Silent closure based pair verifier for fault pattern diagnosis of discrete event systems

Ye Liang, Dimitri Lefebvre, Zhiwu Li

Received: 7 December 2022 | Revised: 21 October 2023 | Accepted: 11 November 2023

IET Control Theory & Applications

DOI: 10.1049/cth2.12593

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Silent closure based pair verifier for fault pattern diagnosis of discrete event systems

Ye Liang[1] | **Dimitri Lefebvre**[2] | **Zhiwu Li**[3]

[1]School of Electro-Mechanical Engineering, Xidian University, Xi'an, China

[2]GREAH Laboratory, Normandy University, Le Havre, France

[3]Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau SAR, China

Correspondence
Ye Liang, School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China.
Email: liangye@stu.xidian.edu.cn

Funding information
National Key R&D Program of China, Grant/Award Number: 2018YFB1700104

Abstract

This paper addresses fault pattern diagnosis of discrete event systems, involving fault pattern detection and diagnosability. A fault pattern is modelled as a finite automaton whose accepted language is the objective to be diagnosed, representing the occurrence of complex or composite faults. A verifier for fault pattern detection based on the synchronous product of a system and a fault pattern is proposed. By removing all silent events, a silent closure is calculated based on the synchronous product, which offers computational advantages for systems that have a large number of silent events. An NSC/FSC pair verifier is then computed by taking the product of a normal silent closure and an accepted silent closure. By studying indeterminate cycles of the NSC/FSC pair verifier, necessary and sufficient verification conditions are established, asserting that a system is diagnosable with respect to a fault pattern if and only if there is no indeterminate cycle in the NSC/FSC pair verifier. It is shown that the proposed method requires polynomial time at most. Finally, a case study to illustrate the results is provided.

1 | INTRODUCTION

Fault diagnosis in a discrete event system (DES) is a crucial and challenging task to ensure its reliability and safety [1], which generally involves two objectives: fault detection and fault diagnosability. The former aims at detecting faults from given observations and the latter focuses on deciding whether faults can be distinguished within a finite delay after their occurrences [2]. Experience with monitoring of dynamic systems shows that there is a large spectrum of faulty situations in practical systems [3], such as multiple faults, intermittent faults [4], and temporary faults [5] that are not consistent with a single event fault. A broader approach is required for such cases and fault pattern diagnosis, introduced in [6], provides a general framework to solve the diagnosis problems by capturing the occurrences of particular strings in a system. The problem of fault pattern diagnosis has received extensive attention [6–15].

In [14, 16], the authors provide methods based on normal behaviours that can also be used for the diagnosability verification of fault patterns. This idea allows the calculation of faulty behaviours obtained by subtracting the normal language from the whole language, or by defining the faulty language as the union of all the behaviours that do not belong to the normal language. In addition, the results in [14] and [16] can also be used to compute the detection delay. The method in [15] proposes a method that involves modelling normal behaviour and calculating faulty behaviours by the subtraction operation. The works in [17–22] work on the problem of diagnosis in continuous time systems.

Another class of methods is based on a systemic construction, namely a diagnoser [2], that is suitable to detect faults on-line and to verify off-line the necessary and sufficient conditions for diagnosability. Basically, such a diagnoser results from the determinisation, i.e. the observer, of a verification structure obtained from the system and pattern. In [23], the authors distinguish two types of pattern diagnosability, S-type and T-type, based on the diagnoser properties. Compared with [23], this work focuses on S-type patterns (but T-type patterns can be viewed as a particular subclass of S-type patterns), and the main difference is that the proposed method in this work has polynomial complexity, which is a significant improvement compared with the diagnoser structure with potentially exponential complexity presented in [23]. In [24], the authors address the problem of fault diagnosis in decentralized DESs by extending the diagnoser into local diagnosers with a coordinator. However, the state space of the corresponding diagnosers

are all in the worst case exponential with respect to the size of the system.

To overcome the potential state explosion problem, a pair verifier technique [25, 26] is introduced to offer a worst case polynomial test with respect to the number of system states for diagnosability. Several extensions have been developed [13, 27–36]. The pair verifier structures are mainly obtained from the self composition of the verifiers. In particular, an algorithm with linear complexity proposed in [31] converts the fault detection problem into state isolation to determine whether the observations allow us to isolate the states to be within a particular set of states that indicates the occurrence of failures. Especially, the authors in [32] propose two types of pair verifiers in a decentralized diagnosis framework with respect to different local decisions, and the authors in [33] address the modular diagnosability problem by computing a pair verifier. The authors in [34] construct a pair verifier to perform codiagnosability analysis. With structures that result from the composition of faulty and normal verifiers [37], additional gains in space complexity can be obtained.

Finally, model checking based approaches are also used for diagnosis purpose. The study in [9] touches upon the diagnosability analysis of DESs transforming the problem of pattern diagnosis into a model checking problem. In [7, 8], the authors introduce the methods of constructing local pattern diagnosers by using subsystems, which extends the pattern diagnosis problem from a monolithic model to a distributed framework. In [13], the authors review the main definitions of diagnosability with regard to intermittent faults, and discuss appropriate verification techniques. In [10], linear-time temporal logic formulas are used to specify failures in a system such that the problem of testing diagnosability is reduced to that of model checking.

This paper also focuses on fault pattern diagnosis of DESs, fault pattern detection, and diagnosability verification. Labelled finite automata are used as models of the system and pattern to be diagnosed. Improving the numerical complexity of the diagnosis approach is the main challenge of this contribution. In the perspective of the aforementioned contributions, we aim to propose verifiers of reduced size, in particular for systems including numerous silent events. Based on the synchronous product of a system and a fault pattern, we propose a silent closure by removing all silent events. Then, an NSC/FSC pair verifier (NSC/FSC PV) is constructed by taking the product of a normal silent closure and fault silent closure, both of which are obtained from the silent closure with respect to normal and faulty behaviors, respectively. The successive steps of our analysis are visualized in Figure 1.

The rest of the paper is organized as follows. Section 2 reviews finite state automata. Section 3 begins with the notions of fault patterns and then touches upon fault pattern diagnosis in DESs, including fault pattern detection and diagnosability. Sections 4. A and B provide the structures for fault pattern diagnosability based on a standard PV and an NSC/FSC PV respectively. Section 4.3 compares the space complexity of different approaches. Section 5 provides a case study to illustrate the results. Section 6 concludes this research.

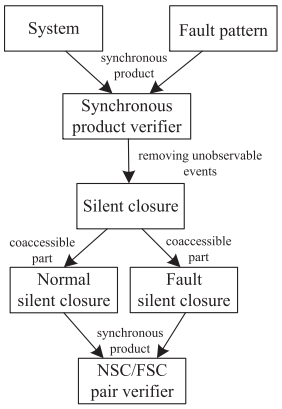

FIGURE 1 Schematic of the methodology.

2 | PRELIMINARIES

We use $\mathbb{N}$ and $\mathbb{N}_0$ to denote the sets of strictly positive integers and non-negative integers, respectively. Write $\mathbb{N}_k = \{1, 2, \ldots, k\}$.

Definition 1. A deterministic finite automaton (DFA) is a four-tuple $G_0 = (L, \Sigma, \delta, l_0)$, where L is the set of states, Σ is the set of events, l_0 is the initial state, and $\delta : L \times \Sigma \to L$ is the partial transition function: $l' = \delta(l, \sigma)$ means that there is a transition labelled with event $\sigma \in \Sigma$ from the state l to state l'. Let Σ^* be the set of all finite strings defined over Σ, including the empty string λ. Transition function δ can be extended to $L \times \Sigma^* \to L$ in an usual way: given $l \in L$, $w \in \Sigma^*$, and $\sigma \in \Sigma$, $\delta(l, \lambda) = l$ and $\delta(l, w\sigma) = \delta((\delta(l, w)), \sigma)$.

The set Σ can be partitioned into two disjoint subsets $\Sigma = \Sigma_o \cup \Sigma_{uo}$, where Σ_o and Σ_{uo} represent the sets of observable and unobservable events, respectively. The concatenation of two strings $w', w'' \in \Sigma^*$ is the string $w = w'w'' \in \Sigma^*$. To be more general, we introduce the notion of output labels. Let $E_\varepsilon = E \cup \{\varepsilon\}$ be the set of output labels, where E is the set of observable labels and ε is the empty label. A labelled finite automaton (LFA) can be defined as follows.

Definition 2. An LFA is a three-tuple $G = (G_0, E_\varepsilon, Lab)$, where $G_0 = (L, \Sigma, \delta, l_0)$ is a DFA, E_ε is the set of output labels, and $Lab : \Sigma \to E_\varepsilon$ is the labelling function, where $Lab(\sigma) \in E$ if $\sigma \in \Sigma_o$, and $Lab(\sigma) = \varepsilon$ if $\sigma \in \Sigma_{uo}$.

In an LFA, there exist transitions modelling the occurrences of silent events (i.e. ε-transitions), which cannot be observed. Moreover, two or more transitions outgoing from a given state could possibly produce the same label. In this case, one detects that an event has occurred but cannot determine exactly which transition has fired. The labelling function, or mask function, defines the observation generated by the occurrence of each event [14].

