Noname manuscript No.
(will be inserted by the editor)

Diagnosability and Attack Detection for Discrete Event Systems under Sensor Attacks

Feng Lin · Stéphane Lafortune · Caisheng Wang

Received: 7 July 2023 / Accepted: 29 June 2024

Abstract This paper extends the theory of diagnosability by investigating fault diagnosis in discrete event systems under sensor attacks using finite-state automata as models. It assumes that an attacker has compromised the communication channel between the system's sensors and the diagnostic engine. While the general attack model utilized by the attacker has been previously studied in the context of supervisory control, its application to fault diagnosis remains unexplored. The attacker possesses the capability to substitute each compromised observable event with a string from an attack language. The attack model incorporates event insertion and deletion, as well as static and dynamic attacks. To formally capture the diagnostic engine's ability to identify faults in the presence of the attacker, a novel concept called CA-diagnosability is introduced. This extends the existing notions of CA-controllability and CA-observability. A testing procedure for CA-diagnosability is developed, and its correctness is proven. Some sufficient conditions for CA-diagnosability that can be easily checked are also proposed and proved. The paper then investigates conditions under which the role of an attacker can be reverted from malicious

This work is supported in part by the US National Science Foundation under grants ECCS-2146615 and ECCS-2144416.

Feng Lin
Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202, USA
Tel.: +1-313-5773428
E-mail: flin@wayne.edu

Stéphane Lafortune
Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA
E-mail: stephane@umich.edu

Caisheng Wang
Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202, USA
E-mail: cwang@wayne.edu

to benevolent, that is, to help the diagnoser to diagnose faults. The paper further applies diagnosability theory to investigate conditions under which the presence of the attacker can be detected.

Keywords Discrete event systems, cyber attacks, diagnosability
Statements and Declarations: The authors have no competing interests to declare that are relevant to the content of this article.

1 Introduction

We consider the standard set-up of event diagnosis in discrete event systems modeled by finite-state automata, as in [1]. However, we assume that an attacker has compromised a subset of the system's sensors. This attacker can initiate a sensor deception attack, i.e., it can edit the string of events input to the diagnostic engine. Such attacks have been the subject of increasing attention in state estimation, fault diagnosis, and supervisory control of discrete event systems (DES) in recent years [2–8]; a list of references can be found in a recent book [9] and a survey/tutorial paper [10]. The increased interest in these problems is motivated by concerns regarding cyber-attacks on both cyber and cyber-physical systems (CPS) [11–13]. In the context of CPS, their higher-level control logic is often modeled as discrete transition systems and thus studied in the context of DES [14–18].

In this paper, we consider a general nondeterministic attack model on the sensors, where each compromised observable event can be edited, in a nondeterministic manner, by a string in a language, thereby capturing event insertion, deletion, and replacement. Moreover, this editing of compromised events can be dynamic in the sense that the attack language is allowed to vary for each compromised event, based on the string of events executed so far. This attack model on sensors was introduced in [19] in the context of supervisory control, where it led to the formalization of the notions of CA-controllability and CA-observability (where CA stands for "cyber-attack"); these notions extend the standard controllability and observability properties to the attack scenario under consideration. In this paper, we name the sensor attack model from [19] as the ALTER model[1] and study event diagnosis under this general attack model. Specifically, we study diagnostic performance in the presence of ALTER attacks, including both static and dynamic attacks. In our problem formulation, the DES of interest is partially observed by a *diagnostic engine* whose goal is to detect each occurrence of an unobservable event of interest (e.g., a *fault* event) in a bounded number of events after the occurrence. The property of *diagnosability*, originally defined in [1], captures this objective in the absence of an attacker. Various techniques exist in the literature to test diagnosability (see, e.g., [16]). We do not assume a specific diagnostic engine in this paper, but it could be a *diagnoser automaton*, as defined in [1].

[1] One could think of the acronym ALTER as capturing the keywords: Attack Language, Transition-BasEd, Replacement.

We first define the notion of CA-diagnosability that captures the ability to still diagnose the occurrences of the unobservable event of interest in the presence of an ALTER attack. This extends the results in [19], which focused on supervisory control and introduced the corresponding properties of CA-controllability and CA-observability, to the realm of event diagnosis. We then present a testing procedure for CA-diagnosability that is based on transforming the system model and testing the standard property of diagnosability on the transformed model. The transformation procedure results in an *extended* automaton that embeds the possible actions of the attacker into the original automaton model of the system[2]. We prove the correctness of this approach under some general assumptions about the attacker. This approach also allows us to design a CA-diagnoser for a system under sensor attacks based on the extended automaton using the conventional design method, if CA-diagnosability is satisfied.

Owing to the ALTER attack model considered in this paper and to the formulation and study of the property of CA-diagnosability, our results differ in nature and complement related recent work on state estimation and diagnosis of DES under lossy or tampered observations, as well as codiagnosability of networked DES; see, e.g., [20–25]. ALTER attacks, as defined in Section 3, allow any transition labeled by a compromised observable event to be replaced by a string in a corresponding attack language. The notion of attack language captures event deletion, insertion, and replacement, as well as nondeterminism of the attacker. These features distinguish ALTER from the other attack models in the works mentioned above.

We then investigate the question that if a system is diagnosable without sensor attacks, will it remain CA-diagnosable under sensor attacks? We derive some sufficient conditions on sensor attacks under which diagnosability implies CA-diagnosability. One of the conditions is very easy to check. If this condition is satisfied, then the sensor attacks can be overcome in the sense that a new CA-diagnoser can be designed to diagnose the occurrences of the unobservable event of interest in the presence of the attacker.

Next, we revert the role of the attacker from malicious to benevolent, i.e., we want to design a helper (rather than an attacker) to help a diagnoser to diagnose the occurrences of the unobservable event of interest which the diagnoser cannot do without the helper. Theoretically, this means to find a helper so that although G is not diagnosable, it is CA-diagnosable with the helper. We show that, in order for this to be possible, the helper needs to observe more events than the diagnoser.

In the last part of the paper, we use the diagnosability theory itself to detect the presence of an attacker, using the same general ALTER model. We present a second model transformation procedure, where the unobservable event to diagnose is an event that is introduced in the attacks of the considered attacker. Under an assumption about the ALTER model, we show that if the

[2] The technique of modifying the system model to embed attacks has been used in other works that have considered attacks on DES, but the details often differ based on the type of attack considered.

attacker remains *stealthy*, as defined in the paper, then its presence will not be detected by the diagnostic engine. The stealthiness of an attacker holds whenever its edit actions remain contained in the original observed system language. The special features of our ALTER model make these results novel and distinct as compared to prior works on attack detection and stealthy (or covert) attacks in the context of sensor deception attacks, such as the work in [6, 22, 26–28].

The contributions described above extend the theory of diagnosability of discrete event systems to account for the presence of sensor deception attacks under a general attack model. In addition, we illustrate some of the results in this paper with an example of a protection relay and a circuit breaker in a power system, where the goal is to diagnose failures of the protection relay or circuit breaker under sensor attacks. Discrete event system theory has been applied to power systems before [29–34]; our focus on attacks on diagnosers differentiates our results from these past works.

This paper is organized as follows. Section 2 presents some necessary background material on the theory of diagnosability of DES. Next, Section 3 describes the general attack model considered in this paper. A prototypical power system example is introduced in Section 4, where the faults of interest pertain to the circuit breaker and the protection relay. The results on CA-diagnosability and its verification are presented in Section 5; the main result therein is Theorem 1. The power system example is revisited in this section. Section 6 presents some sufficient conditions on attack models under which diagnosability implies CA-diagnosability. Section 7 considers how to design a helper to help a diagnoser to diagnose faults that the diagnoser cannot diagnose without the helper. Then, the results on diagnosing the attack itself, Theorems 5 and 6, are presented in Section 8. Section 9 concludes the paper.

A preliminary and partial version of this paper, without proofs and without Sections 6 and 7 among other differences, appears in [35].

2 Diagnosability of Discrete Event Systems

Let us review the theory of diagnosability of DES in this section. As usual, the DES of interest is modeled by a finite deterministic automaton [16,17,36]:

$$G = (Q, \Sigma, \delta, q_o, Q_m),$$

where Q is the state set; Σ is the event set; $\delta : Q \times \Sigma \to Q$ is the transition function, generally, a partial function; q_o is the initial state; and Q_m is the marked state set. In this paper, we assume that all states in G are marked, that is, $Q_m = Q$.

We use Σ^* to denote the set of all finite strings over Σ. The transition function δ is extended to strings, that is, $\delta : Q \times \Sigma^* \to Q$ in the usual way [16]. If $\delta(q, s)$ is defined, we denote it by $\delta(q, s)!$. The language *generated* by G is the set of all strings defined in G from the initial state:

$$L(G) = \{s \in \Sigma^* : \delta(q_o, s)!\}.$$

The language *marked* by G is the set of all strings defined in G from the initial state and end in a marked state:

$$L_m(G) = \{s \in L(G) : \delta(q_o, s) \in Q_m\}.$$

In general, a language $K \subseteq \Sigma^*$ is a set of strings. The prefix closure of K is the set of prefixes of strings in K. A language is prefix-closed if it equals its prefix closure. The prefix closure of K is denoted by $\overline{K}$. By definition, $L(G)$ is prefix-closed. The length of a string $s \in \Sigma^*$ is denoted by $|s|$. The cardinality (the number of its elements) of a set $x \subseteq Q$ is denoted by $|x|$.

The set of observable events is denoted by Σ_o ($\subseteq \Sigma$). $\Sigma_{uo} = \Sigma - \Sigma_o$ is the set of unobservable events. With a slight abuse of notation, the set of all transitions present in G is also denoted by δ: $\delta = \{(q, \sigma, q') : \delta(q, \sigma) = q'\}$. We will use ε-transition (q, ε, q'), where ε is the empty string, when we consider the partial observation below.

As in [1], we make the following two assumptions on G for the rest of this paper:

A1. G is live (can always generate more events), that is, $(\forall q \in Q)(\exists \sigma \in \Sigma)\delta(q, \sigma)!$.
A2. There are no cycles of unobservable events in G, that is, $(\forall q \in Q)(\forall s \in \Sigma^*)\delta(q, s) = q \wedge |s| > 0 \Rightarrow s \notin \Sigma_{uo}^*$.

These standard assumptions can be mitigated at the cost of extra technicalities to (i) deal with terminating traces and (ii) properly handle cycles of unobservable events; see, e.g., Chapter 5 in [37]. To keep the treatment simpler in this paper, we prefer to avoid these complications.

Faults in G are represented by some events. The set of events representing faults is denoted by $\Sigma_f \subset \Sigma$. To diagnose faults, a diagnoser is used, which can observe observable events. We assume that all fault events are unobservable, that is, $\Sigma_f \subseteq \Sigma_{uo}$; otherwise, the diagnosis of the fault events is trivial. Observation is described by the natural projection $P : \Sigma^* \to \Sigma_o^*$ defined as

$$P(\varepsilon) = \varepsilon, \quad P(s\sigma) = \begin{cases} P(s)\sigma & \text{if } \sigma \in \Sigma_o \\ P(s) & \text{if } \sigma \in \Sigma_{uo} \end{cases}.$$

P is extended to a language $K \subseteq L(G)$ as $P(K) = \{P(s) : s \in K\}$. The inverse mapping of P is defined as $P^{-1}(w) = \{s \in \Sigma^* : P(s) = w\}$. P^{-1} is extended to a language $J \subseteq P(L(G))$ as $P^{-1}(J) = \{s \in \Sigma^* : P(s) \in J\}$.

Denote the set of strings in $K \subseteq L(G)$ whose last event is a fault event as

$$\Psi(K) = \{s\sigma \in K : \sigma \in \Sigma_f\}.$$

The goal of diagnosis is to determine the occurrence of any string in $\Psi(L(G))$ after finite delays measured by the number of events occurred afterwards. Formally, diagnosability is defined in [1] as follows.

