault prognosis of partial-observed DESs with guaranteed performance bound and state-estimate protocols, respectively. Specially, Moreira et al. (2011), Viana et al. (2019) and Viana and Basilio (2019) proposed a new algorithm to verification of centralized or decentralized diagnosability, which could have lower computation complexity. Notice that many results referred above usually regard faults as unobservable events and depend on observed input strings to diagnose them based on the formal method. It motivates us to think over that whether we can transform unobservable fault events into observable changes of state transition and find a novel mathematical approach to analyse the problem of diagnosability.

Because of the social need for complex dynamic system security research, it is noteworthy to consider more complex DESs and model them as composite automata. On one hand, it is a meaningful task to analyse hybrid systems, which have wide applications in the real world. For example, the heating system in Sampath et al. (1995) and the exhaust gas recirculation system in Lin (1994)
are classical composite DESs. Particularly, in Roszkowska et al. (2018), the authors presented a composite automaton as the supervisor to ensure multiple mobile robots’ correct concurrent motion. On the other hand, the analysis of composite automata is more challenging for large scales. To our best of knowledge, it is common to model composite automata by enumeration or graph structure. Enumeration gives more detailed information about composite automata and graph structure clearly shows the composite relation. It would be a good job to combine two advantages in one model.

There is no doubt that a legible and simple mathematical structure is more conducive to the in-depth study of problems. Fortunately, among different existing methods, we find a new method, algebraic state space approach (ASSA), to analyse the problem of diagnosability for composite DESs. ASSA, established by semi-tensor product, is an effective tool to obtain an equivalent algebraic expression of a finite-value system. So far, there has been a number of works based on ASSA (Li et al., 2018; Lu et al., 2017), such as feedback control of Boolean control network (Lu et al., 2019; Zhu et al., 2020), stability analysis for some logical systems (Cheng et al., 2015; Han & Chen, 2018; Li back control of Boolean control network (Lu et al., 2019; So far, there has been a number of works based on equivalent algebraic expression of a finite-value system. The arrangement of this paper is shown as follows: Section 2 introduces some preliminaries about notations, semi-tensor product and algebraic structure of Moore automata. Section 3 establishes a novel system model and obtains the overall algebraic structure for composite automata. Section 4 contains three main works: definitions of diagnosability, construction of an observer matrix and verification of diagnosable faults. Section 5 shows the application of our results to the heating system and Section 6 summarizes works in this paper and provides our expectation.

2. Preliminaries

In this section, we introduce some preliminaries about necessary notations, semi-tensor product and algebraic structure of composite automata.

2.1. Notations

(1) \( \mathbb{R}^{m \times n} \) denotes the set of \( m \times n \) real matrices.
(2) \( A(i,j) \) denotes the element of matrix \( A \) at the position \((i,j)\).
(3) \( D := \{0, 1\} \).
(4) \( B \in \mathbb{R}^{m \times n} \) is called a Boolean matrix if \( B(i,j) \in D \) for all \( i \in \{1, 2, \ldots, m\} \) and \( j \in \{1, 2, \ldots, n\} \). Denote the set of all \( m \times n \) Boolean matrices by \( \mathcal{B}^{m \times n}_{\mathbb{B}} \).
(5) \( \text{col}(A) \) denotes the \( i \)-th column of matrix \( A \) and \( \text{col}(A) \) denotes the set of all columns of matrix \( A \).
(6) \( I_n \) is the identity matrix with dimension \( n \).
(7) \( \delta^n_n := \text{col}(I_n), i \in \{1, 2, \ldots, n\} \) and \( \delta^n_n := [0 0 \cdots 0]^T \).
(8) \( \Delta_n := \{ \delta^1_n, \delta^2_n, \ldots, \delta^n_n \} \) and \( \bar{\Delta}_n := \{ \delta^1_n, \delta^1_n, \ldots, \delta^n_n \} \).
(9) \( L \in B_{m \times n} \) is called a logical matrix (or generalized logical matrix) if \( \text{col}(L) \subseteq \Delta_m \) or \( \text{col}(L) \subseteq \bar{\Delta}_m \). Denote the set of \( m \times n \) logical matrices (or generalized logical matrices) by \( L_{m \times n} \) (or \( \bar{L}_{m \times n} \)).
(10) A matrix \( L \in \bar{L}_{m \times n} \) can be denoted by \( L = [\delta^1_m, \delta^2_m, \ldots, \delta^n_m] \) and briefly expressed as \( L = \delta_m[k_1 k_2 \ldots k_n] \) with \( k_i \in \{0, 1, \ldots, m\} \) and \( i \in \{1, 2, \ldots, n\} \).

2.2. Semi-tensor product

Semi-tensor product of matrices is a necessary tool to obtain the algebraic expression of a given system. We can find more information in Cheng and Qi (2010) and Cheng et al. (2012).

**Definition 2.1:** Assume that matrices \( M \in \mathbb{R}^{p \times q} \) and \( N \in \mathbb{R}^{s \times t} \). The semi-tensor product is defined as
\[
M \times N = (M \otimes \mathbb{I}_q^s) (N \otimes \mathbb{I}_t^s),
\]
where \( \alpha = \text{lcm}(q, s) \) denotes the least common multiple of \( q \) and \( s \), and \( \otimes \) represents the Kronecker product.

**Remark 2.1:** When \( q = s \), \( M \times N = MN \), i.e. ordinary matrix multiplication is a special case of semi-tensor product. And for \( M \in \mathbb{R}^{p \times q} \), \( N \in \mathbb{R}^{s \times t} \) and \( Q \in \mathbb{R}^{r \times l} \), we have
\[
(M \times N) \times Q = M \times (N \times Q).
\]

**Definition 2.2:** \( W_{(m, n)} \in L_{mn \times mn} \) is called a swap matrix if
\[
W_{(m, n)} = \delta_{mn}[1, 2m + 1, \ldots, (n - 1)m + 1,
2, m + 2, m + 2, \ldots, (n - 1)m + 2,
\ldots
m, 2m, 3m, \ldots, mn].
\]

**Remark 2.2:** For \( X \in \mathbb{R}^m \) and \( Y \in \mathbb{R}^n \), we have
\[
W_{(m, n)} \times X \times Y = Y \times X.
\]

**Remark 2.3:** Pseudo-commutation is a novel property of semi-tensor product. For a column vector \( Z \in \mathbb{R}^t \), we have
\[
Z \times A = (I_t \otimes A) \times Z.
\]

**Definition 2.3:** Define an operation as
\[
\tau_m(j_1, j_2) := (j_1 - 1)m + j_2.
\]

Figure 1. Examples of two kinds of automaton according to output function: (a) a Moore automaton and (b) a Mealy automaton.

**Remark 2.4:** For two columns \( x \in \Delta_m, y \in \Delta_m \), assume that \( x = \delta^1_m, y = \delta^2_m \). We have
\[
\delta^1_m \times \delta^2_m = \delta^1_m \times \delta^2_m,
\]
where \( j_{1,2} = \tau_m(j_1, j_2) \).

**Lemma 2.1:** Assume that \( x \in \Delta_k \), then
\[
x \times x = x^2 = M_{rk} \times x,
\]
where \( M_{rk} = \text{diag}(\delta^1_k, \delta^2_k, \ldots, \delta^n_k) \) is called a power-reducing matrix.