Given a state $l \in L$ in an LFA, the set of active events at l is defined as $\Lambda(l) = \{\sigma \in \Sigma | \exists l' \in L : l' = \delta(l, \sigma)\}$. Given a string $w \in \Sigma^*$, its length is defined as the number of events in w, denoted by $|w|$. A string $w' \in \Sigma^*$ is said to be a prefix

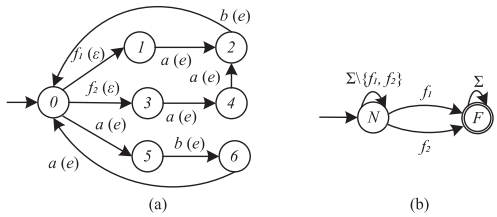

FIGURE 2 (a) Labelled finite automaton (LFA) G_1 and (b) fault pattern Ω_1.

of $w \in \Sigma^*$ if there exists $w'' \in \Sigma^*$ such that $w'w'' = w$, denoted by $w' \leq w$. The language generated by the LFA G is defined as $\mathcal{L}(G) = \{s \in \Sigma^* | \delta(l_0, s)!\}$, where $\delta(l_0, s)!$ means that "$\delta(l_0, s)$ is defined". Given string $w \in \mathcal{L}(G)$, $\mathcal{L}(G)/w$ denotes the post-language of $\mathcal{L}(G)$ after w : $\mathcal{L}(G)/w = \{w' \in \Sigma^* \mid ww' \in \mathcal{L}(G)\}$. A run that begins with the initial state l_0 has the form ρ: $l_0 \xrightarrow{\sigma_0} l_1 \xrightarrow{\sigma_1} \ldots l_n \xrightarrow{\sigma_n} l_{n+1}$, where $l_i, l_{i+1} \in L, \sigma_i \in \Sigma$, and $l_{i+1} = \delta(l_i, \sigma_i)$ for $i \in \{0, 1, \ldots, n\}$. In this case, we say that the run ρ is associated with the string $w = \sigma_0 \ldots \sigma_n \in \Sigma^*$, and use $w(\rho)$ to denote the string associated to ρ. In a similar way, given a string w, we use $\rho(w)$ to denote the run generated by w from the initial state l_0. A run is said to be a cycle if $l_{n+1} = l_0$.

Given an LFA G, a projection function $\mathcal{P} : \Sigma^* \to E^*$ is defined as follows. For $w \in \Sigma^*$ and $\sigma \in \Sigma$,

$$\mathcal{P}(w\sigma) = \begin{cases} \mathcal{P}(w)e & \text{if } Lab(\sigma) = e \in E \\ \mathcal{P}(w) & \text{if } Lab(\sigma) = \varepsilon, \end{cases}$$

and for the empty string λ, $\mathcal{P}(\lambda) = \lambda$.

We use $\mathcal{L}_E(G)$ to denote the observed language of G, defined by $\mathcal{L}_E(G) = \{\mathcal{P}(w) \in E^* | w \in \mathcal{L}(G)\}$. Given an observation w_e, the inverse projection $\mathcal{P}^{-1} : E^* \to \Sigma^*$ is defined by $\mathcal{P}^{-1}(w_e) = \{w \in \mathcal{L}(G) | \mathcal{P}(w) = w_e\}$. Given an LFA G, we use $R_G(w_e)$ to denote the set of states resulting from the execution of an event sequence $w \in \mathcal{P}^{-1}(w_e)$ from state $l \in L$, defined by $R_G(w_e) = \{l \in L | \exists w \in P^{-1}(w_e) : l = \delta(l_0, w)\}$.

Example 1. Consider an LFA G_1 shown in Figure 2(a), where $L = \{0, 1, 2, 3, 4, 5, 6\}$, with 0 being the initial state, $\Sigma = \{a, b, f_1, f_2\}$, $\Sigma_o = \{a, b\}$, $\Sigma_{uo} = \{f_1, f_2\}$, and $E = \{e\}$. The labelling function Lab is defined as $Lab(a) = Lab(b) = e$, and $Lab(f_1) = Lab(f_2) = \varepsilon$. A possible run generated by system G_1 from the initial state is ρ: $0 \xrightarrow{f_1} 1 \xrightarrow{a} 2 \xrightarrow{b} 0$, where the associated string of ρ is $w = f_1 ab$. The projection of w with respect to the set of observable labels is $\mathcal{P}(w) = ee$.

3 | FAULT PATTERN DIAGNOSIS OF AUTOMATA

A fault pattern, simply called a pattern in this research, is defined as a finite state automaton whose accepted language is the objective to be diagnosed, which represents the occurrence of complex or composite faults.

Definition 3. A (fault) pattern of an LFA $G = (G_0, E_\varepsilon, Lab)$ with $G_0 = (L, \Sigma, \delta, l_0)$ is a DFA $\Omega = (S, \Sigma, \delta_\Omega, s_0, s_\Omega)$, where S is the set of states, Σ is the set of events, $s_0 \in S$ is the initial state, $s_\Omega \in S$ is the single final, i.e. accepted state, and $\delta_\Omega : S \times \Sigma \to S$ is the transition function. The fault pattern Ω satisfies a complete condition, i.e. for all $s \in S$, $\Lambda(s) = \Sigma$ and the final state s_Ω is stable, i.e. for all $\sigma \in \Sigma$, $\delta_\Omega(s_\Omega, \sigma) = s_\Omega$.

The language of fault pattern Ω, denoted by $\mathcal{L}(\Omega)$, satisfies $\mathcal{L}(\Omega) = \Sigma^*$ due to its complete condition. We use $\mathcal{L}_\mathcal{A}(\Omega)$ to denote the accepted language of Ω, defined as $\mathcal{L}_\mathcal{A}(\Omega) = \{w \in \mathcal{L}(\Omega) | \delta_\Omega(s_0, w) = s_\Omega\}$, and define the target language of LFA G as $\mathcal{L}_\mathcal{A}(G) = \mathcal{L}(G) \cap \mathcal{L}_\mathcal{A}(\Omega)$.

Example 2. Consider an example of a fault pattern Ω_1, shown in Figure 2(b), where the set of states is $S = \{N, F\}$, the set of the events is Σ, the final state is F, and the initial state is N. The fault pattern defines the occurrence of f_1 or f_2; its accepted language is $\mathcal{L}_\mathcal{A}(\Omega_1) = \Sigma^* f_1 \Sigma^* \cup \Sigma^* f_2 \Sigma^*$.

The definition of the detection function is given as follows.

Definition 4. A detection function $Detect_\Omega : \mathcal{L}_E(G) \to \{Yes, No, Ambiguous\}$ is defined, for any $w_e \in \mathcal{L}_E(G)$, as

- $Detect_\Omega(w_e) = Yes$ if $\mathcal{P}^{-1}(w_e) \subseteq \mathcal{L}_\mathcal{A}(G)$,
- $Detect_\Omega(w_e) = No$ if $\mathcal{P}^{-1}(w_e) \cap \mathcal{L}_\mathcal{A}(G) = \emptyset$,
- $Detect_\Omega(w_e) = Ambiguous$, otherwise.

Given an observed sequence w_e, the output of the detection function in Definition 4 is *Yes* if all the strings generating w_e are contained in $\mathcal{L}_\mathcal{A}(G)$, i.e. the fault pattern has certainly occurred. The output is *No* if no string w whose projection is w_e is in $\mathcal{L}_\mathcal{A}(G)$, indicating that the fault pattern has not occurred. Otherwise, the output is Ambiguous, i.e. it is uncertain whether the fault pattern has occurred or not.

Definition 5. Given an LFA G and a pattern Ω, G is diagnosable with respect to pattern Ω if

$(\exists k \in \mathbb{N})\ (\forall w \in \mathcal{L}_\mathcal{A}(G))(\forall w' \in \mathcal{L}(G)/w)\ (|w'| \geq k) \Rightarrow [\mathcal{P}^{-1}(\mathcal{P}(ww')) \subseteq \mathcal{L}_\mathcal{A}(G)]$.

Note that in Definition 5, we are interested in the first occurrence of the pattern and diagnosability analysis captures only the strings accepted by the pattern. To verify fault pattern diagnosability, we make the following assumption:

(H) Given an LFA G, its observed language $\mathcal{L}_E(G)$ is live, i.e. for all observations $w_e \in \mathcal{L}_E(G)$, there always exists a label $e \in E$ such that $w_e e \in \mathcal{L}_E(G)$.

3.1 | Fault pattern detection based on synchronous product

Inspired by the notion of synchronous product of two automata, this section introduces a synchronous product verifier. Note that a state isolation based verifier can also be

obtained for fault pattern detection [31, 38], whose structure is similar to the synchronous product verifier. For the sake of simplicity, the details are not pursued here.

Definition 6. Given an LFA $G = (G_0, E_\varepsilon, Lab)$ with $G_0 = (L, \Sigma, \delta, l_0)$, and a pattern $\Omega = (S, \Sigma, \delta_\Omega, s_0, s_\Omega)$, a synchronous product verifier G_Ω of G with respect to Ω is an LFA $G_\Omega = (L_{G_\Omega}, \Sigma, \delta_{G_\Omega}, l_0^{G_\Omega}, L_F^{G_\Omega}, E_\varepsilon, Lab)$, where $L_{G_\Omega} \subseteq L \times S$ is the set of states, Σ is the set of events, $l_0^{G_\Omega} = (l_0, s_0)$ is the initial state, $L_F^{G_\Omega} = L \times \{s_\Omega\}$ is the set of final states, and $\delta_{G_\Omega} : (L \times S) \times \Sigma \to (L \times S)$ is the transition function defined for $\sigma \in \Sigma$, $s, s' \in S$, $l, l' \in L$, by $\delta_{G_\Omega}((l, s), \sigma) = (l', s')$, if $\delta(l, \sigma) = l'$ and $\delta_\Omega(s, \sigma) = s'$.