Definition 1 [1]
A DES G is diagnosable with respect to P if

$$\begin{aligned}&(\exists n \in \mathcal{N})(\forall s \in \Psi(L(G)))(\forall u \in L(G)/s)\\&|u| \geq n \Rightarrow (\forall v \in P^{-1}(P(su)) \cap L(G))\Sigma_f \in v,\end{aligned} \tag{1}$$

where $\mathcal{N}$ is the set of natural numbers, $L(G)/s$ denotes the post language after s:

$$L(G)/s = \{u \in \Sigma^* : su \in L(G)\}$$

and $\Sigma_f \in v$ means v contains at least one fault event:

$$(\exists \sigma \in \Sigma_f)v = v'\sigma v'',$$

where v' and v'' are substrings in Σ^*.

In [1], different types of faults are considered. For simplicity, we consider only one type of fault in this paper. The results of this paper can be extended to multi-type faults at the expense of more complex notations.

We consider networked DES under sensor attacks in the next section.

3 Networked Discrete Event Systems under Sensor Attacks

The terminology "ALTER" is used in this paper to name the model introduced in [19] and reviewed in [10]. In this section, we review the ALTER model and then propose a method to implement the ALTER model.

3.1 ALTER Sensor Attack Model

As in [19], let us denote the set of observable events and transitions that may be attacked by $\Sigma_o^a \subseteq \Sigma_o$ and $\delta^a = \{(q, \sigma, q') \in \delta : \sigma \in \Sigma_o^a\}$, respectively. We call events $\sigma \in \Sigma^a$ attackable events and transitions $tr = (q, \sigma, q') \in \delta^a$ attackable transitions.

For a given attackable transition $tr = (q, \sigma, q') \in \delta^a$, we assume that an attacker can change the event σ to any string in an attack language $A_{tr} \subseteq \Sigma^*$. In other words, if a string $s = \sigma_1\sigma_2..., \sigma_{|s|} \in L(G)$ occurs in G, the set of possible strings after attacks, denoted by $\Theta^a(s)$, is obtained as follows. Denote $q_k = \delta(q_o, \sigma_1 \cdots \sigma_k), k = 1, 2, ..., |s|$, then

$$\Theta^a(s) = L_1L_2...L_{|s|},$$

where

$$L_k = \begin{cases} \{\sigma_k\} \text{ if } tr = (q_{k-1}, \sigma_k, q_k) \notin \delta^a \\ A_{tr} \quad \text{if } tr = (q_{k-1}, \sigma_k, q_k) \in \delta^a \end{cases} \tag{2}$$

Note that $\Theta^a(s)$ may contain more than one string. Hence, Θ^a is a mapping $\Theta^a : L(G) \to 2^{\Sigma^*}$.

Note also that this general definition allows for nondeterministic attacks and includes the following special cases. (1) No attack: if $\sigma \in A_{tr}$ and σ is altered to σ (no change), then there is no attack. (2) Deletion: if the empty string $\varepsilon \in A_{tr}$ and σ is altered to ε, then σ is deleted. (3) Replacement: if $\alpha \in A_{tr}$ and σ is altered to α, then σ is replaced by α. (4) Insertion: if $\sigma\alpha \in A_{tr}$ (or $\alpha\sigma \in A_{tr}$) and σ is altered to $\sigma\alpha$ (or $\alpha\sigma$), then α is inserted.

The observation mapping under partial observation and sensor attacks is then given by

$$\Phi^a(s) = P \circ \Theta^a(s) = P(\Theta^a(s)). \tag{3}$$

Hence, Φ^a is a mapping $\Phi^a : L(G) \to 2^{\Sigma_o^*}$.

We extend Θ^a and Φ^a from strings s to languages $K \subseteq L(G)$ in the usual way as

$$\begin{aligned} \Theta^a(K) &= \{\Theta^a(s) : s \in K\} \\ \Phi^a(K) &= \{\Phi^a(s) : s \in K\}. \end{aligned} \tag{4}$$

We add sensor attacks to G as follows. For each transition $tr \in \delta^a$, let us assume that A_{tr} is marked by an automaton F_{tr}, that is, $A_{tr} = L_m(F_{tr})$ for some

$$F_{tr} = (Q_{tr}, \Sigma, \delta_{tr}, q_{o,tr}, Q_{m,tr}).$$

Note that not all states in F_{tr} are marked. We assume that F_{tr} is trim, that is, all states are accessible from $q_{o,tr}$ and co-accessible to $Q_{m,tr}$.

We replace each attackable transition $tr = (q, \sigma, q') \in \delta^a$ by (q, F_{tr}, q') as follows.

$$G_{tr\to(q,F_{tr},q')} = (Q \cup Q_{tr}, \Sigma, \delta_{tr\to(q,F_{tr},q')}, q_o)$$

where $\delta_{tr\to(q,F_{tr},q')} = (\delta - \{(q,\sigma,q')\}) \cup \delta_{tr} \cup \{(q,\varepsilon,q_{o,tr})\} \cup \{(q_{m,tr},\varepsilon,q') : q_{m,tr} \in Q_{m,tr}\}$. In other words, $G_{tr\to(q,F_{tr},q')}$ contains all transitions in δ and δ_{tr}, except (q,σ,q'), plus the ε-transitions from q to the initial state of F_{tr} and from marked states of F_{tr} to q'. If F_{tr} has only one marked state, that is, $Q_{m,tr} = \{q_{m,tr}\}$, then we can use a shortcut by merging q and q' with $q_{0,tr}$ and $q_{m,tr}$, respectively, without the ε-transitions $(q,\varepsilon,q_{0,tr})$ and $(q_{m,tr},\varepsilon,q')$.

Denote the extended automaton obtained after replacing all attackable transitions as

$$G^e = (Q^e, \Sigma, \delta^e, q_o, Q_m^e) = (Q \cup \tilde{Q}, \Sigma, \delta^e, q_o, Q)$$

where $\tilde{Q} = \cup_{tr\in\delta^a} Q_{tr}$ is the set of states added during the replacement and $Q_m^e = Q$ is the set of marked states. Note that G^e is a nondeterministic automaton and its transition function is a mapping $\delta^e : Q^e \times \Sigma \to 2^{Q^e}$. From the construction of G^e, it is obvious that

$$L_m(G^e) = \Theta^a(L(G)), \quad L(G^e) = \overline{\Theta^a(L(G))}. \tag{5}$$

To describe the partial observation, we replace unobservable transitions in G^e by ε-transitions and denote the resulting automaton as

$$G^e_\varepsilon = (Q^e, \Sigma_o, \delta^e_\varepsilon, q_o, Q^e_m) = (Q \cup \tilde{Q}, \Sigma_o, \delta^e_\varepsilon, q_o, Q)$$

where $\delta^e_\varepsilon = \{(q, \sigma, q') : (q, \sigma, q') \in \delta^e \wedge \sigma \in \Sigma_o\} \cup \{(q, \varepsilon, q') : (q, \sigma, q') \in \delta^e \wedge \sigma \notin \Sigma_o\}$. Clearly, G^e_ε is a nondeterministic automaton.

G^e_ε marks the language $\Phi^a(L(G))$ because

$$L_m(G^e_\varepsilon) = P(L_m(G^e)) = P(\Theta^a(L(G))) = \Phi^a(L(G)). \tag{6}$$

3.2 Implementation of the ALTER Model

We now consider how to implement the ALTER model, namely, how to obtain a plant model G such that the attack model, based on the capabilities of the attacker, is consistent with the definitions in Section 3.1. Note that the implementation is not trivial, because ALTER with the set of attack languages $\{A_{tr} : tr \in \delta^a\}$ is transition-based with respect to the transitions of G while the observations of the attackers are event-based.

We need to consider two possible cases. In the first case, sensor attacks are "static," that is, for any two transitions $tr_1 = (q_1, \sigma_1, q_1') \in \delta^a$ and $tr_2 = (q_2, \sigma_2, q_2') \in \delta^a$ with the same event $\sigma_1 = \sigma_2$, we have $A_{tr_1} = A_{tr_2}$. In this case, A_{tr} can also be written as A_σ. In the second case, sensor attacks are "dynamic," that is, $A_{tr_1} \neq A_{tr_2}$ for some $tr_1 = (q_1, \sigma_1, q_1') \in \delta^a$ and $tr_2 = (q_2, \sigma_2, q_2') \in \delta^a$ with the same event $\sigma_1 = \sigma_2$.

The results in this paper work for both static and dynamic attacks, provided the plant model G is adjusted in the case of dynamic attacks. Namely, different implementations are needed for static attacks and dynamic attacks as described below.

For static attacks, the implementation is simple: whenever an attacker sees an event $\sigma \in \Sigma^a_o$, it will replace σ with some strings in the same A_σ. In this case, the plant model G can be used as is and ALTER is defined as in Section 3.1 with the corresponding $\{A_{tr}\}$ sets.

For dynamic attacks, the implementation is more complex: when an attacker sees an event $\sigma \in \Sigma^a_o$, it has the option of choosing a different attack language to use, since this language may not be the same for each occurrence of σ. We assume that this decision depends on the string of events the attacker has observed so far.

Formally, let Σ_{ao} be the set of events observable to the attacker. It is often the case that $\Sigma_{ao} = \Sigma_o$. However, the approach proposed here works as long as $\Sigma^a_o \subseteq \Sigma_{ao}$, that is, the attacker can observe all events that it wants to attack. Let $P_{ao} : \Sigma^* \to \Sigma^*_{ao}$ be the natural projection from Σ^* to Σ^*_{ao}.

An implementation is based on a finite automaton model of the dynamic attacker[3], with event set Σ_{ao}:

$$H = (Y, \Sigma_{ao}, \zeta, y_o, Y_m).$$

We assume that all states in H are marked, that is, $Y_m = Y$. To ensure that all strings observed by the attacker are defined in H, it is required that $P_{ao}(L(G)) \subseteq L(H)$. For example, the dynamic attacker may be defined based on the observer of G with respect to P_{ao} as H. In that case, $P_{ao}(L(G)) = L(H)$.

The set of all possible transitions of H is denoted by $\zeta = \{(y, \sigma, y') : \zeta(y, \sigma) = y'\}$. The set of attackable transitions is denoted by $\zeta^a = \{(y, \sigma, y') \in \zeta : \sigma \in \Sigma_o^a\}$. For each transition $tr = (y, \sigma, y') \in \zeta^a$, we assume that the capabilities of the dynamic attacker are as follows: it can change the event σ to any string in an attack language $A_{tr}^H \subseteq \Sigma^*$. Note that, since the attacker observes Σ_{ao}, it knows which state $y \in Y$ it is in. Hence, when it observes an attackable event, it knows which attackable transition the attackable event corresponds to. Therefore, in the case of dynamic attacks modeled by H and the set $\{A_{tr}^H\}$ as just described, we can obtain an attack model consistent with the definitions of Section 3.1 if we embed H into G as follows. Take the parallel composition of G and H [16]:

$$\hat{G} = (\hat{Q}, \Sigma, \hat{\delta}, \hat{q}_o, \hat{Q}_m) = G||H = (Q \times Y, \Sigma, \delta \times \zeta, (q_o, y_o), Q \times Y),$$

where $Q \times Y$ denotes the product of sets and $\delta \times \zeta$ is defined, for $(q, y) \in Q \times Y$ and $\sigma \in \Sigma$, as

$$(\delta \times \zeta)((q, y), \sigma) = \begin{cases} (\delta(q, \sigma), \zeta(y, \sigma)) & \text{if } \sigma \in \Sigma_{ao} \\ (\delta(q, \sigma), y) & \text{otherwise} \end{cases}.$$

For a transition $\hat{tr} = (\hat{q}, \sigma, \hat{q}') = ((q, y), \sigma, (q', y'))$ with $\sigma \in \Sigma_o^a$, the corresponding attack language is given by $A_{\hat{tr}} = A_{(y,\sigma,y')}^H$.

Since $P_{ao}(L(G)) \subseteq L(H)$ implies $L(\hat{G}) = L(G)$, in the rest of the paper, we assume that, without loss of generality, G is already embedded with some H. If not, we can take $\hat{G} = G||H$, call $\hat{G}$ the new G, and work on the new G.

The ALTER model is both general and specific. It is general in the sense that it includes deletion, insertion, and replacement as special cases. It is specific in the sense that the attacks are specified by attack languages. This makes ALTER different from related work reviewed in the introduction, as attack languages are a general way of capturing two features: (i) the constraints on insertion, deletion, and replacement; and (ii) the nondeterminism of the attack.