2.3. Algebraic structure of automata

\( G \) is a deterministic automaton, denoted by a six-tuple \( G := (X, \Sigma, Y, X_0, \delta, \lambda) \), where \( X \) is the set of states, \( \Sigma = \Sigma_o \cup \Sigma_u \) is the set of input events with observable events \( \Sigma_o \) and unobservable events \( \Sigma_u \), \( Y \) is the set of outputs and \( X_0 \in X \) is the initial state. \( \delta : X \times \Sigma \rightarrow X \) is the partial state transition function, where \( \delta(x, \sigma) = x' \) denotes that state \( x' \) can be reached from \( x \) by event \( \sigma \). For any \( s \in \Sigma^* \), \( \sigma \in \Sigma, \delta(x, \sigma) = \delta^e(x, s, \sigma), \) where \( \Sigma^* \) is the set of input strings upon \( \Sigma \) including \( \epsilon \). Note that there exists a projection \( P : \Sigma^* \rightarrow \Sigma_o^* \) defined as
\[
P(\sigma) = \begin{cases} P(s)\sigma, & \text{if } \sigma \in \Sigma_o \\ P(s), & \text{if } \sigma \in \Sigma_u. \end{cases}
\]

There are two kinds of automata according to output function: Moore automaton and Mealy automaton. Moore automata are automata with (state) outputs, where \( \lambda : X \rightarrow Y \), denoted by Figure 1(a) and Mealy automata are input/output automata, where \( \lambda : X \times \Sigma \rightarrow Y \), denoted by Figure 1(b). There are many physical systems modelled as Moore automata, where the output depends only on a current state (Cassandras & Lafortune, 2009). Notice that if any state from \( X \) can be an initial state and it has nothing to do with outputs, \( G \) can be abbreviated as \( G = (X, \Sigma, \delta) \). All in all, the number of variants for a deterministic automaton can vary with needed.
Given a deterministic Moore automaton $G = (X, \Sigma, Y, x_0, \delta, \lambda)$, we want to establish its algebraic expression with semi-tensor product. Suppose that $X = \{x_1, x_2, \ldots, x_n\}$, $\Sigma = \Sigma_0 = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$ and $Y = \{y_1, y_2, \ldots, y_p\}$. Identify $x_i \sim \delta_{i \alpha}^n$ ($i \in \{1, 2, \ldots, n\}$), $\sigma_j \sim \delta_{j \beta}^m$ ($j \in \{1, 2, \ldots, m\}$) and $y_k \sim \delta_{k \gamma}^p$ ($k \in \{1, 2, \ldots, p\}$). Now, define the state transition structure matrix $F_j \in \mathcal{L}_{n \times m}$ with an input event $\delta_{i m}^n$ as

$$F_j(i', i) = \begin{cases} 1, & \text{if } \delta_{i m}^n = \delta(\delta_{i m}^n, \delta_{i m}^n) \\ 0, & \text{if } \delta(\delta_{i m}^n, \delta_{i m}^n) \text{ is not defined} \end{cases} (9)$$

With all input events, obtain the transition structure matrix $F = [F_1, F_2, \ldots, F_m] \in \mathcal{L}_{n \times (nm)}$. Similarly, Moore output structure matrix $H \in \mathcal{L}_{p \times n}$ can be defined as

$$H(k, i) = \begin{cases} 1, & \text{if } \delta_{k i}^p = \lambda(\delta_{k i}^p) \\ 0, & \text{otherwise} \end{cases} (10)$$

Finally, an algebraic expression of the automaton is obtained as follows:

$$x(t + 1) = F \times \sigma(t) \times x(t),$$

$$y(t) = H \times x(t). (11)$$

For the example in Figure 1(a), $X = \{1, 2, 3\}$, $\Sigma = \{a, b, c, d\}$, $Y = \{01, 02, 03\}$. Identify $1 \sim \delta_{1 1}^1, 2 \sim \delta_{1 2}^1, 3 \sim \delta_{1 3}^1, a \sim \delta_{2 1}^1, b \sim \delta_{2 2}^1, c \sim \delta_{2 3}^1, d \sim \delta_{4 4}^1$ and $01 \sim \delta_{1 1}^2, 02 \sim \delta_{2 1}^2, 03 \sim \delta_{3 1}^2$. With the event $a \sim \delta_{1 1}^1$, we have $F_1 = \delta_{3}[2 \ 0 \ 0]$. With all events in $\Sigma$, we have $F = \delta_{3}[2 \ 0 \ 0 \ 3 \ 0 \ 0 \ 0 \ 0 \ 2 \ 0 \ 0 \ 1]$. Besides, $H = \delta_{3}[1 \ 2 \ 3]$.\

3. System model

In this section, we want to introduce algebraic expressions of two kinds of composite automata and discuss their normative and faulty structures.

There is no denying that the failure of a single widget can paralyse an entire complex system. For example, as a parallel composite automaton by the valve and throttle, the exhaust gas recirculation system, depicted by Figure 2, may break down when the valve is trapped in open or closed. In this case, we can first focus on analysing the evolutionary process of component automata and then take advantage of the output information of composite automata to diagnosis.

It is reasonable to make the following assumptions about given systems:

- Any fault is irreparable. Therefore, the faulty states cannot recover to normal states.
- Any fault is constant. Therefore, the faulty states cannot transit to other category of faulty states.

Consider two deterministic automata

$$G^1 = (X^1, \Sigma^1, x_0^1, \delta_1^1) \quad \text{and} \quad G^2 = (X^2, \Sigma^2, x_0^2, \delta_2^2),$$

where $G^1$ works normally all the time and $G^2$ has several possible faults.

Assume that $X^1 = \{x_1^1, x_2^1, \ldots, x_{m_1}^1\}$, $\Sigma^1 = \Sigma_0^1 = \{\sigma_1^1, \sigma_2^1, \ldots, \sigma_{m_1}^1\}$. We can get the transition structure matrix of $G^1$ with (9), depicted by $F^{(1)} \in \mathcal{L}_{n_1 \times (nm_1)}$. For $G^2$, assume that $X^2 = \{x_1^2, x_2^2, \ldots, x_{m_2}^2\}$, $\Sigma^2 = \{\sigma_1^2, \sigma_2^2, \ldots, \sigma_{m_2}^2, \sigma_1, \sigma_2, \ldots, \sigma_4\}$ with $\Sigma_0^2 = \{\sigma_1^2, \sigma_2^2, \ldots, \sigma_{m_2}^2\}$ and $\Sigma_0 = \Sigma_0^2 = \{\sigma_1, \sigma_2, \ldots, \sigma_4\}$. Define the faulty reach of a state $x \in X^2$ as

$$X^2_f(x^2) = \{x'^2 \in X^2 | \exists \sigma_{f_i} \in \Sigma_f, s.t. x'^2 = \delta(x^2, \sigma_{f_i})\}.$$ 

Obtain that $X^2_f = \bigcup_{x \in X^2} X^2_f(x^2)$ and $X^2_{f_0} = X^2 - X^2_f$. Define the faulty reach of a fault $\sigma_{f_i} \in \Sigma_f$ as

$$X(\sigma_{f_i}) = \{x^2 \in X^2_f | \exists x \in X^2_f \text{ s.t. } x^2 = \delta(x^2, \sigma_{f_i})\}.$$ 

There are two available constructions for $G^2$: a normative system $G^N_2 = (X^2, \Sigma_0^2, x_0^2, \delta_2^2)$ and a faulty one $G^F_2 = (X^2, \Sigma^2, x_0^2, \delta_2^2)$, where $|X^2_N| = |X^2_f| = n_2$. In this case, we can get the transition structure matrices of $G^N_2$ and $G^F_2$ with (9), depicted by $F^{(2)} \in \mathcal{L}_{n_2 \times (nm_2)}$ and $F^{(2)}_f \in \mathcal{L}_{n_2 \times (nm_2)}$, respectively.

Remark 3.1: If $|X^2_N| \neq |X^2_f|$, we could add suppositional states into $X^2$ to resolve it.

Remark 3.2: There exists a key conversion, denoting the faults by accessible changes of state transition using definitions of the faulty reach.