Since Ω satisfies the complete condition, $\mathcal{L}_\mathcal{A}(G_\Omega) = \mathcal{L}_\mathcal{A}(G)$ holds thanks to the definition of accepted language $\mathcal{L}_\mathcal{A}(G_\Omega)$ of G_Ω. Let $\mathcal{L}(G_\Omega)$ be the generated language of G_Ω. Given a state l_{G_Ω} and an observed sequence w_e, we define $R_{G_\Omega}(w_e) = \{l_{G_\Omega} \in L_{G_\Omega} | \exists w \in P^{-1}(w_e) : l_{G_\Omega} = \delta_{G_\Omega}(l_0^{G_\Omega}, w)\}$, where the transition function δ_{G_Ω} is extended from Σ to Σ^* in the usual way.

Proposition 1. *Given an LFA G, a pattern Ω, the synchronous product verifier G_Ω, and an observation $w_e \in \mathcal{L}_E(G)$, the detection function $Detect_\Omega$ satisfies*

(a) $Detect_\Omega(w_e) =$ Yes *if and only if* $R_{G_\Omega}(w_e) \subseteq L_F^{G_\Omega}$,
(b) $Detect_\Omega(w_e) =$ No *if and only if* $R_{G_\Omega}(w_e) \cap L_F^{G_\Omega} = \emptyset$,
(c) $Detect_\Omega(w_e) =$ Ambiguous, *otherwise.*

Proof. Given an observed sequence $w_e \in \mathcal{L}_E(G)$ and an event sequence $w = \sigma_0 \dots \sigma_n \in \mathcal{P}^{-1}(w_e)$, the runs in Ω and G_Ω that begin respectively from the initial states s_0 and $l_0^{G_\Omega}$, associated with w, are ρ_Ω: $s(0) \xrightarrow{\sigma_0} s(1) \xrightarrow{\sigma_1} \cdots s(n) \xrightarrow{\sigma_n} s(n+1)$ and ρ_{G_Ω}: $l_{G_\Omega}(0) \xrightarrow{\sigma_0} l_{G_\Omega}(1) \xrightarrow{\sigma_1} \cdots l_{G_\Omega}(n) \xrightarrow{\sigma_n} l_{G_\Omega}(n+1)$.

To prove (only if) of case (a), assume that $Detect_\Omega(w_e) = Yes$. Then we have $w \in \mathcal{L}_\mathcal{A}(G)$ and $\delta_\Omega(s(0), w) = s_\Omega$. By considering the run ρ_Ω, there exists one or more indices n_1, ..., n_k, such that $s(0) = \cdots = s(n_1) = s_0$, $s(n_h + 1) = \cdots = s(n_{h+1})$, $h \in \mathbb{N}_{k-1}$, and $s(n_k + 1) = \cdots = s(n+1) = s_\Omega$. Considering now the run ρ_{G_Ω} and according to Definition 6, we have $l_{G_\Omega}(0), \dots, l_{G_\Omega}(n_1), l_{G_\Omega}(n_h + 1), \dots, l_{G_\Omega}(n_{h+1}) \in L_{G_\Omega} \setminus L_F^{G_\Omega}$, $h \in \mathbb{N}_{k-1}$, and $l_{G_\Omega}(n_k + 1), \dots, l_{G_\Omega}(n+1) \in L_F^{G_\Omega}$. In particular, the last state of the run ρ_{G_Ω} belongs to $L_F^{G_\Omega}$. Thus $R_{G_\Omega}(w_e) \subseteq L_F^{G_\Omega}$.

To prove (if) of case (a), assume that $R_{G_\Omega}(w_e) \subseteq L_F^{G_\Omega}$. For any w such that $\mathcal{P}(w) = w_e$ there exist one or more indices $n_1, \dots, n_k$ and $k-1$ states $s(n_{h+1}) \in S$, $h \in \mathbb{N}_{k-1}$, such that $l_{G_\Omega}(0), \dots, l_{G_\Omega}(n_1), l_{G_\Omega}(n_h + 1), \dots, l_{G_\Omega}(n_{h+1}) \in L_{G_\Omega} \setminus L_F^{G_\Omega}$, $h \in \mathbb{N}_{k-1}$, and $l_{G_\Omega}(n_k + 1), \dots, l_{G_\Omega}(n+1) \in L_F^{G_\Omega}$. By Definition 6, $s(0) = \cdots = s(n_1) = s_0$, $s(n_h + 1) = \cdots = s(n_{h+1})$ with $h \in \mathbb{N}_{k-1}$ and $s(n_k + 1) = \cdots = s(n+1) = s_\Omega$. Hence, the run ρ_Ω ends in s_Ω and $w = \sigma_0 \sigma_1 \dots \sigma_n \in \mathcal{L}_\mathcal{A}(G)$.

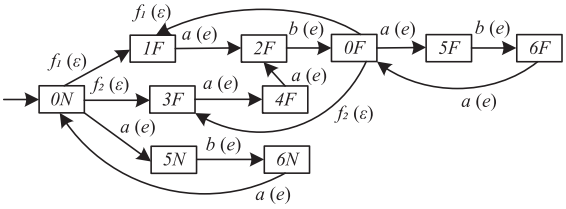

FIGURE 3 Synchronous product verifier $G_{1\Omega_1}$.

By Definition 4, we have $Detect_\Omega(w_e) = Yes$. The proofs of cases (b) and (c) are similar to (a). □

Example 3. Consider the fault pattern Ω_1 in Figure 2(b) and an LFA G_1 in Figure 2(a). The synchronous product verifier $G_{1\Omega_1}$ of G_1 and Ω_1 is shown in Figure 3. Given, for example, an observation ee, by Proposition 1, the detection is *Ambiguous*, since $R_{G_{1\Omega_1}}(ee) = \{6N, 0F, 2F\}$, where $0F, 2F \in L_F^{G_\Omega}$ and $6N \notin L_F^{G_\Omega}$.

3.2 | Fault pattern diagnosability based on diagnoser

In this section, we provide a method for fault pattern diagnosability based on a diagnoser structure [2]. We define $R_{G_\Omega}(l_{G_\Omega}, w_e) = \{l'_{G_\Omega} \in L_{G_\Omega} | \exists w \in \Sigma^* : l'_{G_\Omega} = \delta_{G_\Omega}(l_{G_\Omega}, w), \mathcal{P}(w) = w_e\}$.

Definition 7. Given the synchronous product $G_\Omega = (L_{G_\Omega}, \Sigma, \delta_{G_\Omega}, l_0^{G_\Omega}, L_F^{G_\Omega}, E_\varepsilon, Lab)$ of system G and pattern Ω, the diagnoser of G with respect to Ω is defined as $d_G(\Omega) = (L_d, \delta_d, l_0^d, E, Lab)$, where $L_d \subseteq 2^{L_{G_\Omega}}$ is the set of states, $l_0^d = R_{G_\Omega}(\lambda)$ is the initial state, and $\delta_d : L_d \times E \to L_d$ is the transition function satisfying $l'_d = \delta_d(l_d, e)$ if there exist $l_d, l'_d \in L_d$, $e \in E$ such that $\bigcup_{l_{G_\Omega} \in l_d} R_{G_\Omega}(l_{G_\Omega}, e) = l'_d$.

Definition 7 shows that the diagnoser is obtained by the determinisation of the synchronous product verifier with respect to the set of observable labels E. Diagnosers can be used to check diagnosability by introducing the notions of indeterminate states and cycles.

Definition 8. Given a system G, a pattern Ω, and its diagnoser $d_G(\Omega)$, a state $l_d \in L_d$ is said to be indeterminate if $l_d \cap L_F^{G_\Omega} \neq \emptyset$ and $l_d \cap (L_{G_\Omega} \setminus L_F^{G_\Omega}) \neq \emptyset$. A cyclic run, for short a cycle, formed by states l_1^d, l_2^d, ..., $l_n^d \in L_d$, is said to be an indeterminate cycle if

1) for all $i \in \mathbb{N}_n$, the state l_i^d is indeterminate,
2) for all $i \in \mathbb{N}_n$, there exist at least two states $l_i^{G_\Omega}, l_i^{\widetilde{G_\Omega}} \in l_i^d$ with $l_i^{G_\Omega} = (l_i, s_i)$, $l_i^{\widetilde{G_\Omega}} = (\widetilde{l_i}, \widetilde{s_i})$, $l_i, \widetilde{l_i} \in L$, $s_i = s_\Omega$, and $\widetilde{s_i} \in S \setminus \{s_\Omega\}$ such that there exist at least two cycles in G_Ω composed of the states $l_1^{G_\Omega}, \dots, l_n^{G_\Omega}$, and $l_1^{\widetilde{G_\Omega}}, \dots, l_n^{\widetilde{G_\Omega}}$ with the

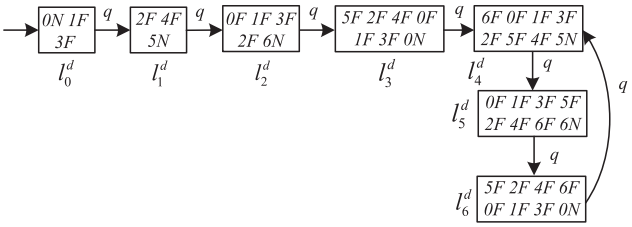

FIGURE 4 Diagnoser $d_{G_1}(\Omega_1)$.