Let us now illustrate the ALTER model using the following example.

[3] In prior work, it is usually assumed that either an attack model is known or an "all-out" attack model is used. We adopt the same viewpoint in this paper. Basically, if one is going to test the vulnerabilities of a system, then one must start with potential vulnerabilities.

Example 1 Let us consider the discrete event system G in Fig. 1 with states $Q = \{1, 2, 3, 4\}$ and events $\Sigma = \{\alpha, \beta, \gamma, \mu\}$. In the figures, symbol $\rightarrow$ denotes the initial state and double circles denote marked states.

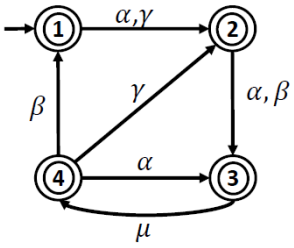

Fig. 1 Discrete event system G of Example 1.

Assume that event μ is unobservable and event α may be attacked, that is, $\Sigma_o = \{\alpha, \beta, \gamma\}$ and $\Sigma_o^a = \{\alpha\}$. The attack is dynamic and we assume that it is implemented by the automaton H in Fig. 2 with $\Sigma_{ao} = \{\alpha, \beta\}$. In H, there are two attackable transitions: $\zeta^a = \{tr_1, tr_2\}$, where $tr_1 = (A, \alpha, A)$ and $tr_2 = (B, \alpha, B)$. Let $A^H_{tr_1} = \{\alpha\}$ and $A^H_{tr_2} = \{\beta\}$, that is, the attacker will do nothing at state A and will replace α by β at state B. The corresponding automaton for $F^H_{tr_2}$ is shown in Fig. 3.

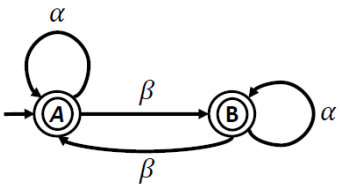

Fig. 2 The automaton H implementing the dynamic attack.

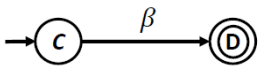

Fig. 3 Automaton F^H_{tr} of attack language A^H_{tr} for transition $tr_2 = (B, \alpha, B) \in \zeta^a$.

The synchronous product $\hat{G} = G \| H$ is shown in Fig. 4.

There are six attackable transitions in $\hat{G}$: $\hat{\delta}^a = \{\hat{tr}_1, \hat{tr}_2, \hat{tr}_3, \hat{tr}_4, \hat{tr}_5, \hat{tr}_6\}$, where $\hat{tr}_1 = ((1, A), \alpha, (2, A))$, $\hat{tr}_2 = ((2, A), \alpha, (3, A))$, $\hat{tr}_3 = ((4, A), \alpha, (3, A))$, $\hat{tr}_4 = ((1, B), \alpha, (2, B))$, $\hat{tr}_5 = ((2, B), \alpha, (3, B))$, and $\hat{tr}_6 = ((4, B), \alpha, (3, B))$.

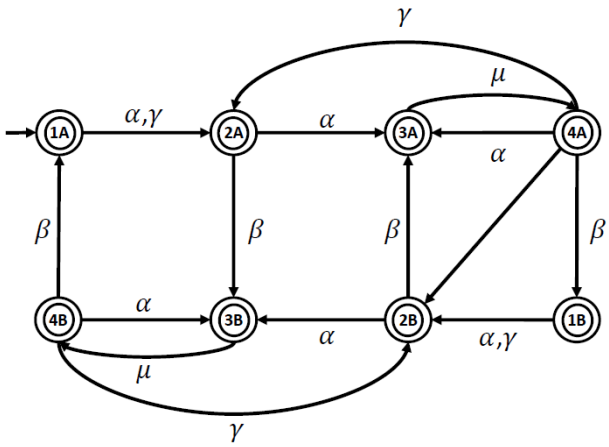

Fig. 4 The synchronous product $\hat{G} = G \| H$ of Example 1.

We have $A_{\hat{tr}_1} = A_{\hat{tr}_2} = A_{\hat{tr}_3} = A^H_{tr_1} = \{\alpha\}$ and $A_{\hat{tr}_4} = A_{\hat{tr}_5} = A_{\hat{tr}_6} = A^H_{tr_2} = \{\beta\}$. The extended automaton $\hat{G}^e$ obtained after replacing all transitions subject to attacks in $\hat{G}$ is shown in Fig. 5. Note that F_{tr_2} has only one marked state. So, we use the shortcut of merging q and q' with $q_{0,tr}$ and $q_{m,tr}$, respectively, without the ε-transitions $(q, \varepsilon, q_{0,tr})$ and $(q_{m,tr}, \varepsilon, q')$.

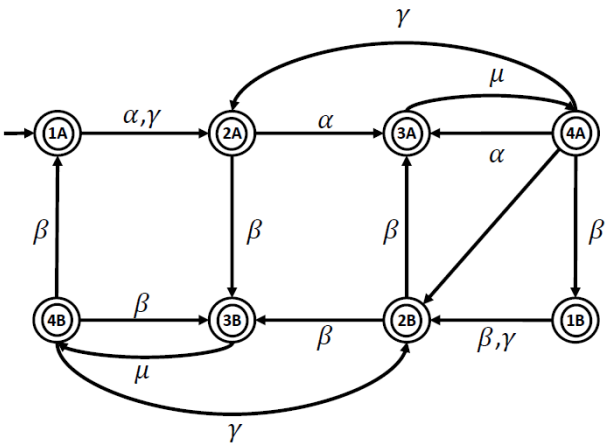

Fig. 5 The extended automaton $\hat{G}^e$

4 Power System Example

In this section, we present an example of a protection relay and a circuit breaker in a power system that will be used to illustrate the theoretical results in this paper.

Example 2 The system is shown in Fig. 6. The circuit breaker will be closed or tripped when the corresponding closing coil (CC) or tripping coil (TC) is energized. There are auxiliary contacts that assist with the functions of the circuit breaker. The auxiliary normally-open (NO) contact a is closed when the main circuit breaker is closed; otherwise, it is open when the circuit breaker is open. The auxiliary normally-closed (NC) contact b is closed when the main circuit breaker is open; otherwise, it is open when the circuit breaker is closed. There can be additional TC and CC contacts that can be controlled remotely or locally via manual operations.

The system works as follows. If there is a downed power line or other accident that has occurred in the power system, causing an over current event on the power line, as shown in Fig. 6, then the protection relay (PR) will be triggered. When the PR is triggered, the corresponding PR contact is closed. Then, the TC of the circuit breaker is energized to trip (open) the circuit breaker to cut the faulty power line from the power system as the auxiliary NO contact a is closed when the main circuit breaker is closed. After the circuit breaker is open for a short period of time, it will automatically try to reclose through a pre-defined reclosing command/procedure (RC). It is noted that when the circuit breaker is open, the contact a is open, and the tripping coil is cut from the control power. At the same time, the auxiliary normally-closed (NC) contact b is closed when the main circuit breaker is open. Therefore, when the RC contact is closed by the reclosing command, the CC is energized to close the circuit breaker. If the over current disappears after the reclosure, then the fault is temporal, and the line returns to normal operation. If the over current stays after the reclosure, then the fault is permanent and the PR will be tripped again, and the circuit breaker will return to and remain open until the repair is made to the power line. It is worth noting that the diagram shown in Fig. 6 is a much-simplified version of a real system. Nevertheless, the main function of the system can be modeled by the automaton G shown in Fig. 7. In the automaton G, the states are $Q = \{1, 2, 3, 4, 5, 6\}$ and the events are

$O.C.$ - Over current,

$Z.C.$ - Zero current,

PR_T - Protection relay tripped,

PR_F - Protection relay failed,

CB_T - Circuit breaker tripped,

CB_F - Circuit breaker failed, and

R - Circuit breaker reclosed.

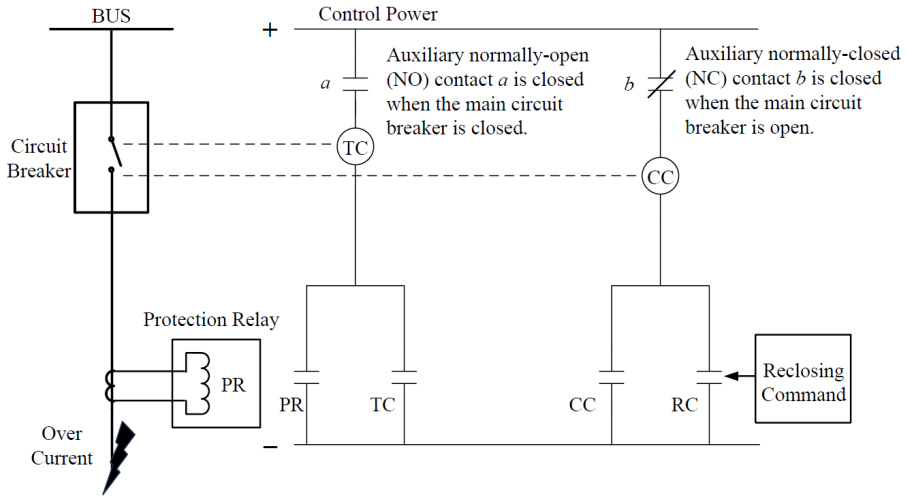

Fig. 6 Protection relay and circuit breaker

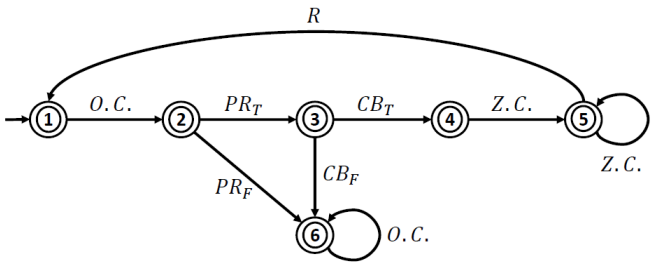

Fig. 7 Automaton G for protection relay and circuit breaker

We assume that only events $O.C.$ and $Z.C.$ are observable (that is, the current can be measured). Hence, $\Sigma_o = \{O.C., Z.C.\}$. We further assume that $\Sigma_o^a = \Sigma_{ao} = \Sigma_o$.

We would like to diagnose faults in the system. Clearly, there are two fault events: PR_F and CB_F. Hence, $\Sigma_f = \{PR_F, CB_F\}$. It can be checked that without sensor attacks, G is diagnosable with respect to P [1,16]. Intuitively, this is because, after the occurrence of either PR_F or CB_F, a diagnoser will see event $O.C.$, while under normal operation, the diagnoser will see event $Z.C.$.

Let us now suppose that an attacker can change the transition $tr = (6, O.C., 6)$ to $(6, Z.C., 6)$, that is, $A_{tr} = \{Z.C.\}$ with F_{tr} shown in Fig. 8. Note that G is already embedded with the attacker model H shown in Fig. 9 because G is isomorphic to $G||H$.

For the automaton G shown in Fig. 7, we can construct the extended automaton G^e as shown in Fig. 10.

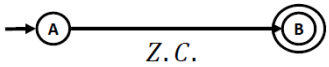

Fig. 8 Automaton F_{tr} of attack language A_{tr} for transition $tr = (6, O.C., 6)$

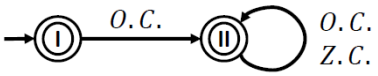

Fig. 9 Automaton H implementing the dynamic attack

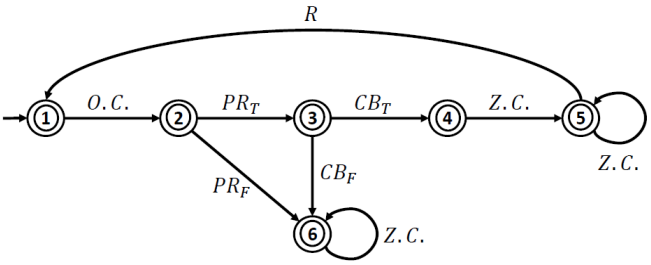

Fig. 10 The extended automaton G^e

5 Diagnosability under Sensor Attacks

Under sensor attacks, after the occurrence of $s \in L(G)$, a diagnoser observes one of the strings $s' \in \Phi^a(s)$. Hence, we extend diagnosability to CA-diagnosability as follows.