Remark 3.3: It is significant to relabel elements in sets $X^2_f$ and $X^2_{f_0}$ which will be presented as a norm later.

There are two common operations on automata: product, denoted by $\times$ in Figure 3(a) and parallel, denoted by $\parallel$ in Figure 3(b).
Then, we can get
\[
\text{Identify } (x^1, x^2) \sim \delta_{n_1, n_2}, \text{ where } \delta_{n_1, n_2} = \delta_{n_1} \times \delta_{n_2} \text{ and } i_{1,2} = t_{\sigma_1}(i_1, i_2). \text{ Besides, assume that } \Sigma = \Sigma_1 \cup \Sigma_2 = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}. \text{ We can take the following approach to obtain the algebraic structure of a composite automaton with parallel operation.}
\]

First, extend transition structure matrices \(F^{(1)}, F^{(2)}\) as \(\hat{F}^{(1)} \in \tilde{L}_{m \times n_2 m}, \hat{F}^{(2)} \in \tilde{L}_{n_2 \times n_2 m}, \text{ respectively, where} \)
\[
\hat{F}^{(i)}_j = \begin{cases} F^{(i)}_j, & \text{if } \sigma_j \in \Sigma_i \\ 0, & \text{otherwise} \end{cases} \quad i = 1, 2; j = 1, 2, \ldots, m \tag{13}
\]
Then, we can get
\[
L^{(i)}(t + 1) = \hat{F}^{(i)} \times \sigma(t) \times L^{(i)}(t) \tag{14}
\] where \(\sigma(t) \in \Sigma\).
Following,
\[
x(t + 1) = x^{(1)}(t + 1) \times x^{(2)}(t + 1) = \hat{F}^{(1)} \times \sigma(t) \times x^{(1)}(t) \times \hat{F}^{(2)} \times \sigma(t) \times x^{(2)}(t) = \hat{F}^{(1)} \times (I_{n \times m} \otimes \hat{F}^{(2)}) \times \sigma(t) \times x^{(2)}(t) = \hat{F}^{(1)} \times (I_{n \times m} \otimes \hat{F}^{(2)}) \times W_{[m, n_2 m]} \times (\sigma(t))^2 \times x^{(1)}(t) \times x^{(2)}(t) = \hat{F}^{(1)} \times (I_{n \times m} \otimes \hat{F}^{(2)}) \times W_{[m, n_2 m]} \times M_{r, m} \times \sigma(t) \times x(t) = \hat{M} \times x(t) \times \sigma(t),
\]
that is, the state transition structure matrix of the parallel composite automaton by \(G^{1}\) and \(G^{2}\) is
\[
x(t + 1) = \hat{M} \times x(t) \times \sigma(t), \tag{15}
\]
where \(\hat{M} = \hat{F}^{(1)} \times (I_{n \times m} \otimes \hat{F}^{(2)}) \times W_{[m, n_2 m]} \times M_{r, m} \times W_{[n_2, m]}\).

Similarly, we can obtain its by \(G^{1}\) and \(G^{2}\) as
\[
x^{(i)}(t + 1) = \hat{M} \times x^{(i)}(t) \times \sigma(t), \tag{16}
\]
where \(\hat{M} = \hat{F}^{(1)} \times (I_{n_1 \times m} \otimes \hat{F}^{(2)}) \times W_{[m, n_2 m]} \times M_{r, m} \times W_{[n_2, m]}\).

It is helpful to take output information of the composite automaton into consideration. First, we present a definition of compatible states between \(X^{2}_f\) and \(X^{2}_f\). It is worth noticing that \(x^{2}_1 \in X^{2}_f\), \(x^{2}_2 \in X^{2}_f\), respectively, where \(x^{2}_1\) is the faulty one.

In fact, the above definition presents a norm about how to relabel elements in sets \(X^{2}_f\) and \(X^{2}_f\). For \(x^i\) and \(x^i\) constructed by compatible states \(x^i\) and \(x^i\), we have \(\delta x^i = \delta x^i\). Without confusion, we assume that states \(x^i\) and \(x^i\) have the same number afterwards. Therefore, we obtain an overall algebraic structure of the parallel composite automata by \(G^{1}\) and \(G^{2}\) as
\[
x(t + 1) = \hat{M} \times x(t) \times \sigma(t), \tag{17}
\]
y(t) = H \times x(t),
where \(\hat{M} = [\hat{M} \ 1, \ 2 \ 1, \ 2, \ldots, m \ \\ \ 0_{n_1 \times n_2} \ 1, \ 2 \ 1, \ 2, \ldots, m \]
For the product operation, extend the transition structure matrices \(F^{(1)}\) and \(F^{(2)}\) as \(\hat{F}^{(1)} \in \tilde{L}_{m \times n_2 m}, \hat{F}^{(2)} \in \tilde{L}_{n_2 \times n_2 m}\), where
\[
\hat{F}^{(i)}_j = \begin{cases} F^{(i)}_j, & \text{if } \sigma_j \in \Sigma_i \\ 0_{n_1 \times n_2}, & \text{otherwise} \end{cases} \quad i = 1, 2; j = 1, 2, \ldots, m
\]
A composite automaton with product of \(G^{1}\) and \(G^{2}\) can be expressed as
\[
x(t + 1) = \hat{M} \times x(t) \times \sigma(t) \tag{19}
\]
where \(\hat{M} = \hat{F}^{(1)} \times (I_{n_1 \times m} \otimes \hat{F}^{(2)}) \times W_{[m, n_2 m]} \times M_{r, m} \times W_{[n_2, m]}\).
composite automaton with product or parallel operation is

\[ x(t + 1) = M \times x(t) \times \sigma(t) \quad (20) \]

where \( M = F^{(1)} \times (I_{m,n} \otimes F^{(2)}) \times W_{[m,n]} \times M_{r,m} \times W_{[n,m]} \).  

**Remark 3.5:** Notice that algebraic expressions of composite automata obtained by product or parallel composition are identical in structure thanks to the convenience of semi-tensor product for model description. This work has a great advantage in obtaining a simpler mathematical structure of composite automata. Afterwards, we will analyse the parallel composite Moore automaton. Results obtained would also be applied to the product composite automata without doubt.

**Example 3.1:** Consider the above system: the exhaust gas recirculation system depicted by Figure 2. Assume that the throttle is \( G^1 = (X^1, \Sigma^1, x_0^1, \delta^1) \), denoted by Figure 4(a), where \( X^1 = \{x_1^1, x_2^1\} \), \( \Sigma^1 = \{\sigma^1_1, \sigma^1_2\} \) and the valve is \( G^2 = (X^2, \Sigma^2, x_0^2, \delta^2) \), denoted by Figure 4(b), where \( X^2 = \{x_1^2, x_2^2, x_3^2, x_4^2\} \), \( \Sigma^2 = \{\sigma^2_1, \sigma^2_2, \sigma^2_3, \sigma^2_4\} \).