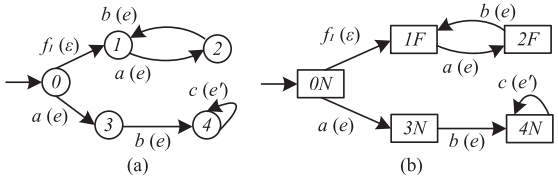

FIGURE 5 (a) LFA G_2 and (b) synchronous product verifier $G_{2\Omega_1}$.

forms of $\rho_{G_\Omega} : l_1^{G_\Omega} \to \cdots \to l_i^{G_\Omega} \to \cdots \to l_n^{G_\Omega}$, and $\widetilde{\rho_{G_\Omega}} : \widetilde{l_1^{G_\Omega}} \to \cdots \to \widetilde{l_i^{G_\Omega}} \to \cdots \to \widetilde{l_n^{G_\Omega}}$, where w and w' are two strings associated with the cycles ρ_{G_Ω} and $\widetilde{\rho_{G_\Omega}}$, respectively, with $\mathcal{P}(w) = \mathcal{P}(w')$.

A cycle in the diagnoser is an indeterminate cycle if it is composed of indeterminate states only, which corresponds to two cycles in the synchronous product verifier: one composed of non-final states only and the other composed of final states only, where both cycles have the same observation. Given an LFA G that satisfies Assumption H, G is diagnosable with respect to Ω iff there is no indeterminate cycle in the diagnoser $d_G(\Omega)$. The proof of this result is similar to [2] and is not presented for economy of space.

Example 4. Consider an LFA G_1 in Figure 2(a), the fault pattern Ω_1 in Figure2(b). The resulting diagnoser $d_{G_1}(\Omega_1)$ is computed in Figure 4. The system G_1 is not diagnosable with respect to pattern Ω_1 since there exists an indeterminate cycle $l_4^d l_5^d l_6^d$ in $d_{G_1}(\Omega_1)$.

4 | VERIFICATION OF FAULT PATTERN DIAGNOSABILITY

This section sums up the principle of a standard PV and provides an NSC/FSC PV structure to check fault pattern diagnosability. In particular, the NSC/FSC PV structure is constructed by removing all silent events of the system, taking advantage of systems with numerous silent events.

4.1 | Fault pattern diagnosability with standard PV

This section introduces a standard PV that can be used for fault pattern diagnosability, which is obtained by the self-product of G_Ω [26].

Definition 9. Given the synchronous product verifier G_Ω of G and Ω, a standard PV $P_G(\Omega)$ is defined as a nondeterministic automaton $P_G(\Omega) = (L_P, \delta_P, l_0^P, E_\varepsilon, Lab)$, where $L_P \subseteq L_{G_\Omega} \times L_{G_\Omega}$ is the set of states, $l_0^P = (l_0^{G_\Omega}, l_0^{G_\Omega})$ is the initial state, and δ_P is the transition relation defined as follows. Let $l_P = (l_{G_\Omega}^1, l_{G_\Omega}^2) \in L_P$, and $\sigma_1, \sigma_2 \in \Sigma$ such that $l_{G_\Omega}^{1'} = \delta_{G_\Omega}(l_{G_\Omega}^1, \sigma_1)$ and $l_{G_\Omega}^{2'} = \delta_{G_\Omega}(l_{G_\Omega}^2, \sigma_2)$. We have:

1. If $Lab(\sigma_1) = Lab(\sigma_2) = e$, $e \in E$, then $((l_{G_\Omega}^1, l_{G_\Omega}^2), e, (l_{G_\Omega}^{1'}, l_{G_\Omega}^{2'})) \in \delta_P$,
2. If $Lab(\sigma_1) = Lab(\sigma_2) = \varepsilon$, then

$$\begin{cases} \left(\left(l_{G_\Omega}^1, l_{G_\Omega}^2\right), \varepsilon, \left(l_{G_\Omega}^{1'}, l_{G_\Omega}^2\right)\right) \in \delta_P \\ \left(\left(l_{G_\Omega}^1, l_{G_\Omega}^2\right), \varepsilon, \left(l_{G_\Omega}^1, l_{G_\Omega}^{2'}\right)\right) \in \delta_P \\ \left(\left(l_{G_\Omega}^1, l_{G_\Omega}^2\right), \varepsilon, \left(l_{G_\Omega}^{1'}, l_{G_\Omega}^{2'}\right)\right) \in \delta_P. \end{cases}$$

By construction, the transition relation δ_P tracks two strings in $\mathcal{L}(G_\Omega)$ which generate the same output from an observational point of view, while updating the failure information as the two strings evolve. The standard PV can be used to check diagnosability by considering indeterminate states and cycles of this structure.

Definition 10. Given the synchronous product verifier G_Ω of a system G and a pattern Ω, and the standard PV $P_G(\Omega)$, a state $l_P = (l_{G_\Omega}^1, l_{G_\Omega}^2) \in L_P$ is said to be indeterminate if $(l_{G_\Omega}^1, l_{G_\Omega}^2) = ((l^1, s^1), (l^2, s^2))$, $l^1, l^2 \in L$, $s^1 \in S \setminus \{s_\Omega\}$, $s^2 = s_\Omega$, or vice verse. A cycle in $P_G(\Omega)$ is said to be *indeterminate* if all states of the cycle are indeterminate.

Given an LFA G that satisfies Assumption H, G is diagnosable with respect to pattern Ω iff there is no indeterminate cycle in $P_G(\Omega)$. The proof of this result is similar to [26] and is not presented for economy of space.

Example 5. Consider an LFA G_2 in Figure 5(a) with $\Sigma_{uo} = \{f_1\}$, $\Sigma_o = \{a, b, c\}$, and $E = \{e, e'\}$, where $Lab(a) = Lab(b) = e$, and $Lab(c) = e'$, and the fault pattern Ω_1 in Figure 2(b). The synchronous product $G_{2\Omega_1}$ and the standard PV $P_{G_2}(\Omega_1)$ are shown in Figures 5(b) and 6, respectively. The system G_2 is diagnosable with respect to Ω_1 since there do not exist indeterminate cycles in $P_{G_2}(\Omega_1)$ (see Definition 10).

On the contrary, consider the LFA G_1 in Figure 2(a) and the pattern Ω_1 in Figure 2(b) previously discussed. The standard PV $P_{G_1}(\Omega_1)$ can be obtained in a similar way. Since there exists at least one indeterminate cycle in $P_{G_1}(\Omega_1)$, G_1 is not diagnosable with respect to the pattern Ω_1. For the sake of simplicity, the details are not pursued here.

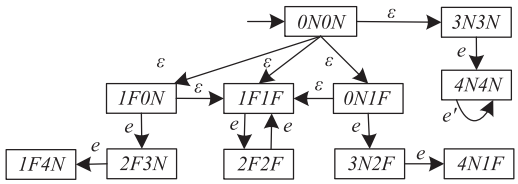

FIGURE 6 Standard PV $P_{G_2}(\Omega_1)$.

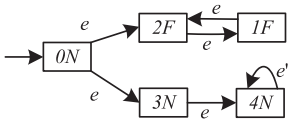

FIGURE 7 Silent closure G_{2S_1}.

4.2 | Fault pattern diagnosability with NSC/FSC PV

In contrast to the standard PV introduced above, the NSC/FSC PV is built on the basis of normal and fault silent closures, which are coaccessible parts of a silent closure respectively, especially offering an advantage for the systems with numerous silent events.

Definition 11. Given the synchronous product verifier $G_\Omega = (L_{G_\Omega}, \Sigma, \delta_{G_\Omega}, l_0^{G_\Omega}, L_F^{G_\Omega}, E_\varepsilon, Lab)$, the silent closure of G_Ω is a four-tuple $G_S = (L_S, E, \delta_S, l_0^S)$, where E is the set of observable labels, L_S is the set of states, defined as $L_S = \{l'_{G_\Omega} \in L_{G_\Omega} | \exists l_{G_\Omega} \in L_{G_\Omega}, \sigma \in \Sigma_o : l'_{G_\Omega} = \delta_{G_\Omega}(l_{G_\Omega}, \sigma)\}$, $l_0^S = l_0^{G_\Omega}$ is the initial state, and $\delta_S \subseteq L_S \times E \times L_S$ is the transition relation, defined as $(l_S, e, l'_S) \in \delta_S$ if and only if there exists a run $l_S \xrightarrow{\sigma_1} l^1_{G_\Omega} \ldots \xrightarrow{\sigma_n} l^n_{G_\Omega} \xrightarrow{\sigma} l'_S$, $n \geq 0$, $l_S, l'_S \in L_S$, in G_Ω such that for $i = 1, 2, \ldots, n$, $l^i_{G_\Omega} \in L_{G_\Omega}$, $Lab(\sigma_i) = \varepsilon$, and $Lab(\sigma) = e$.

The transition relation δ_S can be extended from E to E^* in the usual way.

Example 6. Consider an LFA G_2 in Figure 5(a), the pattern Ω_1 in Figure 2(b), and their synchronous product verifier $G_{2\Omega_1}$ in Figure 5(b). By Definition 11, the silent closure G_{2S_1} of $G_{2\Omega_1}$ is obtained in Figure 7.