Definition 2 A DES G is CA-diagnosable with respect to Φ^a if

$$\begin{aligned}&(\exists n \in \mathcal{N})(\forall s \in \Psi(L(G)))(\forall u \in L(G)/s)\\&|u| \geq n \Rightarrow (\forall v \in (\Phi^a)^{-1}(\Phi^a(su)) \cap L(G))\Sigma_f \in v,\end{aligned} \tag{7}$$

where $(\Phi^a)^{-1}$ is the inverse mapping of Φ^a, that is, $(\Phi^a)^{-1}(v') = \{v \in L(G) : v' \in \Phi^a(v)\}$.

We first show that CA-diagnosability of G is equivalent to (conventional) diagnosability of G^e under the following assumptions:

A3. An attacker cannot delete or insert fault events, that is,

$$(\forall v \in L(G))(\forall v' \in \Theta^a(v))\Sigma_f \notin v \Leftrightarrow \Sigma_f \notin v'. \tag{8}$$

A4. An attacker can only delete/insert a bounded number of events, that is,

$$\begin{aligned}&(\forall s \in L(G))(\forall u \in L(G)/s)(\forall s' \in \Theta^a(s))\\&(\forall u' \in \Theta^a(L(G))/s')(s'u' \in \Theta^a(su)\\&\Rightarrow |(|u| - |u'|)| \leq d)\end{aligned} \tag{9}$$

for some integer $d > 0$.

Note that Assumption A3 will be satisfied if

$$(\forall tr \in \delta^a) A_{tr} \subseteq (\Sigma - \Sigma_f)^*.$$

Note also that Assumption A4 ensures that Assumption A2 is true under attacks. Furthermore, Assumption A4 is needed in several proofs.

Let us now prove the following theorem.

Theorem 1 *Under Assumptions A3 and A4, G is CA-diagnosable with respect to Φ^a if and only if G^e is diagnosable with respect to P, that is,*

$$(\exists n \in \mathcal{N})(\forall s \in \Psi(L(G)))(\forall u \in L(G)/s)$$
$$|u| \geq n \Rightarrow (\forall v \in (\Phi^a)^{-1}(\Phi^a(su)) \cap L(G))\Sigma_f \in v$$

if and only if

$$(\exists n \in \mathcal{N})(\forall s \in \Psi(L(G^e)))(\forall u \in L(G^e)/s)$$
$$|u| \geq n \Rightarrow (\forall v \in P^{-1}(P(su)) \cap L(G^e))\Sigma_f \in v$$

Proof

The proof is in the Appendix. ∎

Since CA-diagnosability of G is equivalent to diagnosability of G^e, all techniques developed for diagnosability can be used to solve problems in CA-diagnosability.

Example 3 Let us continue with the example of a protection relay and a circuit breaker in a power system discussed in the previous section. It can be checked that Assumptions A1 - A4 are all satisfied.

Using standard methods [1, 16], we can check that G^e is not diagnosable with respect to P. Intuitively, this is because the attacker changes $O.C.$ to $Z.C.$ So, after the occurrence of either PR_F or CB_F, a diagnoser will see event $Z.C.$, which is same as it will see under normal operation.

By Theorem 1, G is not CA-diagnosable with respect to Φ^a. Hence, the sensor attack makes a diagnosable system not CA-diagnosable.

If G is CA-diagnosable with respect to Φ^a, then a new diagnoser, called CA-diagnoser, can be designed for G to diagnose faults based on G^e using standard methods [1, 16]. Let us illustrate this by the following modified power system example.

Example 4 Consider again the power system in Example 2, but with a different attacker. The attacker inserts $Z.C.$ after $O.C.$, that is, for $tr = (6, O.C., 6)$, $A'_{tr} = \{O.C.\ Z.C.\}$ with F'_{tr} shown in Fig. 11.

Let us check if G under the new attacker is CA-diagnosable or not by constructing a CA-diagnoser as follows.

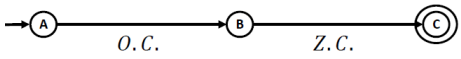

Fig. 11 The new automaton F'_{tr} of attack language A'_{tr} for transition $tr = (6, O.C., 6)$

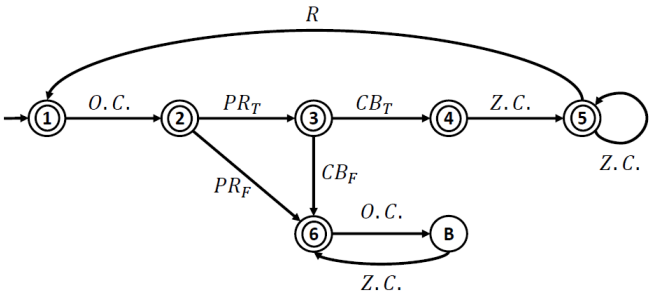

Fig. 12 The new extended automaton G'^e

Step 1: Construct the new extended automaton G'^e as shown in Fig. 12. Note that, since F'_{tr} has only one marked state, we eliminate ε-transitions in G'^e.

Step 2: Construct the label automaton G_{label} as shown in Fig. 13. Note that the fault events are PR_F and CB_F. In the automaton G_{label}, N denotes "normal" and F denotes "faulty".

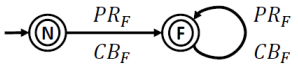

Fig. 13 Label automaton G_{label} for building CA-diagnoser

Step 3: Take parallel composition $\tilde{G}^e = G'^e || G_{label}$ as shown in Fig. 14. Note that $\tilde{G}^e$ and G'^e are isomorphic. However, states in $\tilde{G}^e$ are labeled with N or F.

Step 4: Replace unobservable events in $\tilde{G}^e$ by ε to obtain nondeterministic automaton $\tilde{G}^e_\varepsilon$ as shown in Fig. 15.

Step 5: Convert nondeterministic automaton $\tilde{G}^e_\varepsilon$ to the equivalent deterministic automaton (also called observer) $\tilde{G}^e_{obs}$ as shown in Fig. 16 (see [16] for details). $\tilde{G}^e_{obs}$ is the diagnoser for G'^e. We call $\tilde{G}^e_{obs}$ the CA-diagnoser for G.

Since there is no indeterminate cycle in $\tilde{G}^e_{obs}$, G'^e is diagnosable with respect to P. By Theorem 1, G is CA-diagnosable with respect to Φ'^a.

Therefore, CA-diagnoser $\tilde{G}^e_{obs}$ can be used to diagnose the faults: When $\tilde{G}^e_{obs}$ enters states BF or 6F, we know that a fault has occurred.

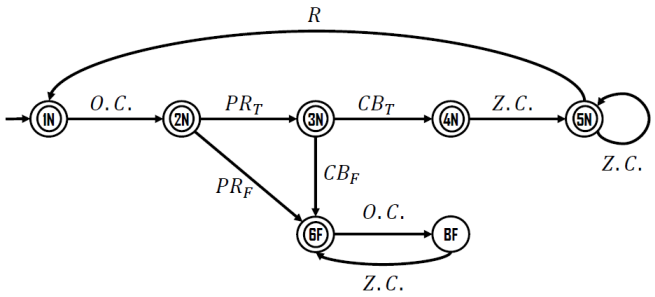

Fig. 14 Parallel composition $\tilde{G}^e = G'^e || G_{label}$

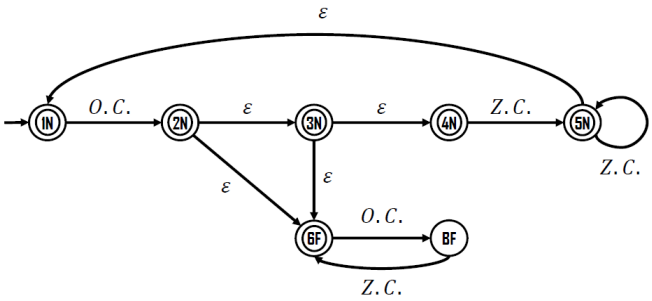

Fig. 15 Nondeterministic automaton $\tilde{G}^e_\varepsilon$

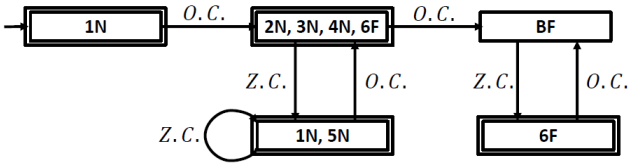

Fig. 16 CA-diagnoser $\tilde{G}^e_{obs}$

6 Sufficient Conditions for CA-Diagnosability

In this section, we investigate the following question: If a system is diagnosable without sensor attacks, under what types of sensor attacks, is it still diagnosable? In other words, we would like to find some (sufficient) conditions on sensor attacks under which diagnosability implies CA-diagnosability. Let us first prove the following theorem.

Theorem 2 *The fact that G is diagnosable with respect to P implies that G is CA-diagnosable with respect to Φ^a if*

$$(\forall s \in L(G))$$
$$(\Phi^a)^{-1}(\Phi^a(s)) \cap L(G) \subseteq P^{-1}(P(s)) \cap L(G).$$

Proof

Clearly, the following is true:

$$\begin{aligned}
&(\forall s \in L(G)) \\
&(\Phi^a)^{-1}(\Phi^a(s)) \cap L(G) \subseteq P^{-1}(P(s)) \cap L(G) \\
\Rightarrow &(\forall s \in \Psi(L(G)))(\forall u \in L(G)/s) \\
&(\Phi^a)^{-1}(\Phi^a(su)) \cap L(G) \subseteq P^{-1}(P(su)) \cap L(G) \\
\Rightarrow &(\forall s \in \Psi(L(G)))(\forall u \in L(G)/s) \\
&(\forall v \in P^{-1}(P(su)) \cap L(G))\Sigma_f \in v \\
&\Rightarrow (\forall v \in (\Phi^a)^{-1}(\Phi^a(su)) \cap L(G))\Sigma_f \in v
\end{aligned}$$

Therefore,

$$\begin{aligned}
&(\exists n \in \mathcal{N})(\forall s \in \Psi(L(G)))(\forall u \in L(G)/s) \\
&|u| \geq n \Rightarrow (\forall v \in P^{-1}(P(su)) \cap L(G^e))\Sigma_f \in v \\
\Rightarrow &(\exists n \in \mathcal{N})(\forall s \in \Psi(L(G)))(\forall u \in L(G)/s) \\
&|u| \geq n \Rightarrow (\forall v \in (\Phi^a)^{-1}(\Phi^a(su)) \cap L(G))\Sigma_f \in v
\end{aligned}$$

■

Let us model the sensor attacks described above by a mapping

$$\Omega : \delta^a \to 2^{\Sigma^*}$$

such that, for $tr = (q, \sigma, q') \in \delta^a$, $\Omega(tr) = A_{tr}$. Extend Ω to

$$\Omega : \delta \to 2^{\Sigma^*}$$

as follows. For $tr = (q, \sigma, q') \in \delta$,

$$\Omega(tr) = \begin{cases} \Omega(tr) & \text{if } tr \in \delta^a \\ \{\sigma\} & \text{otherwise} \end{cases}.$$

In other words, if $tr = (q, \sigma, q') \notin \delta^a$, then simply let $A_{tr} = \{\sigma\}$ (replacing σ by σ).