It is easy to get \( F^{(1)} = \delta_2[2 \ 2 \ 1 \ 1] \). For \( G^2 \), we get compatible states \( x_1^2, x_2^2, x_3^2, x_4^2 \). Therefore, we have \( F^{(2)} = \delta_2[2 \ 2 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] \) and \( F_f^{(2)} = \delta_2[1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] \). Considering the parallel composition of the throttle and the valve, assume that \( \Sigma = \{\sigma_1 = \sigma^1_1, \sigma_2 = \sigma^2_2, \sigma_3 = \sigma^1_1, \sigma_4 = \sigma^1_2\} \). Then, the extended matrices of \( F^{(1)}, F^{(2)} \) and \( F_f^{(2)} \) are \( \hat{F}^{(1)} = \delta_2[2 \ 2 \ 1 \ 1 \ 1 \ 1 \ 1] \); \( \hat{F}^{(2)} = \delta_2[2 \ 2 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] \); \( \hat{F}_f^{(1)} = \delta_2[1 \ 2 \ 1 \ 2 \ 1 \ 2 \ 1 \ 2] \). Next, consider the percentage of nitrogen oxides as output information. Only when both the valve and the throttle are open, the content of nitrogen oxides would decrease. Assume that \( Y = \{y_1 = \text{‘decrease’}, y_2 = \text{‘don’t decrease’}\} \). We can obtain two available constructions for the composite automaton as Figure 5 and their algebraic structures as: \( \hat{M} = \delta_4[2 \ 1 \ 3 \ 1 \ 2 \ 1 \ 4 \ 2 \ 4 \ 3 \ 3 \ 1 \ 4 \ 3 \ 4 \ 2] \), \( \hat{M}_f = \delta_4[1 \ 1 \ 3 \ 1 \ 2 \ 2 \ 4 \ 2 \ 3 \ 3 \ 1 \ 4 \ 3 \ 4 \ 2] \), \( H = \delta_2[2 \ 2 \ 1 \ 1 \ 1] \).

### 4. Fault diagnosis analysis

In this section, we propose the definition of diagnosability based on algebraic structures established in Section 3 and propose a matrix approach to verify whether a fault (respectively, a given system) is diagnosable.

#### 4.1. Definition of diagnosability

In diagnosability analysis, a fault is diagnosable if it can be detected certainly with a finite delay occurrence and a
system is diagnosable if any type of faults can be distinguished with observed information. In Section 3, we have transformed unobservable fault events into observable changes of state transition, which motivates us to compare evolutionary process of the normative system with the faulty one. Considering an initial state \(x(1)\), with an input string \(s = \sigma(1)\sigma(2)\cdots\sigma(t)\), we have

\[
x(t + 1) = \hat{M}^t x(1) \times \sigma(1) \times \sigma(2) \cdots \times \sigma(t),
\]

and

\[
y(t) = H \times \hat{M}^{t-1} x(1) \times \sigma(1) \times \sigma(2) \cdots \times \sigma(t-1),
\]

First, we define a set of all affected states as \(\bar{X} := \{x \in \mathcal{X} : \exists \sigma \in \Sigma, \text{s.t.} \hat{M} \times x \times \sigma \neq \hat{M} \times x \times \sigma\}\), and the corresponding inputs as \(\bar{\Sigma}(x) := \{\sigma \in \Sigma : \hat{M} \times x \times \sigma \neq \hat{M} \times x \times \sigma\}\). Notice that when \(\bar{X} \neq \emptyset\) and \(\bar{\Sigma} \neq \emptyset\), there are actually some faults needing to be detected and isolated in the system, that is, \(\bar{X}\) and \(\bar{\Sigma}\) denote the condition estimation for faults. Besides, we define a set of detectable states for the \(l\)th fault as

\[
C_l := \{(x, x') \in \mathcal{X} \times \mathcal{X} : \exists \sigma \in \Sigma, \text{s.t.} \hat{M}^l \times x \times s, \hat{M}^l \times x \times s \in C_l\},
\]

where \(x \in \mathcal{X} \times \mathcal{X} \times \sigma_l\) denotes \(x \in \mathcal{X}\) and \(x \notin \mathcal{X} \times \sigma_l\).

Now, we give definitions of diagnosability for a fault and a given system.

**Definition 4.1**: Given a composite Moore automaton (17), the \(l\)th fault is said to be diagnosable if \(\forall x \in \mathcal{X} \times \mathcal{X} \times \sigma_l\), \(\exists s = \sigma(1) \times \sigma(2) \times \cdots \times \sigma(t), \text{s.t.} \hat{M}^l \times x \times s, \hat{M}^l \times x \times s \in C_l\).

**Remark 4.1**: In Definition 4.1, the condition \(\forall x \in \mathcal{X} \times \mathcal{X} \times \sigma_l\) guarantees that all possible changes of state transition by the \(l\)th fault could be considered. The result \((\hat{M}^l \times x \times s, \hat{M}^l \times x \times s) \in C_l\) denotes the \(l\)th fault would be detected with a finite delay \(t-1\) after its occurrence. Any type of fault could be distinguished and then the given system is diagnosable. The definition is presented in the following.

**Definition 4.2**: Given a composite Moore automaton (17), the system is said to be diagnosable, if \(\forall x \in \mathcal{X}, \exists l \in \{1, 2, \ldots, q\}, \exists s = \sigma(1) \times \sigma(2) \times \cdots \times \sigma(t), \text{s.t.} \hat{M}^l \times x \times s, \hat{M}^l \times x \times s \in C_l\),

where \(\sigma(1) \in \bar{\Sigma}(x)\).

There is an example to illustrate the result.

**Example 4.1**: Consider the above example in Section 3 continually. We can obtain that \(\mathcal{X} = \{x_1, x_2, x_3, x_4\}\), where \(\bar{\Sigma}(x_1) = \sigma_1, \bar{\Sigma}(x_2) = \sigma_2, \bar{\Sigma}(x_3) = \sigma_1, \bar{\Sigma}(x_4) = \sigma_2\).

For the first fault: the valve is trapped in open, we have

\[
\mathcal{X}(\sigma_1) = \{x_2, x_4\}; \quad C_1 = \{(x_1, x_4), (x_3, x_4)\}.
\]

For the second fault: the valve is trapped in close, we have

\[
\mathcal{X}(\sigma_2) = \{x_1, x_3\}; \quad C_2 = \{(x_4, x_1), (x_4, x_3)\}.
\]

According to the definition of diagnosability, we have:

- for \(x_1 \in \bar{X} \cap \mathcal{X}(\sigma_2), \exists s = \sigma_1 \times \sigma_3, \text{s.t.} (\hat{M}^2 \times x_1 \times s, \hat{M}^2 \times x_1 \times s) \in C_2\);
- for \(x_3 \in \bar{X} \cap \mathcal{X}(\sigma_2), \exists s = \sigma_1, \text{s.t.} (\hat{M} \times x_3 \times s, \hat{M} \times x_3 \times s) \in C_2\).

Then, the second fault is diagnosable.

- for \(x_2 \in \bar{X} \cap \mathcal{X}(\sigma_1), \exists s = \sigma_2 \times \sigma_3, \text{s.t.} (\hat{M}^2 \times x_2 \times s, \hat{M}^2 \times x_2 \times s) \in C_1\);
- for \(x_4 \in \bar{X} \cap \mathcal{X}(\sigma_1), \exists s = \sigma_2, \text{s.t.} (\hat{M} \times x_4 \times s, \hat{M} \times x_4 \times s) \in C_1\).

Therefore, the first fault is diagnosable. In conclusion, the system could be diagnosed with the mostly 1 step delay.

**Remark 4.2**: Notice that in the work of Lin (1994), the exhaust gas recirculation system is not diagnosable since no control can diagnose whether the valve is trapped in close. In another profound definition of diagnosability proposed by Sampath et al. (1995), the result is also different from Example 4.1. Therefore, the proposed diagnosability in this paper is a novel notion and provides a new opportunity to fault diagnosis when existing criteria fail.

### 4.2. Matrix observer

In this section, we would construct matrix observers of given systems, which contain requisite evolutionary information.

Notice that there are many zero elements in \(\hat{M}^l; \hat{M}_f^l\) and \(H\) occupying the storage space. In fact, we care more about the position of non-zero elements in these structure matrices. To exhibit effective information more
clearly and avoid occupying the storage space, we take the following method to reconstruct structure matrices $\hat{M}_r, \hat{M_f}$ and $H$.