Definition 12. Given a silent closure G_S, the normal silent closure (NSC) of G_S is a four-tuple $G_N = (L_N, E, \delta_N, l_0^N)$, where L_N is the set of the states defined as $L_N = \{l_S \in L_S | l_S \notin L_F^{G_\Omega}\}$, E is the set of observable labels, $\delta_N \subseteq L_N \times E \times L_N$ is the transition relation, defined by $(l_N, e, l'_N) \in \delta_N$ for $l_N, l'_N \in L_N$, if there exists $e \in E$ such that $(l_N, e, l'_N) \in \delta_S$, and $l_0^N = l_0^S$ is the initial state.

From Definition 12, one can conclude that for any state $l_N \in L_N$, it holds $l_N \notin L_F^{G_\Omega}$, i.e. the NSC G_N is the coaccessible part of G_S with respect to $L_{G_\Omega} \setminus L_F^{G_\Omega}$.

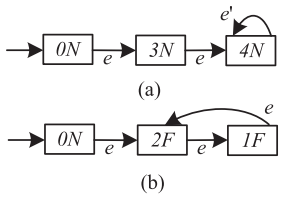

FIGURE 8 (a) Normal silent closure G_{2N_1} and (b) fault silent closure G_{2F_1}.

Definition 13. Given a silent closure G_S, the fault silent closure (FSC) of G_S is a four-tuple $G_F = (L_F, E, \delta_F, l_0^F)$, where L_F is the set of the states defined as $L_F = \{l_S \in L_S | \exists l'_S \in L_S, \exists w_e \in E^* : (l_S, w_e, l'_S) \in \delta_S, l'_S \in L_F^{G_\Omega}\}$, E is the set of observable labels, $\delta_F \subseteq L_F \times E \times L_F$ is the transition relation, defined by $(l_F, e, l'_F) \in \delta_F$ for $l_F, l'_F \in L_F$, if there exists $e \in E$ such that $(l_F, e, l'_F) \in \delta_S$, and $l_0^F = l_0^S$ is the initial state.

From Definition 13, the construction of the fault silent closure G_F is essentially the coaccessible part of G_S with respect to the states $l_S \in L_S$ which satisfy also $l_S \in L_F^{G_\Omega}$.

Example 7. Consider the silent closure G_{2S_1} of $G_{2\Omega_1}$ in Figure 7. By Definitions 12 and 13, the NSC G_{2N_1} and FSC G_{2F_1} are obtained in Figures 8(a) and 8(b), respectively.

The NSC/FSC PV is formally introduced as follows.

Definition 14. Given an NSC G_N and an FSC G_F, an NSC/FSC PV of G and Ω is a four-tuple $NFP_G(\Omega) = (L_{NFP}, E, \delta_{NFP}, l_0^{NFP})$, where $L_{NFP} \subseteq L_N \times L_F$ is the set of states, E is the set of observable labels, $\delta_{NFP} \subseteq L_{NFP} \times E \times L_{NFP}$ is the transition relation, defined by $(l_{NFP}\ e,\ l'_{NFP}) \in \delta_{NFP}$ for $l_{NFP} = (l_N, l_F)$, $l'_{NFP} = (l'_N, l'_F) \in L_{NFP}$, if there exist $l_N, l'_N \in L_N, l_F, l'_F \in L_F, e \in E$ such that $(l_N, e, l'_N) \in \delta_N$ and $(l_F, e, l'_F) \in \delta_F$, and $l_0^{NFP} = (l_0^N, l_0^F)$ is the initial state.

By construction, the transition relation δ_{NFP} tracks two strings of the NSC G_N and the FSC G_F, respectively, which generate the same output from an observational point of view, while updating the pattern information as the two strings evolve.

It follows from Definitions 12 and 13 that in G_N, all states l_N satisfy $l_N \in L_{G_\Omega} \setminus L_F^{G_\Omega}$, and in G_F, some states l_F satisfy $l_F \in L_{G_\Omega} \setminus L_F^{G_\Omega}$ but others satisfy $l_F \in L_F^{G_\Omega}$. In other words, all the states in G_N contain only the non-faulty information, and in G_F, some states encode the non-faulty information, while others encode the faulty information. Since an NSC/FSC PV $NFP_G(\Omega)$ is essentially the strict composition of an NSC G_N and an FSC G_F, picking any state $l_{NFP} = (l_N, l_F)$ in $NFP_G(\Omega)$, its component l_N contains only non-faulty information, while the component l_F contains faulty and non-faulty information. Then, picking any cycle $\rho_{NFP} : (l^1_N, l^1_F) \to \ldots \to (l^n_N, l^n_F) \to (l^1_N, l^1_F)$ of the $NFP_G(\Omega)$, it is obvious that there exist only two forms of the cycle: one satisfies $l^i_N, l^i_F \in L_{G_\Omega} \setminus L_F^{G_\Omega}$ for $i = 1, \ldots, n$, and the other satisfies $l^i_N \in L_{G_\Omega} \setminus L_F^{G_\Omega}$, $l^i_F \in$

$L_F^{G_\Omega}$ for $i = 1, \dots, n$. *$NFP_G(\Omega)$-indeterminate state* and *$NFP_G(\Omega)$-indeterminate cycle* of the $NFP_G(\Omega)$ are introduced that will be used later.

Definition 15. Given an NSC/FSC PV $NFP_G(\Omega)$, a state $(l_N, l_F) \in L_{NFP}$ is said to be an $NFP_G(\Omega)$-indeterminate state if $l_F \in L_F^{G_\Omega}$.

Note that for an $NFP_G(\Omega)$, there exist only two types of cycles: (i) cycles composed by the states that are all $NFP_G(\Omega)$-indeterminate; and (ii) those composed by the states that are all not $NFP_G(\Omega)$-indeterminate. Given a cycle in $NFP_G(\Omega)$, if all states of the cycle are $NFP_G(\Omega)$-indeterminate, there necessarily exist two cycles in G_N and G_F, respectively, one including only the states l_N satisfying $l_N \in L_{G_\Omega} \setminus L_F^{G_\Omega}$, and the other including only the states l_F satisfying $l_F \in L_F^{G_\Omega}$.

Definition 16. A cycle of $NFP_G(\Omega)$ is *$NFP_G(\Omega)$-indeterminate* if all the states of the cycle are $NFP_G(\Omega)$-indeterminate.

Proposition 2. *Let G be an LFA that satisfies Assumption H. It is diagnosable with respect to pattern Ω iff there is no $NFP_G(\Omega)$-indeterminate cycle in $NFP_G(\Omega)$.*

Proof. (only if) Assume that there exists an indeterminate cycle $cl_{NFP} : l_m^{NFP} \to l_{m+1}^{NFP} \cdots l_n^{NFP} \to l_m^{NFP}$ in the $NFP_G(\Omega)$, for $i = m, m+1, \dots, n$, $m, n \in \mathbb{N}$, $l_i^{NFP} \in L_{NFP}$. Let $\rho_{NFP} : l_0^{NFP} \to \cdots l_m^{NFP} \cdots l_n^{NFP} \to l_m^{NFP}$ be a run in $NFP_G(\Omega)$ and $w_{NFP} = w(\rho_{NFP})$.

From Definition 14, there necessarily exist two runs

$$\rho_N : l_0^N \to \cdots l_{m'}^N \cdots l_{n'}^N \to l_{m'}^N,$$
$$\rho_F : l_0^F \to \cdots l_{m''}^F \cdots l_{n''}^F \to l_{m''}^F,$$

with $w_N = w(\rho_N)$ and $w_F = w(\rho_F)$ in G_N and G_F respectively, such that w_{NFP}, w_N and w_F have the same observation. Observer that these two runs contain two cycles $cl_N : l_{m'}^N \cdots l_{n'}^N \to l_{m'}^N$ and $cl_F : l_{m''}^F \cdots l_{n''}^F \to l_{m''}^F$ such that $l_{i'}^N \in L_{G_\Omega} \setminus L_F^{G_\Omega}$, $i' = m', \dots, n'$, and $l_{i''}^F \in L_F^{G_\Omega}$, $i'' = m'', \dots, n''$.

Then, there exist two prefixes w_N^{pr}, w_F^{pr} of w_N and w_F respectively, with $(l_0^N, w_N^{pr}, l_{m'}^N) \in \delta_N$ and $(l_0^F, w_F^{pr}, l_{m''}^F) \in \delta_F$ such that $(l_{m'}^N, l_{m''}^F) = l_m^{NFP}$, and the two sequences w_N^{pr}, w_F^{pr} have the same observation, where δ_N (resp. δ_F) can be extended from E to E^* in the usual way. Also, there exist two strings $w_N^{cl} = w(cl_N)$ and $w_F^{cl} = w(cl_F)$ in G_N and G_F respectively, such that $w_N = w_N^{pr} w_N^{cl}$ and $w_F = w_F^{pr} w_F^{cl}$.

Repeating the cycle cl_{NFP} with any k times, $k \in \mathbb{N}$, there exist two sequences $w_N^{pr}(w_N^{cl})^k$ and $w_F^{pr}(w_F^{cl})^k$ in G_N and G_F having the same observation. By Definitions 12 and 13, there necessarily exist two runs in G_S corresponding to $w_N^{pr}(w_N^{cl})^k$ and $w_F^{pr}(w_F^{cl})^k$, respectively, having the same observation. By Definition 11, there exist two strings in G_Ω having the same observation: one accepted by Ω while another is not. Thus, there exist two strings in G having the same observation: one accepted by Ω while another is not, which completes the necessity.