Theorem 3 *The fact that G is diagnosable with respect to P implies that G is CA-diagnosable with respect to Φ^a if*

$$\begin{aligned}
&(\forall tr = (q_1, \sigma, q_2) \in \delta)(\forall tr' = (q_1', \sigma', q_2') \in \delta) \\
&\sigma \neq \sigma' \Rightarrow P(\Omega(tr)) \cap P(\Omega(tr')) = \emptyset.
\end{aligned} \tag{10}$$

Proof

Let us first prove by contradiction that if Equation (10) is true, then, for all $s \in L(G)$,

$$(\Phi^a)^{-1}(\Phi^a(s)) \cap L(G) \subseteq P^{-1}(P(s)) \cap L(G).$$

Suppose $(\Phi^a)^{-1}(\Phi^a(s)) \cap L(G) \not\subseteq P^{-1}(P(s)) \cap L(G)$. Then

$$\begin{aligned}
&(\Phi^a)^{-1}(\Phi^a(s)) \cap L(G) \not\subseteq P^{-1}(P(s)) \cap L(G)\\
\Rightarrow&(\exists s' \in L(G))s' \in (\Phi^a)^{-1}(\Phi^a(s)) \wedge s' \notin P^{-1}(P(s))\\
\Rightarrow&(\exists s' \in L(G))\Phi^a(s) \cap \Phi^a(s') \neq \emptyset \wedge P(s') \neq P(s)\\
\Rightarrow&(\exists s' \in L(G))(\exists w)w \in \Phi^a(s) \wedge w \in \Phi^a(s')\\
&\wedge P(s') \neq P(s)\\
\Rightarrow&(\exists s' \in L(G))(\exists w)w \in P(\Theta^a(s)) \wedge w \in P(\Theta^a(s'))\\
&\wedge P(s') \neq P(s)\\
\Rightarrow&(\exists s' \in L(G))(\exists w)(\exists v \in \Theta^a(s))(\exists v' \in \Theta^a(s'))\\
&w = P(v) \wedge w = P(v') \wedge P(s') \neq P(s)\\
\Rightarrow&(\exists s' \in L(G))(\exists v \in \Theta^a(s))(\exists v' \in \Theta^a(s'))\\
&P(v) = P(v') \wedge P(s') \neq P(s).
\end{aligned}$$

$P(s') \neq P(s)$ means that s and s' are different for at least one observable event. Without loss of generality, assume that s and s' are different for exactly one observable event, that is, $s = s_1\sigma s_2$ and $s' = s'_1\sigma' s'_2$ such that $P(s_1) = P(s'_1) \wedge P(s_2) = P(s'_2) \wedge \sigma \neq \sigma' \wedge \sigma, \sigma' \in \Sigma_o$.

Denote the corresponding transitions of σ and σ' by tr and tr' respectively. Then,

$$\begin{aligned}
&(\exists v \in \Theta^a(s))(\exists v' \in \Theta^a(s'))P(v) = P(v')\\
\Rightarrow&(\exists u \in \Omega(tr))(\exists u' \in \Omega(tr'))P(u) = P(u')\\
&\text{(because } tr \text{ and } tr' \text{ must be replaced}\\
&\text{by strings with the same projection)}\\
\Rightarrow&P(\Omega(tr)) \cap P(\Omega(tr')) \neq \emptyset,
\end{aligned}$$

which contradicts Equation (10).

Since

$$\begin{aligned}
&(\forall s \in L(G))\\
&(\Phi^a)^{-1}(\Phi^a(s)) \cap L(G) \subseteq P^{-1}(P(s)) \cap L(G).
\end{aligned}$$

by Theorem 2, the fact that G is diagnosable with respect to P implies that G is CA-diagnosable with respect to Φ^a.

■

Theorem 3 says that, given a diagnosable G, for G to be CA-diagnosable, the attacker cannot replace two different observable events with two strings having the same projection.

If the sufficient condition in Theorems 2 or 3 is satisfied, then we can design a CA-diagnoser to diagnose faults in G. In general, the CA-diagnoser may be different from the original diagnoser.

Note that Equation (10) is not satisfied in the power system example when $A_{tr} = \{Z.C.\}$, because

$$(\exists tr = (6, O.C, 6) \in \delta)(\exists tr' = (5, Z.C., 5) \in \delta)$$
$$O.C. \neq Z.C. \wedge P(\Omega(tr)) \cap P(\Omega(tr')) = \{Z.C.\} \neq \emptyset.$$

On the other hand, Equation (10) is satisfied in the power system example when $A'_{tr} = \{O.C.\ Z.C.\}$. In particular, for $tr = (6, O.C, 6)$ and $tr' = (5, Z.C., 5)$,

$$P(\Omega(tr)) \cap P(\Omega(tr')) = \{Z.C.\} \cap \{O.C.\ Z.C.\} = \emptyset.$$

7 From Attacker to Helper

In this section, we revert the role of the attacker from malicious to benevolent. In other words, we consider the situation where the attacker is actually a helper, whose goal is to help the diagnoser to diagnose faults in the sense that while G is not diagnosable, it is CA-diagnosable. Intuitively, it would seem that if the helper observes the same set of events as the diagnoser, it cannot help in event diagnosis. On the other hand, if the helper has access to more observable events, then it might be in a position to resolve lack of diagnosability. Let us first show that this is possible using the following example.

Example 5 Let us consider the system modeled by G shown in Fig. 17. The event set is $\Sigma = \{f, u, \alpha, \beta, \gamma\}$, where f is the fault event. We assume that u is not observable and β is not observable to the diagnoser, but is observable to the helper. We further assume that α can be edited by the helper. Hence, $\Sigma_o = \{\alpha, \gamma\}$, $\Sigma_o^a = \{\alpha\}$, and $\Sigma_{ao} = \{\alpha, \beta, \gamma\}$.

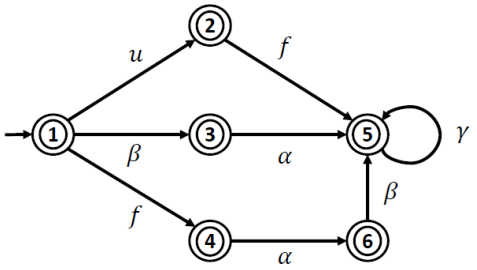

Fig. 17 Automaton G of the system in Example 5

It can be checked that G is not diagnosable with respect to P. Intuitively, this is because the two strings $\beta\alpha\gamma^*$ and $f\alpha\beta\gamma^*$ have the same projection $\alpha\gamma^*$ and hence cannot be distinguished by the diagnoser.

Suppose that the helper can change the transition $tr = (4, \alpha, 6)$ to $(4, \gamma, 6)$, that is, $A_{tr} = \{\gamma\}$ with F_{tr} shown in Fig. 18. Note that the helper can do

so because event β is observable to the helper (but not observable to the diagnoser). Hence, the helper can distinguish transition $(4, \alpha, 6)$ from $(3, \alpha, 5)$.

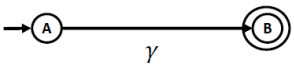

Fig. 18 Automaton F_{tr} of Example 5

The resulting extended automaton G^e is shown in Fig. 19. It can be checked that G^e is diagnosable with respect to P and hence G is CA-diagnosable with respect to Φ^a. Intuitively, this is because by replacing $(4, \alpha, 6)$ with $(4, \gamma, 6)$, string $f\alpha\beta\gamma^*$ become $f\gamma\beta\gamma^*$. Since $P(f\gamma\beta\gamma^*) = \gamma\gamma^*$, while $P(\beta\alpha\gamma^*) = \alpha\gamma^*$, the two strings $f\gamma\beta\gamma^*$ and $\beta\alpha\gamma^*$ become distinguishable to the diagnoser.

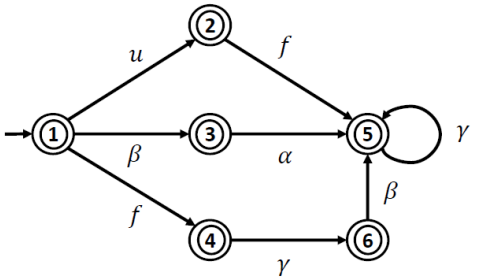

Fig. 19 Extended automaton G^e of Example 5

In the above example, the reason that the helper can help the diagnoser to diagnose the fault is that the helper can observe β, while the diagnoser cannot. Note that in Example 5 the helper is not inserting event β, which is unobservable to the diagnoser, but instead it replaces one occurrence of α by γ. This type of solution might be desirable in instances where communication of certain events (here, β) must be avoided for various reasons (e.g., security). The helper allows the diagnoser to work properly while keeping β unobservable to the diagnoser. In the following theorem, we prove that this is necessary, that is, if the helper observes less than or the same as the diagnoser, then it cannot help the diagnoser.

Theorem 4 *If $\Sigma_{ao} \subseteq \Sigma_o$ and G is not diagnosable with respect to P, then G is not CA-diagnosable with respect to Φ^a for any Φ^a satisfying Assumptions A3 and A4.*

Proof

Suppose $\Sigma_{ao} \subseteq \Sigma_o$ and G is not diagnosable with respect to P, that is,

$$\neg(\exists n \in \mathcal{N})(\forall s \in \Psi(L(G)))(\forall u \in L(G)/s)$$

$$
\begin{aligned}
&|u| \geq n \Rightarrow (\forall v \in P^{-1}(P(su)) \cap L(G))\Sigma_f \in v \\
\Leftrightarrow &(\forall n \in \mathcal{N})(\exists s \in \Psi(L(G)))(\exists u \in L(G)/s) \\
&|u| \geq n \wedge (\exists v \in L(G))P(v) = P(su) \wedge \Sigma_f \notin v
\end{aligned}
$$

We want to prove that for any Φ^a satisfying Assumptions A3 and A4, G is not CA-diagnosable with respect to Φ^a. Let G^e be the extended automaton corresponding to Φ^a. By Theorem 1, we only need to prove that G^e is not diagnosable with respect to P, that is,

$$
\begin{aligned}
&\neg(\exists n' \in \mathcal{N})(\forall s' \in \Psi(L(G^e)))(\forall u' \in L(G^e)/s') \\
&|u'| \geq n' \Rightarrow (\forall v' \in P^{-1}(P(s'u')) \cap L(G^e))\Sigma_f \in v' \\
\Leftrightarrow &(\forall n' \in \mathcal{N})(\exists s' \in \Psi(L(G^e)))(\exists u' \in L(G^e)/s') \\
&|u'| \geq n' \wedge (\exists v' \in L(G^e))P(v') = P(s'u') \wedge \Sigma_f \notin v'.
\end{aligned}
$$

Let us first prove

$$
\begin{aligned}
&(\exists s \in \Psi(L(G)))(\exists u \in L(G)/s) \\
&(\exists v \in L(G))P(v) = P(su) \wedge \Sigma_f \notin v
\end{aligned} \tag{11}
$$

implies

$$
\begin{aligned}
&(\exists s' \in \Psi(L(G^e)))(\exists u' \in L(G^e)/s') \\
&(\exists v' \in L(G^e))P(v') = P(s'u') \wedge \Sigma_f \notin v'.
\end{aligned} \tag{12}
$$

For s, u, v in (11), consider $s'u' \in \Theta^a(su)$ and $v' \in \Theta^a(v)$. Since $s \in \Psi(L(G)) \Rightarrow \Sigma_f \in su$. By Assumption A3, $\Sigma_f \in s'u'$. Let $s'u' \in \Theta^a(su)$ be such that $s' \in \Psi(\Theta^a(L(G))) \wedge s' \in \Theta^a(s)$.

Since $P(v) = P(su)$ and $\Sigma_{ao} \subseteq \Sigma_o$, we have $P_{ao}(v) = P_{ao}(su)$. Hence, any transition tr along $P_{ao}(v)$ $(= P_{ao}(su))$ is replaced by the same language A_{tr}. Let us pick the same string in A_{tr} for v' as for $s'u'$. Then $P(v') = P(s'u')$.

Because

$$
\begin{aligned}
&su \in L(G) \Rightarrow s'u' \in \Theta^a(L(G)) \Rightarrow u' \in \Theta^a(L(G))/s' \\
&v \in L(G) \Rightarrow v' \in \Theta^a(L(G)) \\
&\Sigma_f \notin v \Rightarrow \Sigma_f \notin v'(\text{ by Assumption A3}),
\end{aligned}
$$

we have

$$
\begin{aligned}
&(\exists s' \in \Psi(\Theta^a(L(G))))(\exists u' \in \Theta^a(L(G))/s') \\
&(\exists v' \in \Theta^a(L(G)))P(v') = P(s'u') \wedge \Sigma_f \notin v'.
\end{aligned}
$$

that is, (12) is true.