Firstly, for $\hat{M} = \delta_n[\alpha_{11} \cdots \alpha_{1m} \cdots \alpha_{n1} \cdots \alpha_{nm}]$, we should know that $\alpha_{ij} = \hat{f}$ when $x_\tau = \delta(x_i, \sigma_j)$, where $i, j \in \{1, 2, \ldots, n\}$, and $j \in \{1, 2, \ldots, m\}$, that is, $\alpha_{ij}$ denotes the position information what we care about.

Then, we reconstruct $\hat{M}$ as

$$P = \begin{bmatrix}
\alpha_{11} & \cdots & \alpha_{1j} & \cdots & \alpha_{1m} \\
\alpha_{21} & \cdots & \alpha_{2j} & \cdots & \alpha_{2m} \\
\vdots & \cdots & \vdots & \cdots & \vdots \\
\alpha_{n1} & \cdots & \alpha_{nj} & \cdots & \alpha_{nm}
\end{bmatrix},$$

where $P \in \mathbb{R}^{n \times m}$. Here, we use the row, column and element values in $P(i,j)$ to represent three types of information of $\alpha_{ij}$, i.e. the starting state, the operated event and the reached state in a state transition. Then, we reconstruct $\hat{M_r}$ as $P_r \in \mathbb{R}^{n \times m}$ similarly.

Next, for $H = \delta_p[\beta_1 \beta_2 \cdots \beta_n] \in \Lambda_p \times n$, define $Q^{(1)} = [\beta_1 \beta_2 \cdots \beta_n]^T \in \mathbb{R}^{n \times 1}$ and

$$Q^{(t+1)} = \begin{bmatrix}
Q^{(1)}(\alpha_{11},1) & \cdots & Q^{(1)}(\alpha_{11},m^t-1) & \cdots & Q^{(1)}(\alpha_{1m},1) \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
Q^{(1)}(\alpha_{1m},1) & \cdots & Q^{(1)}(\alpha_{1m},m^t-1) & \cdots & Q^{(1)}(\alpha_{nm},1) \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
Q^{(1)}(\alpha_{nm},1) & \cdots & Q^{(1)}(\alpha_{nm},m^t-1)
\end{bmatrix},$$

where $Q^{(t+1)} \in \mathbb{R}^{n \times m}$. In fact, $Q^{(t+1)}$ is reconstructed from $H \times \hat{M_f}$, which would be proved in Lemma 4.1.

**Lemma 4.1:** Assume that $x(1) = x_i$, $i \in \{1, 2, \ldots, n\}$, there exists an input string $s = \sigma(1)\sigma(2)\cdots \sigma(t), s.t. x_\tau = \delta(x_i, s)$ and $y_k = \lambda(x_\tau)$. We have

$$Q^{(t+1)}(i,j) = k,$

where $j = \tau_m(\tau_m(j_1, j_2), \ldots, j_t)$.

**Proof:** For initial state $x(1) = x_i$, $i \in \{1, 2, \ldots, n\}$, there exists an input string $s = \sigma(1)\sigma(2)\cdots \sigma(t), s.t. x_\tau = \delta(x_i, s)$ and $y_k = \lambda(x_\tau)$. According to the definition of the transition structure matrix, we have its equivalent algebraic expression as

$$\delta_n^f = \hat{M}_r \times \delta_n^p \times \delta_m^l,$$

$$\delta_p^f = H \times \delta_n^p$$

where $\delta_m^l = \delta_m^l \times \delta_m^l \cdots \times \delta_m^l$. Note that $j = \tau_m(\tau_m(j_1, j_2), \ldots, j_t)$ is obvious. Therefore, we just need to prove that $Q^{(t+1)}(i,j) = k$ is equivalent to (21). Proof by induction.

When $t = 1$, $\delta_n^f = \hat{M} \times \delta_n^l \times \delta_m^p \times \delta_m^p = H \times \delta_n^l$. We have

$$Q^{(2)}(i,j) = Q^{(1)}(\alpha_{ij},1) = Q^{(1)}(i,1) = k.$$

The conclusion is established. Suppose that the conclusion maintains until $t - 1$. Then

$$H \times \hat{M}_r \times \sigma(1) \times \sigma(2) \times \cdots \times \sigma(t) = \hat{M}_r \times \sigma(1) \times \sigma(2) \times \cdots \times \sigma(t) = \delta_n^{(t-1)}(\alpha_{ij}, j'),$$

where $\delta_m^{i-1} = \delta_m^{i} \times \delta_m^{i} \cdots \times \delta_m^{i}$. Notice that $\delta_m^{j} = \delta_m^{i} \times \delta_m^{i} \cdots \delta_m^{i}$, and according to the definition of $Q^{(t+1)}$,

$$Q^{(t+1)}(\alpha_{ij}, j') = Q^{(t+1)}(i,j)$$

We construct the matrix $Q^{(t+1)}$ by label, which plays the role of an observer. On the other hand, we can obtain the labelled matrix $Q^{(t+1)}$ in the same way to reduce the occupation by redundant zero elements in $H \times \hat{M}_r$.

**Remark 4.3:** From Lemma 4.1, we know that matrix $Q^{(t+1)}$ describes the evolutionary process for a given system. The labelled number of the row depicts information for initial states and the columns denote all possible inputs with length of $t$. The element value of $Q^{(t+1)}(i,j)$ shows the obtained output from initial state $x_i$ with input $s$. The observer could also be applied to other properties’ analysis, such as observability, controllability, detectability and so on.

### 4.3. Verification of diagnosability

In this section, we will apply the matrix observer to the verification of diagnosability and present an algorithm to illustrate the main result in this paper.

First, for the set of effected states $\tilde{X}$ and the corresponding inputs $\hat{X}^c(x)$, we give the equivalent notion from labelled matrices $P$ and $P_f$. We define an ordered set as $\mathcal{R} := \{(i,j) | P(i,j) \neq P_f(i,j), i \in \{1, 2, \ldots, n\}, j \in \{1, 2, \ldots, m\}\}$. It is easy to obtain that $(i,j) \in \mathcal{R} \iff \tilde{x}_i \in \tilde{X} \land \hat{X}^c(x_i) = \sigma_j$, i.e. $\mathcal{R}$ is the ordered set of $\tilde{X}$ and $\hat{X}^c$, only containing the labelled numbers. Besides, define other two labelled sets for the $l$th fault as $\mathcal{X}_l := \{(i,j) | \tilde{x}_i = \chi_{(\sigma_l)} \}$ and $\mathcal{Y}_l := \{(k,l) | \forall (x, x') \in C_t, H \times x = y_k \land H \times x' = y_k \}$. Note that $\mathcal{X}_l$ represents the labelled set of faulty reach from $\sigma_l$ and $\mathcal{Y}_l$ depicts the labellings of corresponding outputs from detectable states in $C_t$.