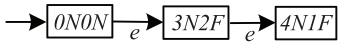

FIGURE 9 NSC/FSC PV $NFP_{G_2}(\Omega_1)$.

(if) Suppose that G is not diagnosable with respect to Ω, i.e. there exists $w^{pr} \in \mathcal{L}_\mathcal{A}(G)$, $w \in \mathcal{L}_\mathcal{A}(G)/w^{pr}$, and $w' \notin \mathcal{L}_\mathcal{A}(G)$ such that $\mathcal{P}(w^{pr}w) = \mathcal{P}(w')$, i.e. they have the same observation. By Definitions 6 and 11, there necessarily exist two runs in G_S corresponding to $w^{pr}w$ and w', respectively, one reaching the final state and the other not. By Definitions 12 and 13, there exist two runs $\rho_N = \rho(w_N)$ and $\rho_F = \rho(w_F)$ with the forms of

$$\rho_N : l_0^N \xrightarrow{w_N} l^N,$$
$$\rho_F : l_0^F \xrightarrow{w_F} l^F,$$

such that w_N and w_F have the same observation with $w^{pr}w$ and w', $l^N \in L_{G_\Omega} \setminus L_F^{G_\Omega}$, and $l^F \in L_F^{G_\Omega}$.

Moreover, there exist two runs $\rho_N^{pr} = \rho(w_N^{pr})$ and $\rho_F^{pr} = \rho(w_F^{pr})$ in G_N and G_F with the forms of

$$\rho_N^{pr} : l_0^N \xrightarrow{w_N^{pr}} l_{pr}^N,$$
$$\rho_F^{pr} : l_0^F \xrightarrow{w_F^{pr}} l_{pr}^F,$$

such that w_N^{pr} and w_F^{pr} have the same observation, $l_{pr}^N \in L_{G_\Omega} \setminus L_F^{G_\Omega}$, and $l_{pr}^F \in L_F^{G_\Omega}$, where w_N^{pr} is the prefix of w_N, and w_F^{pr} is the prefix of w_F. Then, according to Definition 14, there is a run in $NFP_G(\Omega)$ of the form $\rho_{NFP} : l_m^{NFP} \to l_n^{NFP}$ such that $l_m^{NFP} = (l_{pr}^N, l_{pr}^F)$, $l_n^{NFP} = (l^N, l^F)$, and $l_{pr}^F, l^F \in L_F^{G_\Omega}$.

By considering string $w^{pr}w$ of increasing symbol, the length of w_N and w_F also increases. Then, there eventually exist two states l_i^{NFP}, l_j^{NFP}, $m \le i \le j \le n$ such that $l_i^{NFP} = l_j^{NFP}$, and the set of states $\{l_r^{NFP}, r = i, i+1, \dots, j\}$ forms an indeterminate cycle. This ends the proof. □

Example 8. Consider an LFA G_2 in Figure 5(a) and the pattern Ω_1 in Figure 2(b). By Definition 14, the NSC/FSC PV $NFP_{G_2}(\Omega_1)$ is obtained in Figure 9. Based on Proposition 2, G_2 is diagnosable with respect to Ω_1 since there do not exist indeterminate cycles $NFP_{G_2}(\Omega_1)$ (see Definition 16).

In addition, consider also the LFA G_1 in Figure 2(a) and the pattern Ω_1 in Figure 2(b). By Definition 14, the NSC/FSC PV $NFP_{G_1}(\Omega_1)$ can be obtained in a similar way. Since there exists at least one indeterminate cycle in $NFP_{G_1}(\Omega_1)$, by Proposition 2, G_1 is not diagnosable with respect to the pattern Ω_1. For the sake of simplicity, the details are not pursued here.

4.3 | Discussion and comparison

It is worth noting that the methods in [14] and [16] can be used not only for single event fault scenarios but also for complex

TABLE 1 Literature review for the diagnosability and detection delay problems.

Objective \ Delay counting	Event	Observable event
Detection delay evolution	[14, 16, 39–42]	[43, 44]
K-diagnosability	[16, 40, 42, 45–47]	[6, 43]

TABLE 2 Space complexity comparison.

Structure	Complexity	Process
Synchronous product G_Ω	$O(n \times m)$	$G \to G_\Omega$
Diagnoser $d_G(\Omega)$	$O(2^{n \times m})$	$G \to G_\Omega \to d_G(\Omega)$
Standard PV $P_G(\Omega)$	$O(n^2 m^2)$	$G \to G_\Omega \to P_G(\Omega)$
FSC/NSC PV $NFP_G(\Omega)$	$O(n_O^2 m^2)$	$G \to G_\Omega \to G_S \to G_N, G_F \to NFP_G(\Omega)$

fault patterns diagnosability verification. Moreover, the proposed structures in [14] and [16] can be used for computing the detection delay. This notion has been adopted for DES by counting the number of events (Table 1). In our case, the detection delay can be counted in terms of the number of observable events based on the proposed NSC/FSC PV structure.

We compare the space complexity of the methods based on state isolation with its diagnoser, synchronous product with its diagnoser, and the standard PV, as shown in Table 2. Given an LFA G and a pattern Ω, let $|L| = n$ and $|S| = m$. Let us also introduce n_O as the number of system states that are reachable by at least one non silent event.

The number of reachable states of synchronous product verifier G_Ω of G and Ω is $n \times m$ at most. Consequently, the number of reachable states of diagnoser $d_G(\Omega)$ is $2^{n \times m}$ at most (see Section 3.2) and the number of reachable states of standard PV $P_G(\Omega)$ is $(n \times m)^2$ at most (see Section 4.1). From the perspective of complexity, the advantage of considering verification structures based on the silent closure of the synchronous product G_Ω is to remove all states that cannot be reached by an observable event. Consequently, depending on the number and location of the silent events in the system, the size of the final NSC/FSC PV can be highly reduced. According to Section 4.2, the number of states of silent closure G_S is $(n_O + 1) \times m$ at most and the numbers of reachable states of NSC G_N and FSC G_F are $(n_O + 1) \times (m - 1)$ and $(n_O + 1) \times m$ at most, respectively. So, finally, the number of states of NSC/FSC PV $NFP_G(\Omega)$ is $(n_O + 1)^2 \times m \times (m - 1)$ at most. The space complexity of all structures can be found in Table 2.

5 | ILLUSTRATIVE EXAMPLE

5.1 | Manufacturing cell model

This section presents an example of a flexible manufacturing system, which is a work cell composed of a robot that loads and unloads workpieces between input buffer I, output buffer O, and two machines M_1 and M_2 shown in Figure 10(a). TR_I, TR_O, TR_1, and TR_2 represent different areas in the work cell, related to I, O, M_1, and M_2, where the robot can move.

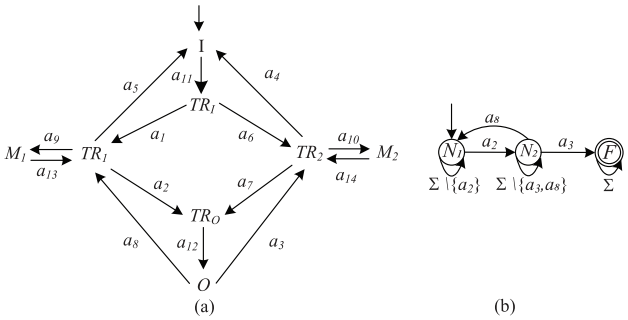

FIGURE 10 (a) Robot work cell G and (b) fault pattern Ω.

TABLE 3 Meaning of the states in Figure 10(a).

State	Meaning of the state
I	Robot stays in input buffer I
M_1	Robot stays in machine 1 M_1
M_2	Robot stays in machine 2 M_2
O	Robot stays in output buffer O
TR_I	Robot stays in safety buffer TR_I
TR_1	Robot stays in safety buffer TR_1
TR_2	Robot stays in safety buffer TR_2
TR_O	Robot stays in safety buffer TR_O

The robot work cell is modelled as an LFA G, as shown in Figure 10(a), where the set of the states is $L = \{I, O, M_1, M_2, TR_I, TR_O, TR_1, TR_2\}$, the set of the events is $\Sigma = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}\}$. Specifically, Tables 3 and 4 describe the meanings of the states and the events in the work cell, respectively. Table 4 (Scenario 1) also outlines the labels that correspond to the events, defined as $Lab(a_9) = m_1$, $Lab(a_{10}) = m_2$, $Lab(a_{11}) = i$, and $Lab(a_{12}) = o$. It is worth noting that this work cell focuses on the behaviours of the robot. Therefore, each state and event described in Tables 3 and 4 characterizes the specific details of the robot.

5.2 | Fault pattern and diagnosability analysis

In the configuration of this work cell, it is not allowed for the robot to execute behaviours that move from areas TR_1 to TR_2 in the counter-clockwise direction, i.e. the fault pattern of interest is the occurrence of the event a_2 followed by the occurrence of the event a_{12} and then by a_3, as shown in Figure 10(b).

In the following, we analyze the diagnosability of the work cell with respect to the fault pattern Ω, according to the NSC/FSC PV approach proposed in Section 4.2. For this purpose, we compute first the silent closure G_S, then the faulty silent closure G_F (Figure 11), and the normal silent closure

TABLE 4 Meaning of the events in Figure 10(a) and the output labels.