Let us now prove

$$
\begin{aligned}
&(\forall n \in \mathcal{N})(\exists s \in \Psi(L(G)))(\exists u \in L(G)/s) \\
&|u| \geq n \wedge (\exists v \in L(G))P(v) = P(su) \wedge \Sigma_f \notin v
\end{aligned} \tag{13}
$$

implies

$$\begin{aligned}&(\forall n' \in \mathcal{N})(\exists s' \in \Psi(L(G^e)))(\exists u' \in L(G^e)/s')\\&|u'| \geq n' \wedge (\exists v' \in L(G^e))P(v') = P(s'u') \wedge \Sigma_f \notin v'.\end{aligned} \tag{14}$$

Suppose (13) is true. Then, for any $n' \in \mathcal{N}$, let $n = n' + d$, where d is given in Assumption A4.

Because (11) $\Rightarrow$ (12), we have

$$\begin{aligned}&(\exists s \in \Psi(L(G)))(\exists u \in L(G)/s)\\&|u| \geq n \wedge (\exists v \in L(G))P(v) = P(su) \wedge \Sigma_f \notin v\\\Rightarrow&(\exists s' \in \Psi(\Theta^a(L(G))))(\exists u' \in \Theta^a(L(G))/s')\\&(\exists v' \in \Theta^a(L(G)))P(v') = P(s'u') \wedge \Sigma_f \notin v'.\end{aligned}$$

Furthermore, by the proof of (11) $\Rightarrow$ (12),

$$\begin{aligned}&s \in L(G) \wedge u \in L(G)/s \wedge s' \in \Theta^a(s)\\&\wedge u' \in \Theta^a(L(G))/s' \wedge s'u' \in \Theta^a(su)\end{aligned}$$

By Assumption A4, we have $|(|u| - |u'|)| \leq d$. Hence,

$$\begin{aligned}&|u| \geq n \wedge |(|u| - |u'|)| \leq d\\\Rightarrow&|u| \geq n \wedge |u| - |u'| \leq d\\\Rightarrow&|u| \geq n \wedge |u'| \geq |u| - d\\\Rightarrow&|u'| \geq n - d = n'.\end{aligned}$$

Therefore, (14) is true.

■

8 Sensor Attack Detection

Diagnosability theory can also be used to detect sensor attacks. We show how to do this in this section.

One obvious method to detect sensor attacks is to check if the observed string is in $P(L(G))$ or not. If a string $w \notin P(L(G))$ is observed, then a sensor attack must have occurred. To avoid being detected, an attacker may want to ensure that the attacks are "stealthy" (or covert) in the sense that the observed language under sensor attacks is contained in $P(L(G))$, that is, $\Phi^a(L(G)) \subseteq P(L(G))$ or equivalently $\overline{\Phi^a(L(G))} \subseteq P(L(G))$ (since $P(L(G))$ is prefix-closed). Stealthy attacks on sensors or actuators have been investigated in the literature; see, e.g., [6, 22, 26–28].

Another method to detect sensor attacks is to translate the sensor attack detection problem into a diagnosis problem. Other works have considered the same approach for different attack models. In the case of ALTER, we propose to proceed as follows. We insert an artificial (unobservable) fault event ρ after

an event is altered by an attacker, as described below. Then, detecting a sensor attack is equivalent to diagnosing the fault event ρ.

Formally, let the system under sensor attacks be G. Assume that G has no fault events, because we want to focus on the sensor attack detection problem. For any transition $tr = (q, \sigma, q') \in \delta^a$, we modify its corresponding language A_{tr} by adding ρ after σ is altered by an attacker as follows.

$$\overline{A}_{tr} = (A_{tr} - \{\sigma\})\{\rho\} \cup (\{\sigma\} \cap A_{tr}). \tag{15}$$

We construct the extended automaton for $\bar{A}_{tr}$. Since $\rho \notin \Sigma$, the resulting extended automaton has one more unobservable event. To distinguish it from the extended automaton in previous sections, let us denote it by

$$\bar{G}^e = (\bar{Q}^e, \Sigma \cup \{\rho\}, \bar{\delta}^e, q_o).$$

Since the artificial fault event ρ is inserted whenever an event $\sigma \in \Sigma_o^a$ is altered by an attacker, all attacks can be detected within finite steps after the attacks if and only if $\bar{G}^e$ is diagnosable with respect to P and ρ.

To investigate the relationship between stealthiness and diagnosability, we make the following assumption:

A5. An attacker can always choose not to alter an event, that is,

$$(\forall tr = (q, \sigma, q') \in \delta^a)\sigma \in A_{tr}. \tag{16}$$

Note that Assumption A5 implies that $L(G) \subseteq L(\bar{G}^e)$. Note further that, as to be shown in Example 6, if Assumption A5 is not true, then the natural and intuitive result that the attacker cannot be detected if the attacker is stealthy is not true.

The following theorem shows that, under Assumption A5, if an attacker is stealthy, then none of its attacks can be detected.

Theorem 5 *If $\Phi^a(L(G)) \subseteq P(L(G))$ and Φ^a satisfies Assumption A5, then no attack (event ρ) in the corresponding $\bar{G}^e$ can be detected, that is,*

$$\begin{aligned} &(\forall s\rho \in L(\bar{G}^e))(\forall n \in \mathcal{N})(\exists u \in L(\bar{G}^e)/s\rho) \\ &|u| \geq n \wedge (\exists v \in L(\bar{G}^e))P(v) = P(s\rho u) \wedge \rho \notin v. \end{aligned} \tag{17}$$

Proof

Suppose $\Phi^a(L(G)) \subseteq P(L(G))$ and Φ^a satisfies Assumption A5. We prove that Equation (17) is true as follows.

For any $s\rho \in L(\bar{G}^e)$ and $n \in \mathcal{N}$, let u be a string in $L_m(\bar{G}^e)/s\rho$ such that $|u| \geq n$. By Assumption A1 (G and hence $\bar{G}^e$ is live), such a u exists.

Since $\Phi^a(L(G)) \subseteq P(L(G))$ and $u \in L_m(\bar{G}^e)/s\rho$, we have

$$\begin{aligned} &s\rho u \in L_m(\bar{G}^e) \\ \Rightarrow &s\rho u \in \Theta^a(L(G)) \\ &\text{(because } L_m(G^e) = \Theta^a(L(G))) \end{aligned}$$

$$
\begin{aligned}
&\Rightarrow P(s\rho u) \in P(\Theta^a(L(G))) = \Phi^a(L(G)) \subseteq P(L(G)) \\
&\Rightarrow (\exists v \in L(G))P(v) = P(s\rho u) \\
&\Rightarrow (\exists v \in L(\bar{G}^e))P(v) = P(s\rho u) \\
&\quad \text{(by Assumption A5, } L(G) \subseteq L(\bar{G}^e)).
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
&(\forall s\rho \in L(\bar{G}^e))(\forall n \in \mathcal{N})(\exists u \in L_m(\bar{G}^e)/s\rho) \\
&|u| \geq n \wedge (\exists v \in L(\bar{G}^e))P(v) = P(s\rho u) \wedge \rho \notin v \\
\Rightarrow &(\forall s\rho \in L(\bar{G}^e))(\forall n \in \mathcal{N})(\exists u \in L(\bar{G}^e)/s\rho) \\
&|u| \geq n \wedge (\exists v \in L(\bar{G}^e))P(v) = P(s\rho u) \wedge \rho \notin v \\
&(\text{since } L_m(\bar{G}^e) \subseteq L(\bar{G}^e)).
\end{aligned}
$$

This completes the proof.

■

The following theorem shows that if an attacker is stealthy, then $\bar{G}^e$ is not diagnosable with respect to P and ρ.

Theorem 6 *If $\Phi^a(L(G)) \subseteq P(L(G))$ and Φ^a satisfies Assumption A5, then the corresponding $\bar{G}^e$ is not diagnosable with respect to P and ρ.*

Proof

Suppose $\Phi^a(L(G)) \subseteq P(L(G))$ and Φ^a satisfies Assumption A5. Since the logic implication $(\forall x)A(x) \Rightarrow (\exists x)A(x)$ is always true, by Theorem 5,

$$
\begin{aligned}
&(\forall s\rho \in L(\bar{G}^e))(\forall n \in \mathcal{N})(\exists u \in L(\bar{G}^e)/s\rho) \\
&|u| \geq n \wedge (\exists v \in L(\bar{G}^e))P(v) = P(s\rho u) \wedge \rho \notin v \\
\Rightarrow &(\forall n \in \mathcal{N})(\forall s\rho \in L(\bar{G}^e))(\exists u \in L(\bar{G}^e)/s\rho) \\
&|u| \geq n \wedge (\exists v \in L(\bar{G}^e))P(v) = P(s\rho u) \wedge \rho \notin v \\
\Rightarrow &(\forall n \in \mathcal{N})(\exists s\rho \in L(\bar{G}^e))(\exists u \in L(\bar{G}^e)/s\rho) \\
&|u| \geq n \wedge (\exists v \in L(\bar{G}^e))P(v) = P(s\rho u) \wedge \rho \notin v \\
\Rightarrow &\neg(\exists n \in \mathcal{N})(\forall s\rho \in L(\bar{G}^e))(\forall u \in L(\bar{G}^e)/s\rho) \\
&|u| \geq n \Rightarrow (\forall v \in P^{-1}(P(su)) \cap L(\bar{G}^e))\rho \notin v.
\end{aligned}
$$

Therefore, $\bar{G}^e$ is not diagnosable with respect to P and ρ.

■

From Theorem 6 and its proof, we know that Equation (17) is stronger than non-diagnosability. Intuitively, this is because Equation (17) requires that all attackers are not detectable, while non-diagnosability only requires that some attackers are not detectable.

The following example shows that if Assumption A5 is not satisfied, then the result of Theorem 3 (and hence the result of Theorem 2) is not true.

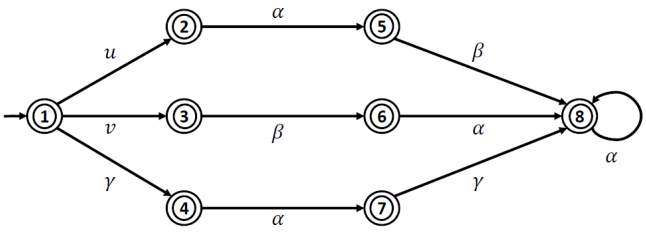

Fig. 20 Automaton G of the system in Example 6

Example 6 Let us consider the system modeled by G shown in Fig. 20. The event set is $\Sigma = \{u, v, \alpha, \beta, \gamma\}$. We assume that α and γ are observable to both the diagnoser/attack detector and the attacker, that is $\Sigma_o = \Sigma_{ao} = \{\alpha, \gamma\}$. We further assume that α is attackable, that is, $\Sigma_o^a = \{\alpha\}$.

Suppose that the attacker can change the transitions $(2, \alpha, 5)$ and $(6, \alpha, 8)$ to $(2, \alpha\alpha, 5)$ and $(6, \alpha\alpha, 8)$, respectively, that is, the attacker can insert an extra α in transitions $(2, \alpha, 5)$ and $(6, \alpha, 8)$. Note that the attacker can do so because, by observing γ, the attacker can distinguish transitions $(2, \alpha, 5)$ and $(6, \alpha, 8)$ from transition $(4, \alpha, 7)$.

We modify the corresponding language $A_{tr} = \{\alpha\alpha\}$ by adding ρ afterwards, that is, $\bar{A}_{tr} = \{\alpha\alpha\rho\}$. The corresponding $\bar{F}_{tr}$ is shown in Fig. 21.

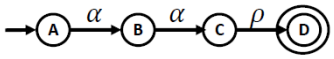

Fig. 21 Automaton $\bar{F}_{tr}$ of Example 6

The resulting extended automaton $\bar{G}^e$ is shown in Fig. 22. It is not difficult to see that $\overline{\Phi^a(L(G))} = P(L(G)) = \alpha^* + \gamma\alpha\gamma\alpha^*$. Hence, $\Phi^a(L(G)) \subseteq P(L(G))$ and the attacks are stealthy.

On the other hand, it can be checked that $\bar{G}^e$ is diagnosable with respect to P and ρ. In fact, all attacks can be detected. Intuitively, this is because if the diagnoser sees α before seeing γ, then an attack has occurred. Note that Assumption A5 is not satisfied, because $\alpha \notin A_{tr} = \{\alpha\alpha\}$.