Now, we give the necessary and sufficient condition of a diagnosable fault.
Theorem 4.1: For a composite automaton (17), the following facts are equivalent:

1. The lth fault is diagnosable.
2. \( \forall i \in \mathcal{X}_l \land (i,j_l) \in \mathcal{R}_l, \exists t \in \mathbb{N}^+, \)
   \[
   (Q_t^{l+1}(i,j),Q_t^{l+1}(i,j)) \in \mathcal{Y}_l,
   \]
   where \( j = \tau_m(\tau_m(j_1,j_2),\ldots,j_l). \)

Proof: 

'\( (1) \Rightarrow (2) ' \) Assume that the lth fault is diagnosable. Then, \( \forall x \in \hat{X} \land \mathcal{X}(\sigma_l), \exists s = \sigma(1) \land \sigma(2) \land \cdots \land \sigma(t), \) \( (\hat{M}^t \times x \times s,\hat{M}^t_\ell \times x \times s) \in \mathcal{C}_l. \) According to Lemma 4.1 and definitions of \( \mathcal{R}_l, \mathcal{X}_l \) and \( \mathcal{Y}_l, \) we have \( \forall i \in \mathcal{X}_l \land (i,j_l) \in \mathcal{R}_l, \exists t \in \mathbb{N}^+, (Q_t^{l+1}(i,j),Q_t^{l+1}(i,j)) \in \mathcal{Y}_l, \)

'\( (2) \Rightarrow (1) ' \) Proof by contradiction. Suppose that the lth fault is undiagnosable. Then, \( \exists x \in \hat{X} \land \mathcal{X}(\sigma_l), \forall s = \sigma(1) \land \sigma(2) \land \cdots \land \sigma(t), \) \( (\hat{M}^t \times x \times s,\hat{M}^t_\ell \times x \times s) \notin \mathcal{C}_l. \) i.e. \( \exists i \in \mathcal{X}_l \land (i,j_l) \in \mathcal{R}_l, \forall t \in \mathbb{N}^+, (Q_t^{l+1}(i,j),Q_t^{l+1}(i,j)) \notin \mathcal{Y}_l, \) where \( j = \tau_m(\tau_m(j_1,j_2),\ldots,j_l). \) It is contrary to the condition. \qed

In fact, Definition 4.1 and Theorem 4.1 are in line. To verify whether a kind of fault is diagnosable, we should consider all possible changes reflected by \( i \in \mathcal{X}_l \land (i,j_l) \in \mathcal{R}_l \) and draw the final conclusion from detectable outputs \( \mathcal{Y}_l. \) The following Theorem 4.2 shows that any change in \( \mathcal{R}_l \) could be detected and isolated when the given system is diagnosable.

Theorem 4.2: For a composite automaton (17), the following facts are equivalent:

1. The system is diagnosable.
2. \( \forall (i,j) \in \mathcal{R}, \exists I \in \{1,2,\ldots,q\}, \exists t \in \mathbb{N}^+, \)
   \[
   (Q_t(i,j),Q_t(i,j)) \in \mathcal{Y}_l,
   \]
   where \( j = \tau_m(\tau_m(j_1,j_2),\ldots,j_l). \)

Proof: It is obvious that the given system could be diagnosed when each fault is diagnosable. \qed

Note that the above theorems present a matrix approach to determine whether a fault (respectively, a composite automaton) is diagnosable. It should be pointed out that the finite delay \( t_l - 1 \) should be limited strictly since that once a fault happens, the longer the delay, the greater the system loss. Next, we glue the matrix approach as the following algorithm.

Algorithm: A given system, modelled as a discrete event system, has more than one component. Every component can be modelled as a deterministic automaton. We take the parallel composite Moore automaton constructed by \( G_1 \) and \( G_2 \) for example. Transform the faulty events into changes of state transition as in Section 2. We can take the following steps to verify whether the lth fault is diagnosable, and then, the diagnosability of a given system could be ascertained according to the result of each type of fault.

- Step 1: Transform faulty events into changes of state transition in the automaton with faults, and then, establish the state transition structure matrix for every component automaton by (9).
- Step 2: Assume that \( \Sigma = \Sigma^1 \cup \Sigma^2. \) Obtain the extended transition structure matrices by (13).
- Step 3: Obtain the overall algebraic structure for the composite automata as (17).
- Step 4: For the lth fault, obtain the corresponding sets \( \mathcal{X}_l \) and \( \mathcal{Y}_l. \) And reconstruct \( \hat{M} \times \hat{M}_l \) as \( P \) and \( P_l, \) respectively. Comparing \( P \) with \( P_l, \) obtain another set \( \mathcal{R}. \)
- Step 5: Construct matrix observers \( Q^{l+1}_t \) and \( Q^{l+1}_t. \) Meanwhile, verify whether for \( \forall (i,j) \in \mathcal{X}_l \land (i,j_l) \in \mathcal{R}_l, \exists t_l \) \( (\hat{M}^{Q^{l+1}_t}(i,j),\hat{M}^{Q^{l+1}_t}(i,j)) \in \mathcal{Y}_l, \) where \( j = \tau_m(\tau_m(j_1,j_2),\ldots,j_l). \) If so, the lth fault can be diagnosed with a finite delay \( t_l - 1. \) Otherwise, it is undiagnosable.

Remark 4.4: Consider the time complexity for the above approach to verify the diagnosability of the lth fault with a finite delay \( t_l - 1. \) Table 1 shows the dimensions of all needed matrices in Algorithm 1. The maximum dimension is \( |\mathcal{X}| \times |\Sigma| \) for \( Q^{l+1}_t \) and \( Q^{l+1}_t. \) in Step 5. Moreover, in the worst case, \( |\mathcal{X}| = |\mathcal{Y}| \times (|\mathcal{Y}| - 1). \) The complexity of verify whether \( (Q^{l+1}_t(i,j),Q^{l+1}_t(i,j)) \in \mathcal{Y}_l \) is linear. Therefore, the time complexity of Algorithm 1 is \( O(|\mathcal{X}| \times |\Sigma| \times |\mathcal{Y}|^2 \times |\Sigma|). \) Furthermore, the verification of diagnosability for a given system is \( O(|\mathcal{X}| \times |\Sigma| \times |\mathcal{Y}|^2 \times |\Sigma|). \)

| Table 1. Dimensions of all needed matrices in Algorithm 1. |
|------------------------|------------------------|
| Matrix           | Dimension               |
| \( F^{(l)} \)    | \( |\mathcal{X}| \times |\mathcal{X}| \times |\Sigma| \) |
| \( P^{(l)} \)    | \( |\mathcal{X}| \times |\mathcal{Y}| \times |\Sigma| \) |
| \( M, \hat{M} \) | \( |\mathcal{X}| \times |\mathcal{Y}| \times |\Sigma| \) |
| \( P, P_l \)     | \( |\mathcal{X}| \times |\Sigma| \) |
| \( H \)          | \( |\mathcal{Y}| \times |\mathcal{X}| \) |
| \( Q^{(l)} \)    | \( |\mathcal{X}| \times |\Sigma| \) |
| \( Q^{(l+1)}, \hat{Q}^{(l+1)} \) | \( |\mathcal{X}| \times |\Sigma| \times |\Sigma| \times |\Sigma| \) |
5. Example

Heating system, modelled as a DES, is a typical example for fault diagnosis. It is a parallel composite automaton, constituted by the pump, valve and controller, which are shown in Figure 6. The component valve contains two fault states SC (for ‘Stuck-closed’) and SO (for ‘Stuck-open’), which are taken by faulty events STUCK-CLOSED and STUCK-OPEN. Besides, there are two sensors, a valve flow sensor and a pump pressure sensor. The valve flow sensor has two possible values: Flow (F) or No Flow (NF) and the pump pressure sensor also has two possible values: Positive Pressure (PP) or No Pressure (NP). The parallel composition of three automata decides together outputs of two sensors. It is of particular interest that the flow sensor will return F only when the valve is open and the pump is on.