Events	Meaning of the events	Labels in Scenario 1	Labels in Scenario 2
a_1	Robot left TR_I and arrived at TR_1	ε	ε
a_2	Robot left TR_1 and arrived at TR_O.	ε	ε
a_3	Robot left O and arrived at TR_2	ε	ε
a_4	Robot left TR_2 and arrived at I	ε	ε
a_5	Robot left TR_1 and arrived at I	ε	ε
a_6	Robot left TR_I and arrived at TR_2	ε	e_6
a_7	Robot left TR_2 and arrived at TR_O	ε	e_7
a_8	Robot left O and arrived at TR_1	ε	e_8
a_9	Robot arrived at M_1	m_1	m_1
a_{10}	Robot arrived at M_2	m_2	m_2
a_{11}	Robot left I and arrived at TR_I	i	i
a_{12}	Robot left TR_I and arrived at O	o	o
a_{13}	Robot left M_1 and arrived at TR_1	ε	ε
a_{14}	Robot left M_2 and arrived at TR_2	ε	ε

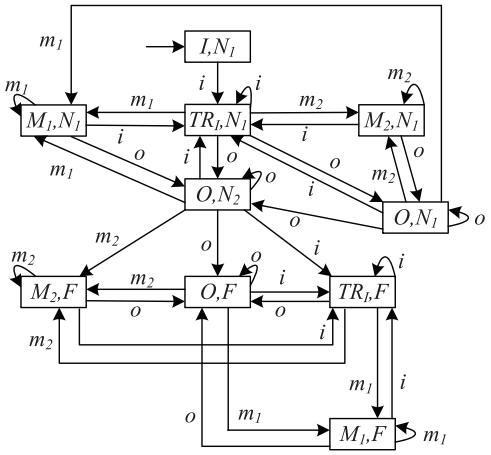

FIGURE 11 Faulty silent closure G_F.

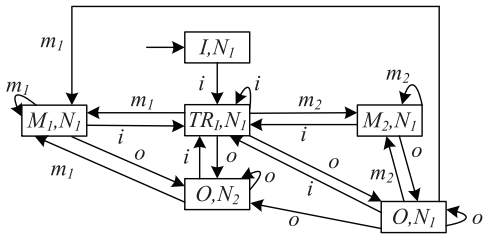

FIGURE 12 Normal silent closure G_N.

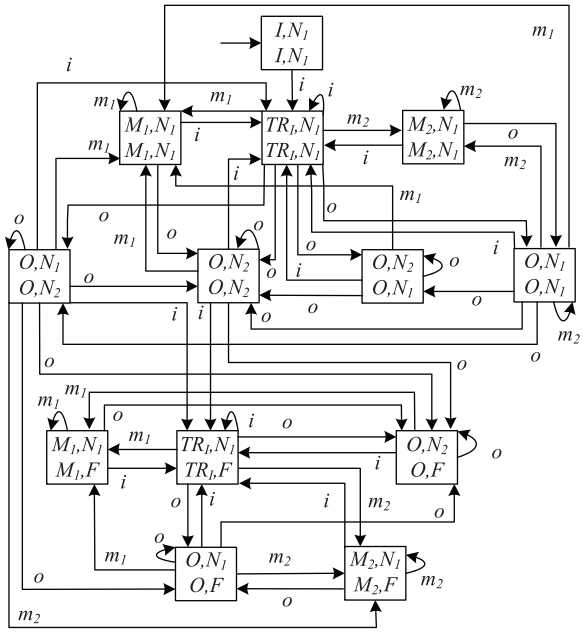

FIGURE 13 NSC/FSC PV of Scenario 1.

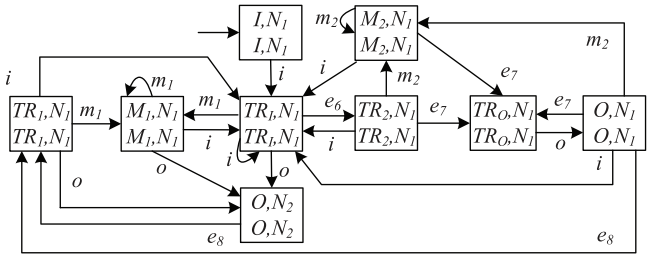

FIGURE 14 NSC/FSC PV of Scenario 2.

G_N (Figure 12). Finally, the NSC/FSC PV is constructed (Figure 13). It is obvious that there exists at least one indeterminate cycle $(M_2, N_1), (M_2, F) \xrightarrow{m_2} (M_2, N_1), (M_2, F)$. Consequently, by Proposition 2, we can conclude that the robot work cell G is not diagnosable with respect to the fault pattern $\mathbf{\Omega}$.

Observe that by adding three more observable labels: $\text{Lab}(a_6) = e_6$, $\text{Lab}(a_7) = e_7$, $\text{Lab}(a_8) = e_8$, $\text{Lab}(a_9) = m_1$, $\text{Lab}(a_{10}) = m_2$, $\text{Lab}(a_{11}) = i$, and $\text{Lab}(a_{12}) = o$ (Scenario 2 in Table 4), the conclusion about diagnosability improves. The NSC/FSC PV of Scenario 2 is detailed in Figure 14: there is no indeterminate cycle and the robot work cell G is diagnosable with respect to the fault pattern $\mathbf{\Omega}$.

In order to discuss the complexity aspects, let us first mention that for both scenarios we have $n = |L| = 8$ and $m = |S| = 3$. Then for the first scenario, $n_O = 4$ whereas for the second scenario $n_O = 7$. Then, Table 5 details the number of states and transitions of the main structures used for fault pattern diagnosability verification with the proposed approach and for the two scenarios of this case study. Let us first notice that the exact sizes of the resulting NSC/FSC PVs (resp. 13 for scenario 1 and 9 for scenario 2) are much lower than the worst case sizes (resp. 144 for scenario 1 and 441 for scenario 2) provided by the complexity analysis. These worst case complexities also improve the worst case complexity of the pair verifier (576

TABLE 5 Space complexity analysis (number of states × number of transitions).

Structure	Scenario 1	Scenario 2
Synchronous product G_Ω	18 × 31	18 × 31
Silent closure $G_S(\Omega)$	10 × 37	16 × 47
Normal silent closure $G_N(\Omega)$	6 × 20	9 × 23
Faulty silent closure $G_F(\Omega)$	10 × 37	16 × 47
NSC/FSC PV $\text{NFP}_G(\Omega)$	13 × 57	9 × 23

for both scenarios). Then, in Table 5, we observe also that the normal and fault silent closures for Scenario 2 are larger than those for Scenario 1. However, the size of the final NSC/FSC PV structure for Scenario 2 is smaller than that of Scenario 1. This indicates that even if n_O provides a first indication of the gain in complexity, it is not enough to evaluate the advantage of the proposed approach for a particular system and scenario.

6 | CONCLUSION

This paper deals with the fault pattern diagnosis of discrete event systems, which includes fault pattern detection and diagnosability checking. The contribution has focused at first on the space complexity issues related to the construction of the related verification structures. For that purpose, we propose a synchronous product verifier, suitable to manipulate complex patterns in a systematic way. In order to verify the fault pattern diagnosability, an NSC/FSC PV is constructed based on the silent closure of the synchronous product verifier. The proposed method requires polynomial time at most, offering computational advantages for systems with numerous silent events. A case study of a flexible manufacturing system illustrates our approach and shows that it is suitable for real systems and practical situations.

However, we should mention some limitations that also open future studies. The main question is how to define exactly the class of the systems that will benefit at first from the proposed approach. The number but also the location of the silent events in the model play important roles. Providing some guidelines for practitioners belongs to our further works. Another limitation of our perspective is that it is based on an explicit model of the faulty behaviours. Such behaviours are often more difficult to characterize and one should discuss how faulty model based approaches can be combined with healthy model based approaches to improve efficiency. Our future work will also consider diagnosis issues of timed patterns characterized by a sequence of events, occurring in a given order at specific values of time or within specific time intervals.

AUTHOR CONTRIBUTIONS

Ye Liang: Conceptualization; data curation; formal analysis; investigation; methodology; software; validation; visualization; writing—original draft; writing—review and editing. **Dimitri Lefebvre**: Conceptualization; formal analysis; methodology; project administration; resources; supervision; validation; writing—review and editing. **Zhiwu Li**: Funding acquisition; project administration; supervision; writing—review and editing.