9 Conclusion

We have studied the diagnosability properties of discrete event systems when the communication channel from the sensors to the diagnoser is compromised by sensor deception attacks in the context of a general attack model. This has led to the formulation of the new notion of CA-diagnosability, which parallels the notions of CA-controllability and CA-observability introduced in prior works pertaining to supervisory control under attack. A testing procedure for

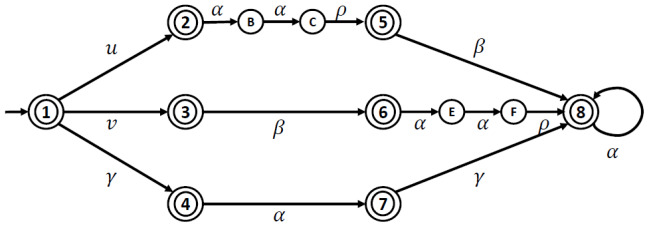

Fig. 22 Extended automaton $\bar{G}^e$ of Example 6

CA-diagnosability was presented, based on model transformation. This model transformation also allows us to design a CA-diagnoser for a system under sensor attacks if CA-diagnosability is satisfied. Some sufficient conditions were derived on attackers, which can be easily checked, ensuring that diagnosability implies CA-diagnosability. Conditions under which the role of an attacker can be reverted from malicious to benevolent were also investigated. Finally, the detection of the attacker was considered from the viewpoint of diagnosing a triggering (unobservable) attack event embedded in the system model. Results were obtained regarding the ability to detect such attacks using the methodologies from the theory of diagnosability.

In future work, it would be of interest to study in more depth special instances of the general attack model considered in this paper, in order to allow for greater resilience of the diagnostic engine and/or for greater ability at attack detection. It would also be of interest to further develop the case study considered in this paper, where the goal is to achieve resilient diagnosis of failures of the protection relay or the circuit breaker in a prototypical power system.

10 Appendix

Proof of Theorem 1

Let us take negations of the conditions in Theorem 1 as follows.

$$
\begin{aligned}
&\neg(\exists n \in \mathcal{N})(\forall s \in \Psi(L(G)))(\forall u \in L(G)/s) \\
&\quad |u| \geq n \Rightarrow (\forall v \in (\Phi^a)^{-1}(\Phi^a(su)) \cap L(G))\Sigma_f \in v \\
\Leftrightarrow &(\forall n \in \mathcal{N})(\exists s \in \Psi(L(G)))(\exists u \in L(G)/s) \\
&\quad \neg(|u| \geq n \Rightarrow (\forall v \in (\Phi^a)^{-1}(\Phi^a(su)) \cap L(G))\Sigma_f \in v) \\
\Leftrightarrow &(\forall n \in \mathcal{N})(\exists s \in \Psi(L(G)))(\exists u \in L(G)/s) \\
&\quad |u| \geq n \wedge \neg(\forall v \in (\Phi^a)^{-1}(\Phi^a(su)) \cap L(G))\Sigma_f \in v \\
\Leftrightarrow &(\forall n \in \mathcal{N})(\exists s \in \Psi(L(G)))(\exists u \in L(G)/s) \\
&\quad |u| \geq n \wedge (\exists v \in (\Phi^a)^{-1}(\Phi^a(su)) \cap L(G))\Sigma_f \notin v \\
\Leftrightarrow &(\forall n \in \mathcal{N})(\exists s \in \Psi(L(G)))(\exists u \in L(G)/s)
\end{aligned}
$$

$$
\begin{aligned}
&|u| \geq n \wedge (\exists v \in L(G))v \in (\Phi^a)^{-1}(\Phi^a(su)) \wedge \Sigma_f \notin v \\
\Leftrightarrow &(\forall n \in \mathcal{N})(\exists s \in \Psi(L(G)))(\exists u \in L(G)/s) \\
&|u| \geq n \wedge (\exists v \in L(G))(\exists w \in \Phi^a(su)) \\
&v \in (\Phi^a)^{-1}(w) \wedge \Sigma_f \notin v \\
\Leftrightarrow &(\forall n \in \mathcal{N})(\exists s \in \Psi(L(G)))(\exists u \in L(G)/s) \\
&|u| \geq n \wedge (\exists v \in L(G))(\exists w \in \Phi^a(su)) \\
&w \in \Phi^a(v) \wedge \Sigma_f \notin v \\
\Leftrightarrow &(\forall n \in \mathcal{N})(\exists s \in \Psi(L(G)))(\exists u \in L(G)/s) \\
&|u| \geq n \wedge (\exists v \in L(G))\Phi^a(v) \cap \Phi^a(su) \neq \emptyset \wedge \Sigma_f \notin v
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
&\neg(\exists n' \in \mathcal{N})(\forall s' \in \Psi(L(G^e)))(\forall u' \in L(G^e)/s') \\
&|u'| \geq n' \Rightarrow (\forall v' \in P^{-1}(P(s'u')) \cap L(G^e))\Sigma_f \in v' \\
\Leftrightarrow &(\forall n' \in \mathcal{N})(\exists s' \in \Psi(L(G^e)))(\exists u' \in L(G^e)/s') \\
&\neg(|u'| \geq n' \Rightarrow (\forall v' \in P^{-1}(P(s'u')) \cap L(G^e))\Sigma_f \in v') \\
\Leftrightarrow &(\forall n' \in \mathcal{N})(\exists s' \in \Psi(L(G^e)))(\exists u' \in L(G^e)/s') \\
&|u'| \geq n' \wedge \neg(\forall v' \in P^{-1}(P(s'u')) \cap L(G^e))\Sigma_f \in v' \\
\Leftrightarrow &(\forall n' \in \mathcal{N})(\exists s' \in \Psi(L(G^e)))(\exists u' \in L(G^e)/s') \\
&|u'| \geq n' \wedge (\exists v' \in P^{-1}(P(s'u')) \cap L(G^e))\Sigma_f \notin v' \\
\Leftrightarrow &(\forall n' \in \mathcal{N})(\exists s' \in \Psi(L(G^e)))(\exists u' \in L(G^e)/s') \\
&|u'| \geq n' \wedge (\exists v' \in L(G^e))v' \in P^{-1}(P(s'u')) \wedge \Sigma_f \notin v' \\
\Leftrightarrow &(\forall n' \in \mathcal{N})(\exists s' \in \Psi(L(G^e)))(\exists u' \in L(G^e)/s') \\
&|u'| \geq n' \wedge (\exists v' \in L(G^e))P(v') = P(s'u') \wedge \Sigma_f \notin v'.
\end{aligned}
$$

Hence, equivalently, we need to prove

$$
\begin{aligned}
&(\forall n \in \mathcal{N})(\exists s \in \Psi(L(G)))(\exists u \in L(G)/s) \\
&|u| \geq n \wedge (\exists v \in L(G))\Phi^a(v) \cap \Phi^a(su) \neq \emptyset \wedge \Sigma_f \notin v
\end{aligned}
$$

if and only if

$$
\begin{aligned}
&(\forall n' \in \mathcal{N})(\exists s' \in \Psi(L(G^e)))(\exists u' \in L(G^e)/s') \\
&|u'| \geq n' \wedge (\exists v' \in L(G^e))P(v') = P(s'u') \wedge \Sigma_f \notin v'
\end{aligned}
$$

It is natural to define that a (non-prefix-closed) language is diagnosable if and only if its prefix closure is diagnosable. Hence, we replace $L(G^e)$ by $L_m(G^e)$ as

$$
\begin{aligned}
&(\forall n' \in \mathcal{N})(\exists s' \in \Psi(L(G^e)))(\exists u' \in L(G^e)/s') \\
&|u'| \geq n' \wedge (\exists v' \in L(G^e))P(v') = P(s'u') \wedge \Sigma_f \notin v'
\end{aligned}
$$

$$\begin{aligned}&\Leftrightarrow(\forall n' \in \mathcal{N})(\exists s' \in \Psi(L_m(G^e)))(\exists u' \in L_m(G^e)/s')\\&\quad |u'| \geq n' \wedge (\exists v' \in L_m(G^e))P(v') = P(s'u') \wedge \Sigma_f \notin v'.\end{aligned}$$

By Equation (5), $L_m(G^e) = \Theta^a(L(G))$. Thus,

$$\begin{aligned}&(\forall n' \in \mathcal{N})(\exists s' \in \Psi(L_m(G^e)))(\exists u' \in L_m(G^e)/s')\\&\quad |u'| \geq n' \wedge (\exists v' \in L_m(G^e))P(v') = P(s'u') \wedge \Sigma_f \notin v'\\&\Leftrightarrow(\forall n' \in \mathcal{N})(\exists s' \in \Psi(\Theta^a(L(G))))(\exists u' \in \Theta^a(L(G))/s')\\&\quad |u'| \geq n' \wedge (\exists v' \in \Theta^a(L(G)))P(v') = P(s'u') \wedge \Sigma_f \notin v'.\end{aligned}$$

Therefore, equivalently, we need to prove

$$\begin{aligned}&(\forall n \in \mathcal{N})(\exists s \in \Psi(L(G)))(\exists u \in L(G)/s)\\&|u| \geq n \wedge (\exists v \in L(G))\Phi^a(v) \cap \Phi^a(su) \neq \emptyset \wedge \Sigma_f \notin v\end{aligned} \tag{18}$$

if and only if

$$\begin{aligned}&(\forall n' \in \mathcal{N})(\exists s' \in \Psi(\Theta^a(L(G))))(\exists u' \in \Theta^a(L(G))/s')\\&|u'| \geq n' \wedge (\exists v' \in \Theta^a(L(G)))P(v') = P(s'u') \wedge \Sigma_f \notin v'.\end{aligned} \tag{19}$$

Let us first prove

$$\begin{aligned}&(\exists s \in \Psi(L(G)))(\exists u \in L(G)/s)\\&(\exists v \in L(G))\Phi^a(v) \cap \Phi^a(su) \neq \emptyset \wedge \Sigma_f \notin v\end{aligned} \tag{20}$$

if and only if

$$\begin{aligned}&(\exists s' \in \Psi(\Theta^a(L(G))))(\exists u' \in \Theta^a(L(G))/s')\\&(\exists v' \in \Theta^a(L(G)))P(v') = P(s'u') \wedge \Sigma_f \notin v'.\end{aligned} \tag{21}$$

Proof of (21) $\Rightarrow$ (20): Suppose (21) is true. Then, for s', u', v' in (21), we have

$$\begin{aligned}&s'u' \in \Theta^a(L(G)) \wedge v' \in \Theta^a(L(G))\\&\Rightarrow(\exists su \in L(G))s'u' \in \Theta^a(su)\\&\quad \wedge (\exists v \in L(G))v' \in \Theta^a(v).\end{aligned}$$

Since $s' \in \Psi(\Theta^a(L(G))) \Rightarrow \Sigma_f \in s'u'$. By Assumption A3, $\Sigma_f \in su$. Let $su \in L(G)$ be such that $s \in \Psi(L(G)) \wedge s' \in \Theta^a(s)$. Then,

$$\begin{aligned}&su \in L(G) \Rightarrow u \in L(G)/s\\&\Sigma_f \notin v' \Rightarrow \Sigma_f \notin v \quad \text{(by Assumption A3)}\\&P(v') = P(s'u') \Rightarrow (\exists w)w = P(v') = P(s'u')\\&\Rightarrow (\exists w)w \in P(\Theta^a(v)) \wedge w \in P(\Theta^a(su))\\&\Rightarrow (\exists w)w \in \Phi^a(v) \wedge w \in \Phi^a(su)\\&\Rightarrow \Phi^a(v) \cap \Phi^a(su) \neq \emptyset.\end{aligned}$$

Therefore,

$$(\exists s \in \Psi(L(G)))(\exists u \in L(G)/s)$$
$$(\exists v \in L(G))\Phi^a(v) \cap \Phi^a(su) \neq \emptyset \wedge \Sigma_f \notin v,$$

that is, (20) is true.
Proof of (20) $\Rightarrow$ (21): Suppose (20) is true. Then, for s, u, v in (20), we have

$$\begin{aligned}
&\Phi^a(v) \cap \Phi^a(su) \neq \emptyset \\
\Rightarrow &P(\Theta^a(v)) \cap P(\Theta^a(su)) \neq \emptyset \\
\Rightarrow &(\exists w) w \in P(\Theta^a(v)) \wedge w \in P(\Theta^a(su)) \\
\Rightarrow &(\exists w, v', s'u') v' \in \Theta^a(v) \wedge s'u' \in \Theta^a(su) \\
&\wedge w = P(v') = P(s'u') \\
\Rightarrow &(\exists v', s'u') v' \in \Theta^a(v) \wedge s'u' \in \Theta^a(su) \\
&\wedge P(v') = P(s'u').
\end{aligned}$$

Since $s \in \Psi(L(G)) \Rightarrow \Sigma_f \in su$. By Assumption A3, $\Sigma_f \in s'u'$. Let $s'u' \in \Theta^a(su)$ be such that $s' \in \Psi(\Theta^a(L(G))) \wedge s' \in \Theta^a(s)$. Then,

$$\begin{aligned}
&su \in L(G) \Rightarrow s'u' \in \Theta^a(L(G)) \Rightarrow u' \in \Theta^a(L(G))/s' \\
&v \in L(G) \Rightarrow v' \in \Theta^a(L(G)) \\
&\Sigma_f \notin v \Rightarrow \Sigma_f \notin v' (\text{ by Assumption A3}).
\end{aligned}$$

Therefore,

$$(\exists s' \in \Psi(\Theta^a(L(G))))(\exists u' \in \Theta^a(L(G))/s')$$
$$(\exists v' \in \Theta^a(L(G)))P(v') = P(s'u') \wedge \Sigma_f \notin v'.$$

that is, (20) is true.