The pump, valve and controller are modelled as three deterministic automata, respectively. The pump is $G^1 = (X^1, \Sigma^1, \delta_1)$, where $X^1 = \{x_1 = \text{POFF}, x_2 = \text{PON}\}$, $\Sigma^1 = \{\sigma_1 = \text{STOP} - \text{PUMP}'\}$, $\sigma_2 = \text{START} - \text{PUMP}'$. The valve is $G^2 = (X^2, \Sigma^2, \delta_2)$, where $X^2 = \{x_2 = \text{VC}, x_2 = \text{VO}, x_2 = \text{SC}, x_2 = \text{SO}\}$, $\Sigma^2 = \{\sigma_1 = \text{CLOSE} - \text{VALVE}'\}$, $\sigma_2 = \text{OPEN} - \text{VALVE}'$, $\sigma_3 = \text{STUCK} - \text{CLOSE}'$, $\sigma_4 = \text{STUCK} - \text{OPEN}'$. The controller is $G^3 = (X^3, \Sigma^3, \delta_3)$, where $X^3 = \{x_3 = \text{C1}, x_3 = \text{C2}, x_3 = \text{C3}, x_3 = \text{C4}\}$, $\Sigma^3 = \{\sigma_1 = \text{STOP} - \text{PUMP}'\}$, $\sigma_2 = \text{START} - \text{PUMP}'$, $\sigma_3 = \text{CLOSE} - \text{VALVE}'$, $\sigma_4 = \text{OPEN} - \text{VALVE}'$. Besides, assume that the output of the parallel composite automaton is $Y = \{y_1 = (PP, F), y_2 = (PP, NF), y_3 = (NP, NF)\}$. Now, we apply the algorithm in Section IV to diagnose the faults STUCK-CLOSED and STUCK-OPEN within a 2-step delay.

Step 1. For the valve, we have the compatible states $x_1^2$ and $x_2^2$, $x_3^2$ and $x_4^2$. Therefore, we can obtain its normative and faulty constructions, depicted by Figure 7. Next, get the transition structure matrices of three automata.

$$F^{(1)} = \delta_2[1 \ 1 \ 2 \ 2], \quad F^{(2)} = \delta_2[1 \ 1 \ 2 \ 2],$$
$$F^{(3)} = \delta_4[0 \ 0 \ 4 \ 0 \ 3 \ 0 \ 0 \ 0 \ 0 \ 0], \quad F^{(4)} = \delta_2[1 \ 2 \ 0 \ 0]$$

Step 2. Assume that $\Sigma = \{\sigma_1 = \text{STOP} - \text{PUMP}'\}$, $\sigma_2 = \text{START} - \text{PUMP}'$, $\sigma_3 = \text{CLOSE} - \text{VALVE}'$, $\sigma_4 = \text{OPEN} - \text{VALVE}'$. Then, the extended transition structure matrices for $F^{(1)}, F^{(2)}, F^{(3)}$ and $F^{(4)}$ are

$$\hat{F}^{(1)} = \delta_2[1 \ 1 \ 2 \ 2 \ 1 \ 1 \ 2], \quad \hat{F}^{(2)} = \delta_2[1 \ 2 \ 1 \ 2 \ 1 \ 1 \ 2],$$
$$\hat{F}^{(3)} = \delta_2[1 \ 2 \ 1 \ 2 \ 1 \ 2 \ 1], \quad \hat{F}^{(4)} = F^{(3)}.$$
\( \delta \)

\[ x(t + 1) = \tilde{M} \times f(t) \times x(t) \times \sigma(t), \]
\[ y(t) = H \times x(t), \]

where \( \tilde{M} = [\hat{M} \ \hat{M}_f] \) with \( \hat{M} = \delta_{16}[0000 0110 0040000 \]
\[ 0100006 01500800000100040110040000 \]
\[ 09000014 015008000000900001401500800 \]
\[ 000130], \hat{H} = \delta_3[3333333322221111]. \]

Step 4. For the first fault: STUCK-CLOSE, we have
\( \mathcal{X}_1 = \{1, 2, 3, 4, 9, 10, 11, 12\}, \)
\( \mathcal{Y}_1 = \{(1, 2), (1, 3), (3, 2)\}. \)

For the second fault: STUCK-OPEN, we have
\( \mathcal{X}_2 = \{5, 6, 7, 8, 13, 14, 15, 16\}, \)
\( \mathcal{Y}_2 = \{(2, 1), (3, 1), (2, 3)\}. \)

Moreover, we can obtain \( P \) and \( P_f \) from \( \hat{M} \) and \( \hat{M}_f \), respectively.

\[ P = \begin{bmatrix}
0 & 0 & 4 & 0 & 0 & 0 & 8 & 0 & 0 & 0 \\
0 & 11 & 0 & 0 & 0 & 15 & 0 & 0 & 0 & 11 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & (1) & 0 & 0 \\
(6) & 0 & 0 & 0 & 6 & 0 & 0 & 0 & (4) & 0 \\
4 & 0 & 0 & 0 & 0 & 8 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 15 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 9 & 0 & 0 & 0 & 0 & (9) & 0 & 0 & 0 \\
0 & 0 & 14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}^T \]

\[ P_f = \begin{bmatrix}
0 & 0 & 4 & 0 & 0 & 0 & 8 & 0 & 0 & 0 \\
0 & 11 & 0 & 0 & 0 & 15 & 0 & 0 & 0 & 11 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & (5) & 0 & 0 \\
(2) & 0 & 0 & 0 & 6 & 0 & 0 & 0 & (10) & 0 \\
\end{bmatrix} \]

Comparing \( P \) with \( P_f \), we have \( \mathcal{R} = \{(1, 4), (8, 3), (9, 4), (16, 3)\} \).

Step 5. Consider a 2-step delay.
For \( 1 \in \mathcal{X}_1 \cap (1, 4) \in \mathcal{R} \), we have
\( Q_f(1, 4, 2) = 2, \)
i.e. \( (Q_f(1, 14), Q_f(2, 1, 14)) \in \mathcal{Y}_1. \)

For \( 9 \in \mathcal{X}_1 \cap (9, 4) \in \mathcal{R} \), we have
\( Q_f(9, 4, 1) = 1, \)
\( Q_f(9, 4, 2) = 2, \)
i.e. \( (Q_f(1, 9, 4), Q_f(1, 9, 4)) \in \mathcal{Y}_1. \)

Therefore, the first fault ‘STUCK − CLOSE’ can be diagnosed with the mostly 1-step delay.
For \( 8 \in \mathcal{X}_2 \cap (8, 3) \in \mathcal{R} \), we have
\( Q_f(8, 3, 1) = 3; \)
\( Q_f(8, 3, 2) = 3; \)
\( Q_f(8, 3, 3) = 3; \)
\( Q_f(8, 3, 4) = 3; \)
\( Q_f(8, 4, 3) = 3; \)
\( Q_f(8, 4, 3, 4) = 3; \)
\( Q_f(8, 4, 3, 4, 2) = 1. \)

For \( 16 \in \mathcal{X}_2 \cap (16, 3) \in \mathcal{R} \), we have
\( Q_f(16, 3, 1) = 1, \)
\( Q_f(16, 3, 2) = 1, \)
\( Q_f(16, 3, 3) = 1, \)
i.e. \( (Q_f(16, 3), Q_f(16, 3), Q_f(16, 3)) \in \mathcal{Y}_2. \)

Therefore, the second fault ‘STUCK − OPEN’ cannot be diagnosed with a 2-step delay.
In conclusion, the system cannot be diagnosed with a 2-step delay.

**Figure 7.** The normative and the faulty construction of the valve.
6. Conclusion

In this paper, we have studied the problem of diagnosability for composite Moore automata based on semi-tensor product of matrices. Different from existing methods, we have transformed the faults into changes of state transition and established an overall algebraic structure of the given system, based on which the matrix-based definitions of diagnosability for a fault and the system have been proposed. Following that we have constructed matrix observers and presented corresponding theorems and an algorithm to determine whether a fault (respectively, the system) is diagnosable. In conclusion, we have proved that the matrix approach to analysis of diagnosability for DESs is feasible and novel.

It is meaningful to find out another novel approach to the property analysis of DESs and the matrix observer established in this paper can be used to other properties verification. Besides, there is a ponderable task to solve in the future, that is, the computational complexity of proposed methods in this paper is not lower than several existing results.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work is supported by National Natural Science Foundation of China under Grants 61973175, 61573199 and U1804150.