ACKNOWLEDGEMENTS

This work was supported by the National Key R&D Program of China under Grant 2018YFB1700104.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Ye Liang https://orcid.org/0000-0003-2208-9897

REFERENCES

1. Cassandras, C.G., Lafortune, S.: Introduction to Discrete Event Systems. Springer, Berlin, Heidelberg (2008)
2. Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., Teneketzis, D.: Diagnosability of discrete-event systems. IEEE Trans. Autom. Control 40(9), 1555–1575 (1995)
3. Zhou, D., Zhao, Y., Wang, Z., He, X., Gao, M.: Review on diagnosis techniques for intermittent faults in dynamic systems. IEEE Trans. Ind. Electron. 67(3), 2337–2347 (2019)
4. Contant, O., Lafortune, S., Teneketzis, D.: Diagnosis of intermittent faults. Discrete Event Dyn. Syst. 14(2), 171–202 (2004)
5. Biswas, S.: Diagnosability of discrete event systems for temporary failures. Comput. Electr. Eng. 38(6), 1534–1549 (2012)
6. Jéron, T., Marchand, H., Pinchinat, S., Cordier, M.-O.: Supervision patterns in discrete event systems diagnosis. In: Proceedings of the 8th International Workshop on Discrete Event Systems, pp. 262–268. IEEE, Piscataway, NJ (2006)
7. Ye, L., Dague, P.: A general algorithm for pattern diagnosability of distributed discrete event systems. In: Proceedings of the 24th IEEE International Conference on Tools with Artificial Intelligence, pp. 130–137. IEEE, Piscataway, NJ (2012)
8. Yan, Y., Ye, L., Dague, P.: Diagnosability for patterns in distributed discrete event systems. In: Proceedings of the 21st International Workshop on Principles of Diagnosis, pp. 345–352. PHM Society, Rochester, NY (2010)
9. Gougam, H.E., Pencolé, Y., Subias, A.: Diagnosability analysis of patterns on bounded labeled prioritized Petri nets. Discrete Event Dyn. Syst. 27(1), 143–180 (2017)
10. Jiang, S., Kumar, R.: Failure diagnosis of discrete-event systems with linear-time temporal logic specifications. IEEE Trans. Autom. Control 49(6), 934–945 (2004)
11. Wang, Z., Wang, J., Wang, Y.: An intelligent diagnosis scheme based on generative adversarial learning deep neural networks and its application to planetary gearbox fault pattern recognition. Neurocomputing 310, 213–222 (2018)
12. Watanabe, A.T., Sebem, R., Leal, A.B., Hounsell, M.d.S.: Fault prognosis of discrete event systems: an overview. Annu. Rev. Control 51, 100–110 (2021)
13. Boussif, A., Ghazel, M., Basilio, J.C.: Intermittent fault diagnosability of discrete event systems: an overview of automaton-based approaches. Discrete Event Dyn. Syst. 31(1), 59–102 (2021)
14. Qiu, W., Kumar, R.: Decentralized failure diagnosis of discrete event systems. IEEE Trans. Syst. Man Cybern. Part A: Syst. Humans 36(2), 384–395 (2006)

15. Carvalho, L.K., Moreira, M.V., Basilio, J.C.: Comparative analysis of related notions of robust diagnosability of discrete-event systems. Annu. Rev. Control 51, 23–36 (2021)
16. Yoo, T.-S., Garcia, H.E.: Diagnosis of behaviors of interest in partially-observed discrete-event systems. Syst. Control Lett. 57(12), 1023–1029 (2008)
17. Lin, Y., Stadtherr, M.A.: Fault detection in continuous-time systems with uncertain parameters. In: Proceedings of the 2007 American Control Conference, pp. 3216–3221. IEEE, Piscataway, NJ (2007)
18. Lin, Y., Stadtherr, M.A.: Fault detection in nonlinear continuous-time systems with uncertain parameters. AlChE J. 54(9), 2335–2345 (2008)
19. Boem, F., Ferrari, R.M., Parisini, T., Polycarpou, M.M.: Distributed fault diagnosis for continuous-time nonlinear systems: The input–output case. Annu. Rev. Control 37(1), 163–169 (2013)
20. Cheng, P., He, S., Stojanovic, V., Luan, X., Liu, F.: Fuzzy fault detection for Markov jump systems with partly accessible hidden information: an event-triggered approach. IEEE Trans. Cybern. 52(8), 7352–7361 (2021)
21. Cheng, P., Chen, M., Stojanovic, V., He, S.: Asynchronous fault detection filtering for piecewise homogenous Markov jump linear systems via a dual hidden Markov model. Mech. Syst. Sig. Process. 151, 107353 (2021)
22. Cheng, P., He, S., Dong, H., Chen, W., Zhang, W.: Extended state observer-based finite-region control for 2-D Markov jump systems. Int. J. Robust Nonlinear Control 33(2), 1010–1026 (2023)
23. Genc, S., Lafortune, S.: Diagnosis of patterns in partially-observed discrete-event systems. In: Proceedings of the 45th IEEE Conference on Decision and Control, pp. 422–427. IEEE, Piscataway, NJ (2006)
24. Debouk, R., Lafortune, S., Teneketzis, D.: Coordinated decentralized protocols for failure diagnosis of discrete event systems. Discrete Event Dyn. Syst. 10(1), 33–86 (2000)
25. Jiang, S., Huang, Z., Chandra, V., Kumar, R.: A polynomial algorithm for testing diagnosability of discrete-event systems. IEEE Trans. Autom. Control 46(8), 1318–1321 (2001)
26. Yoo, T.-S., Lafortune, S.: Polynomial-time verification of diagnosability of partially observed discrete-event systems. IEEE Trans. Autom. Control 47(9), 1491–1495 (2002)
27. Jiang, S., Kumar, R., Garcia, H.E.: Diagnosis of repeated/intermittent failures in discrete event systems. IEEE Trans. Rob. Autom. 19(2), 310–323 (2003)
28. Hu, Y., Ma, Z., Li, Z.: Design of supervisors for active diagnosis in discrete event systems. IEEE Trans. Autom. Control 65(12), 5159–5172 (2020)
29. Boussif, A., Ghazel, M.: Diagnosability analysis of intermittent faults in discrete event systems using a twin-plant structure. Int. J. Control Autom. Syst. 18(3), 682–695 (2020)
30. Chen, J., Kumar, R.: Polynomial test for stochastic diagnosability of discrete-event systems. IEEE Trans. Autom. Sci. Eng. 10(4), 969–979 (2013)
31. Hadjicostis, C.N.: Estimation and Inference in Discrete Event Systems: A Model-Based Approach with Finite Automata. Springer, Cham (2019)
32. Wang, Y., Yoo, T.-S., Lafortune, S.: Diagnosis of discrete event systems using decentralized architectures. Discrete Event Dyn. Syst. 17(2), 233–263 (2007)
33. Li, B., Basilio, J.C., Khlif-Bouassida, M., Toguyéni, A.: Polynomial time verification of modular diagnosability of discrete event systems. IFAC-PapersOnLine 50(1), 13618–13623 (2017)
34. Ran, N., Su, H., Giua, A., Seatzu, C.: Codiagnosability analysis of bounded Petri nets. IEEE Trans. Autom. Control 63(4), 1192–1199 (2017)
35. Hu, Y., Ma, Z., Li, Z.: An improved approach to test diagnosability of bounded petri nets. IEEE/CAA J. Autom. Sin. 4(2), 297–303 (2017)
36. Ye, L., Dague, P., Nouioua, F.: An improved approach to test diagnosability of bounded Petri nets. In: Proceedings of the 52nd IEEE Conference on Decision and Control, pp. 5009–5015. IEEE, Piscataway, NJ (2013)
37. Moreira, M.V., Jesus, T.C., Basilio, J.C.: Polynomial time verification of decentralized diagnosability of discrete event systems. IEEE Trans. Autom. Control 56(7), 1679–1684 (2011)
38. Liang, Y., Lefebvre, D., Li, Z.: Fault pattern diagnosis of discrete-event systems by means of logical verifiers. In: Proceedings of the 11th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, pp. 551–556. IEEE, Piscataway, NJ (2022)
39. Yoo, T.-S., Garcia, H.: Computation of fault detection delay in discrete-event systems. In: Proceedings of the 14th International Workshop on Principles of Diagnosis, pp. 207–212. ACM, New York (2003)
40. Chouchane, A., Ghazel, M., Boussif, A.: K-diagnosability analysis of bounded and unbounded Petri nets using linear optimization. Automatica 147, 110689 (2023)
41. Shu, S., Lin, F.: Delayed detectability of discrete event systems. IEEE Trans. Autom. Control 58(4), 862–875 (2012)
42. Takai, S., Kumar, R.: A generalized framework for inference-based diagnosis of discrete event systems capturing both disjunctive and conjunctive decision-making. IEEE Trans. Autom. Control 62(6), 2778–2793 (2016)
43. Sayed Mouchaweh, M., Philippot, A., Carré-Ménétrier, V., Riera, B.: Detectability and diagnosability of discrete event systems. In: Proceedings of the 2nd International Conference on Informatics in Control, Automation and Robotics, vol. 5, pp. 14–17. IEEE, Piscataway, NJ (2005)
44. Liu, Y., Liu, Z., Yin, X., Li, S.: An improved approach for verifying delayed detectability of discrete-event systems. Automatica 124, 109291 (2021)
45. Basile, F., Chiacchio, P., De Tommasi, G.: On K-diagnosability of petri nets via integer linear programming. Automatica 48(9), 2047–2058 (2012)
46. Cabasino, M.P., Giua, A., Lafortune, S., Seatzu, C.: A new approach for diagnosability analysis of Petri nets using verifier nets. IEEE Trans. Autom. Control 57(12), 3104–3117 (2012)
47. Yokota, S., Yamamoto, T., Takai, S.: Computation of the delay bounds and synthesis of diagnosers for decentralized diagnosis with conditional decisions. Discrete Event Dyn. Syst. 27, 45–84 (2017)

How to cite this article: Liang, Y., Lefebvre, D., Li, Z.: Silent closure based pair verifier for fault pattern diagnosis of discrete event systems. IET Control Theory Appl. 1–11 (2023). https://doi.org/10.1049/cth2.12593