Let us now prove (18) if and only if (19).
Proof of (19) $\Rightarrow$ (18): Suppose (19) is true. Then, for any $n \in \mathcal{N}$, let $n' = n + d$, where d is given in Assumption A4.

Because (21) $\Rightarrow$ (20), we have

$$\begin{aligned}
&(\exists s' \in \Psi(\Theta^a(L(G))))(\exists u' \in \Theta^a(L(G))/s') \\
&|u'| \geq n' \wedge (\exists v' \in \Theta^a(L(G)))P(v') = P(s'u') \wedge \Sigma_f \notin v' \\
\Rightarrow &(\exists s \in \Psi(L(G)))(\exists u \in L(G)/s) \\
&(\exists v \in L(G))\Phi^a(v) \cap \Phi^a(su) \neq \emptyset \wedge \Sigma_f \notin v
\end{aligned}$$

Furthermore, by the proof of (21) $\Rightarrow$ (20),

$$\begin{aligned}
&s \in L(G) \wedge u \in L(G)/s \wedge s' \in \Theta^a(s) \\
&\wedge u' \in \Theta^a(L(G))/s' \wedge s'u' \in \Theta^a(su)
\end{aligned}$$

By Assumption A4, we have $|(|u| - |u'|)| \leq d$. Hence,

$$\begin{aligned} & |u'| \geq n' \wedge |(|u| - |u'|)| \leq d \\ \Rightarrow & |u'| \geq n' \wedge |u'| - |u| \leq d \\ \Rightarrow & |u'| \geq n' \wedge |u| \geq |u'| - d \\ \Rightarrow & |u| \geq n' - d = n. \end{aligned}$$

Therefore, (18) is true.
Proof of (18) $\Rightarrow$ (19): Suppose (18) is true. Then, for any $n' \in \mathcal{N}$, let $n = n' + d$, where d is given in Assumption A4.

Because (20) $\Rightarrow$ (21), we have

$$\begin{aligned} & (\exists s \in \Psi(L(G)))(\exists u \in L(G)/s) \\ & |u| \geq n \wedge (\exists v \in L(G))\Phi^a(v) \cap \Phi^a(su) \neq \emptyset \wedge \Sigma_f \notin v \\ \Rightarrow & (\exists s' \in \Psi(\Theta^a(L(G))))(\exists u' \in \Theta^a(L(G))/s') \\ & (\exists v' \in \Theta^a(L(G)))P(v') = P(s'u') \wedge \Sigma_f \notin v'. \end{aligned}$$

Furthermore, by the proof of (20) $\Rightarrow$ (21),

$$\begin{aligned} & s \in L(G) \wedge u \in L(G)/s \wedge s' \in \Theta^a(s) \\ & \wedge u' \in \Theta^a(L(G))/s' \wedge s'u' \in \Theta^a(su) \end{aligned}$$

By Assumption A4, we have $|(|u| - |u'|)| \leq d$. Hence,

$$\begin{aligned} & |u| \geq n \wedge |(|u| - |u'|)| \leq d \\ \Rightarrow & |u| \geq n \wedge |u| - |u'| \leq d \\ \Rightarrow & |u| \geq n \wedge |u'| \geq |u| - d \\ \Rightarrow & |u'| \geq n - d = n'. \end{aligned}$$

Therefore, (19) is true.

■

References

1. M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 40, no. 9, pp. 1555–1575, 1995.
2. M. R. C. Alves, P. N. Pena, and K. Rudie, "Discrete-event systems subject to unknown sensor attacks," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 32, pp. 143–158, Mar. 2022.
3. S. Matsui and S. Lafortune, "Synthesis of winning attacks on communication protocols using supervisory control theory: two case studies," *Discrete Event Dynamic Systems*, vol. 32, no. 4, pp. 573–610, 2022.
4. M. Wakaiki, P. Tabuada, and J. P. Hespanha, "Supervisory control of discrete-event systems under attacks," *Dynamic Games and Applications*, vol. 9, no. 4, pp. 965–983, 2019.
5. L. K. Carvalho, Y.-C. Wu, R. Kwong, and S. Lafortune, "Detection and mitigation of classes of attacks in supervisory control systems," *Automatica*, vol. 97, pp. 121–133, 2018.

6. L. Lin, Y. Zhu, and R. Su, "Synthesis of covert actuator attackers for free," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 30, pp. 561–577, 2020.
7. A. Rashidinejad, B. Wetzels, M. Reniers, L. Lin, Y. Zhu, and R. Su, "Supervisory control of discrete-event systems under attacks: An overview and outlook," in *2019 18th European Control Conference (ECC)*, pp. 1732–1739, IEEE, 2019.
8. R. Su, "Supervisor synthesis to thwart cyber attack with bounded sensor reading alterations," *Automatica*, vol. 94, pp. 35–44, 2018.
9. J. C. Basilio, C. N. Hadjicostis, and R. Su, "Analysis and control for resilience of discrete event systems: Fault diagnosis, opacity and cyber security," *Foundations and Trends in Systems and Control*, vol. 8, no. 4, pp. 285–443, 2021.
10. C. N. Hadjicostis, S. Lafortune, F. Lin, and R. Su, "Cybersecurity and supervisory control: A tutorial on robust state estimation, attack synthesis, and resilient control," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 3020–3040, IEEE, 2022.
11. I. R. Porche III, *Cyberwarfare: An Introduction to Information-Age Conflict*. Artech House, 2019.
12. S. M. Dibaji, M. Pirani, D. B. Flamholz, A. M. Annaswamy, K. H. Johansson, and A. Chakrabortty, "A systems and control perspective of CPS security," *Annual Reviews in Control*, vol. 47, pp. 394–411, 2019.
13. W. Duo, M. Zhou, and A. Abusorrah, "A survey of cyber attacks on cyber physical systems: Recent advances and challenges," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 5, pp. 784–800, 2022.
14. P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM journal on Control and Optimization*, vol. 25, no. 1, pp. 206–230, 1987.
15. F. Lin and W. M. Wonham, "On observability of discrete-event systems," *Information sciences*, vol. 44, no. 3, pp. 173–198, 1988.
16. C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Springer Nature, 3rd ed., 2021.
17. W. M. Wonham and K. Cai, *Supervisory control of discrete-event systems*. Springer, 2019.
18. W. Wonham, K. Cai, and K. Rudie, "Supervisory control of discrete-event systems: A brief history," *Annual Reviews in Control*, vol. 45, pp. 250–256, 2018.
19. S. Zheng, S. Shu, and F. Lin, "Modeling and control of discrete event systems under joint sensor-actuator cyber attacks," in *IEEE International Conference on Automation, Control and Robotics Engineering (CACRE 2021)*, pp. 1–8, IEEE, 2021.
20. L. K. Carvalho, M. V. Moreira, and J. C. Basilio, "Comparative analysis of related notions of robust diagnosability of discrete-event systems," *Annual Reviews in Control*, vol. 51, pp. 23–36, 2021.
21. S. Takai, "A general framework for diagnosis of discrete event systems subject to sensor failures," *Automatica*, vol. 129, p. 109669, 2021.
22. Q. Zhang, C. Seatzu, Z. Li, and A. Giua, "Selection of a stealthy and harmful attack function in discrete event systems," *Scientific Reports*, vol. 12, Sept. 2022.
23. Y. Li, C. N. Hadjicostis, and N. Wu, "Tamper-tolerant diagnosability under bounded or unbounded attacks," *IFAC-PapersOnLine*, vol. 55, no. 28, pp. 52–57, 2022.
24. Y. Tong, Y. Wang, and A. Giua, "A polynomial approach to verifying the existence of a threatening sensor attacker," *IEEE Control Systems Letters*, vol. 6, pp. 2930–2935, 2022.
25. M. V. Alves, R. J. Barcelos, L. K. Carvalho, and J. C. Basilio, "Robust decentralized diagnosability of networked discrete event systems against dos and deception attacks," *Nonlinear Analysis: Hybrid Systems*, vol. 44, p. 101162, 2022.
26. Q. Zhang, Z. Li, C. Seatzu, and A. Giua, "Stealthy attacks for partially-observed discrete event systems," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, pp. 1161–1164, IEEE, 2018.
27. R. Meira-Góes, E. Kang, R. H. Kwong, and S. Lafortune, "Synthesis of sensor deception attacks at the supervisory layer of cyber–physical systems," *Automatica*, vol. 121, p. 109172, 2020.
28. R. Tai, L. Lin, and R. Su, "Synthesis of optimal covert sensor–actuator attackers for discrete-event systems," *Automatica*, vol. 151, p. 110910, 2023.

29. A. Ghasaei, Z. J. Zhang, W. M. Wonham, and R. Iravani, "A discrete-event supervisory control for the AC microgrid," *IEEE Transactions on Power Delivery*, vol. 36, no. 2, pp. 663–675, 2020.
30. A. Kharrazi, Y. Mishra, and V. Sreeram, "Discrete-event systems supervisory control for a custom power park," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 483–492, 2017.
31. S. Reshmila and R. Devanathan, "Diagnosis of power system failures using observer based discrete event system," in *2016 IEEE First International Conference on Control, Measurement and Instrumentation (CMI)*, pp. 131–135, IEEE, 2016.
32. B. Zhao, F. Lin, C. Wang, X. Zhang, M. P. Polis, and L. Y. Wang, "Supervisory control of networked timed discrete event systems and its applications to power distribution networks," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 2, pp. 146–158, 2015.
33. J. H. Saleh, E. A. Saltmarsh, F. M. Favarò, and L. Brevault, "Accident precursors, near misses, and warning signs: Critical review and formal definitions within the framework of discrete event systems," *Reliability Engineering & System Safety*, vol. 114, pp. 148–154, 2013.
34. M. Romero-Rodríguez, R. Delpoux, L. Piétrac, J. Dai, A. Benchaib, and E. Niel, "An implementation method for the supervisory control of time-driven systems applied to high-voltage direct current transmission grids," *Control Engineering Practice*, vol. 82, pp. 97–107, 2019.
35. F. Lin, S. Lafortune, and C. Wang, "Diagnosability of discrete event systems under sensor attacks," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 3572–3578, 2023.
36. C. N. Hadjicostis, *Estimation and Inference in Discrete Event Systems*. Springer, 2021.
37. C. Seatzu, M. Silva, and J. H. Van Schuppen, *Control of discrete-event systems*, vol. 433. Springer, 2013.