ORCID

Zengqiang Chen http://orcid.org/0000-0002-1415-4073

References

Cassandras, C. G., & Lafortune, S. (2009). Introduction to discrete event systems. Springer Science & Business Media.

Chen, Z., Zhou, Y., Zhang, Z., & Liu, Z. (2020). Semi-tensor product of matrices approach to the problem of fault detection for discrete event systems(DESs). IEEE Transactions on Circuits and Systems II: Express Briefs, 67(12), 3098–3102. https://doi.org/10.1109/TCSCI.8920

Cheng, D., He, F., Qi, H., & Xu, T. (2015). Modeling, analysis and control of networked evolutionary games. IEEE Transactions on Automatic Control, 60(9), 2402–2415. https://doi.org/10.1109/TAC.2015.2404471

Cheng, D., & Qi, H. (2010). A linear representation of dynamics of Boolean networks. IEEE Transactions on Automatic Control, 55(10), 2251–2258. https://doi.org/10.1109/TAC.2010.2043294

Cheng, D., Qi, H., & Zhao, Y. (2012). An introduction to semi-tensor product of matrices and its applications. World Scientific.

Contant, O., Lafortune, S., & Tenekezis, D. (2004). Diagnosis of intermittent faults. Discrete Event Dynamic Systems, 14(2), 171–202. https://doi.org/10.1023/B:DISC.0000018570.20941.d2

Fornasini, E., & Valcher, M. E. (2015). Fault detection analysis of Boolean control networks. IEEE Transactions on Automatic Control, 60(10), 2734–2739. https://doi.org/10.1109/TAC.2015.2396646

Genc, S., & Lafortune, S. (2007). Distributed diagnosis of place-bounded Petri nets. IEEE Transactions on Automation Science and Engineering, 4(2), 206–219. https://doi.org/10.1109/TASE.2006.879916

Han, X., & Chen, Z. (2018). A matrix-based approach to verifying stability and synthesizing optimal stabilizing controllers for finite-state automata. Journal of the Franklin Institute, 355(17), 8642–8663. https://doi.org/10.1016/j.jfranklin.2018.09.009

Han, X., Chen, Z., Liu, Z., & Zhang, Q. (2017). Calculation of siphons and minimal siphons in Petri nets based on semi-tensor product of matrices. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(3), 531–536. https://doi.org/10.1109/TSMC.2015.2507162

Han, X., Chen, Z., & Zhao, J. (2017). Matrix approach to detectability of discrete event systems under partial observation. In 2017 13th IEEE conference on automation science and engineering (CASE) (pp. 187–192). IEEE. https://doi.org/10.1109/COASE.2017.8256101

HashtrudiZad, S., Kwong, R. H., & Wonham, W. M. (2003). Fault diagnosis in discrete-event systems: Framework and model reduction. IEEE Transactions on Automatic Control, 48(7), 1199–1212. https://doi.org/10.1109/TAC.2003.814099

He, J., Cao, J., & Rutkowski, L. (2020). Event-triggered control for output regulation of probabilistic logical systems with delays. IEEE Transactions on Systems, Man, and Cybernetics: Systems. Advance online publication. https://doi.org/10.1109/TSMC.2020.2964498

Li, H., Wang, Y., & Liu, Z. (2014). Stability analysis for switched Boolean networks under arbitrary switching signals. IEEE Transactions on Automatic Control, 59(7), 1978–1982. https://doi.org/10.1109/TAC.2014.2298731

Li, H., Zhao, G., Meng, M., & Feng, J. (2018). A survey on applications of semi-tensor product method in engineering. Science China Information Sciences, 61(1), 010202:1–010202:17. https://doi.org/10.1007/s11432-017-9238-1

Lin, F. (1994). Diagnosability of discrete event systems and its applications. Discrete Event Dynamic Systems, 4(2), 197–212. https://doi.org/10.1007/BF01441211

Lu, J., Li, H., Liu, Y., & Li, F. (2017). Survey on semi-tensor product method with its applications in logical networks and other finite-valued systems. IET Control Theory & Applications, 11(13), 2040–2047. https://doi.org/10.1049/iet-cta.2016.1659

Lu, J., Yang, J., Lou, J., & Qiu, J. (2019). Event-triggered sampled feedback synchronization in an array of output-coupled Boolean control networks. IEEE Transactions on Cybernetics. Advance online publication. https://doi.org/10.1109/TCYB.2019.2939761

Moreira, M. V., Jesus, T. C., & Basilio, J. C. (2011). Polynomial time verification of decentralized diagnosability of discrete event systems. IEEE Transactions on Automatic Control, 56(7), 1679–1684. https://doi.org/10.1109/TAC.2011.2124950

Reshmila, S., & Rajagopalan, D. (2019). Diagnosability of a class of discrete event systems based on observations. Control Theory and Technology, 17(3), 265–275. https://doi.org/10.1007/s11768-019-7298-3

Roszkowska, E., Dulewicz, P., & Janiec, L. (2018). Coordination of concurrent mobile robot motion processes with composite
Yin, X., & Li, Z. (2016a). Decentralized fault prognosis of discrete-event systems with guaranteed performance bound. *Automatica, 69*, 375–379. https://doi.org/10.1016/j.automatica.2016.03.015

Yin, X., & Li, Z. (2016b). Reliable decentralized fault prognosis of discrete-event systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 46*(11), 1598–1603. https://doi.org/10.1109/TSMC.2015.2499178

Yin, X., & Li, Z. (2018). Decentralized fault prognosis of discrete-event systems using state-estimation-based protocols. *IEEE Transactions on Cybernetics, 49*(4), 1302–1313. https://doi.org/10.1109/TCYB.6221036

Zaytoon, J., & Lafortune, S. (2013). Overview of fault diagnosis methods for discrete event systems. *Annual Reviews in Control, 37*(2), 308–320. https://doi.org/10.1016/j.arcontrol.2013.09.009

Zhang, Z., Chen, Z., Han, X., & Liu, Z. (2019). On the output feedback stabilisation of discrete event dynamic systems based upon the approach of semi-tensor product of matrices. *International Journal of Systems Science, 50*(8), 1595–1608. https://doi.org/10.1080/00207721.2019.1618941

Zhang, Z., Chen, Z., & Liu, Z. (2020). Reachability and controllability analysis of probabilistic finite automata via a novel matrix method. *Asian Journal of Control. Advance online publication.*

Zhang, Z., Xia, C., & Chen, Z. (2020). On the stabilization of nondeterministic finite automata via static output feedback. *Applied Mathematics and Computation, 365*, 124687:1–124687:11. https://doi.org/10.1016/j.amc.2019.124687

Zhang, Z., Xia, C., Chen, S., Yang, T., & Chen, Z. (2020). Reachability analysis of networked finite state machine with communication losses: A switched perspective. *IEEE Journal on Selected Areas in Communications, 38*(5), 845–853. https://doi.org/10.1109/JSAC.49

Zhao, J., Chen, Z., & Liu, Z. (2019). A novel matrix approach for the stability and stabilization analysis of colored Petri nets. *Science China Information Sciences, 62*(9), 192202:1–192202:14. https://doi.org/10.1007/s11432-018-9562-y

Zhu, G., Li, Z., Wu, N., & Al-Ahmari, A. (2017). Fault identification of discrete event systems modeled by Petri nets with unobservable transitions. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49*(2), 333–345. https://doi.org/10.1109/TSMC.2017.2762823

Zhu, S., Liu, Y., Lou, J., Lu, J., & Alsaaedi, F. E. (2020). Sampled-data state feedback control for the set stabilization of Boolean control networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 50*(4), 1580–1589. https://doi.org/10.1109/TSMC.6221021