Diagnosability of Discrete Event Systems under Sensor Attacks ⋆

Feng Lin * **Stéphane Lafortune** ** **Caisheng Wang** *

* *Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202, USA (flin@wayne.edu; cwang@wayne.edu)*
** *Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA (stephane@umich.edu)*

Abstract: This paper considers fault diagnosis in discrete event systems modeled by finite-state automata, according to the theory of diagnosability, but it assumes that an attacker has compromised the communication channel from the system's sensors to the diagnostic engine. The attacker operates according to a general attack model that has been studied previously in the context of supervisory control, but not in the context of fault diagnosis. Specifically, the attacker is able to replace each occurrence of a compromised observable event with a string in an attack sublanguage; in particular, this general model embeds event insertion and deletion as well as static and dynamic attacks. The new notion of CA-diagnosability is defined in order to formally capture the ability of the diagnostic engine to still diagnose the occurrences of faults in the presence of the attacker, as captured by its attack model. This extends prior results on supervisory control under attack, where the corresponding properties of CA-controllability and CA-observability were introduced, to the realm of fault diagnosis. A testing procedure for CA-diagnosability is developed and its correctness is proved. Then, diagnosability theory is used to study conditions under which the presence of the attacker can be detected based on the corrupted observations. The results in the paper are illustrated using an example of a protection relay and a circuit breaker in a power system, where the faults are the failures of the protection relay or of the circuit breaker.

Keywords: Discrete event systems, cyber attacks, diagnosability.

1. INTRODUCTION

We consider the standard set-up of event diagnosis in discrete event systems modeled by finite-state automata, as in Sampath et al. (1995). 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; a list of references can be found in the recent book by Basilio et al. (2021) and survey/tutorial paper by Hadjicostis et al. (2022). The increased interest in these problems is motivated by concerns regarding cyber-attacks on both cyber and cyber-physical systems (CPS) (Porche III, 2019; Dibaji et al., 2019; Duo et al., 2022). 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.

In this paper, we consider a general nondeterministic attack model on the sensors, where each compromised observable event can be edited by a sublanguage (e.g., event replacement or insertion); this is similar to the attack model for sensors in Zheng et al. (2021), which pertains to supervisory control. We study diagnostic performance in the presence of this class of deception 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 Sampath et al. (1995), captures this objective in the absence of an attacker. Various techniques exist in the literature to test diagnosability (see, e.g., (Cassandras and Lafortune, 2021)). We do not assume a specific diagnostic engine in this paper, but it could be a *diagnoser automaton*, as defined in Sampath et al. (1995).

Our contributions cover two distinct problems under the same general attack model, as described above. In the first part of the paper, we define the notion of CA-diagnosability (CA for "cyber attack") that captures the ability to still diagnose the occurrences of the unobservable event of interest in the presence of the attacker, as captured by its attack model. This extends the results in Zheng et al. (2021), which focused on supervisory control and introduced the corresponding properties of CA-

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

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. (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.) We prove the correctness of this approach to test CA-diagnosability under some assumptions about the attacker. Owing to the general 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., (Carvalho et al., 2021; Takai, 2021; Zhang et al., 2022; Li et al., 2022; Tong et al., 2022; Alves et al., 2022).

In the second part of the paper, we use diagnosability theory itself to detect the presence of an attacker, using the same general attack model as in the first part of the paper. 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 attack model, we show that if the 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 attack 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 Zhang et al. (2018); Meira-Góes et al. (2020); Lin et al. (2020); Zhang et al. (2022).

The two aspects of our contributions [1] 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 the theoretical results in this paper by 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 (Biswas et al., 2004; Ghasaei et al., 2020; Afzalian et al., 2008; Kharrazi et al., 2017; Reshmila and Devanathan, 2016; Zhao et al., 2012); our focus on attacks on diagnosers differentiate 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, Sections 3 and 4 describe 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. Then, the results on diagnosing the attack itself, Theorems 2 and 3, are presented in Section 6. Section 7 concludes the paper.

[1] The proofs of our results are omitted due to space limitations. They can be obtained from the authors.

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 (Cassandras and Lafortune, 2021; Hadjicostis, 2021):

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

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.

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 (Cassandras and Lafortune, 2021). 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 possible transitions 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 observation below.

As in Sampath et al. (1995), 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, $(\forall q \in Q)(\forall s \in \Sigma^*)\delta(q, s) = q \wedge |s| > 0 \Rightarrow s \notin \Sigma_{uo}^*$.

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 number of events occurred afterwards. Formally, diagnosability is defined in Sampath et al. (1995) as follows.

Definition 1. (Sampath et al., 1995)

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:

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

In Sampath et al. (1995), 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

As in Zheng et al. (2021), 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.

For a given transition $tr = (q, \sigma, q') \in \delta^a$, we assume that an attacker can change the event σ to any string in a 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

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

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

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

We replace each 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'.

Denote the extended automaton obtained after replacing all transitions subject to attacks as

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

where $\tilde{Q}$ 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

$$G_\varepsilon^e = (Q^e, \Sigma_o, \delta_\varepsilon^e, q_o, Q_m^e) = (Q \cup \tilde{Q}, \Sigma_o, \delta_\varepsilon^e, q_o, Q)$$

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

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

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

4. ATTACKER MODEL

Sensor attacks can be either "static" or "dynamic." A sensor attack is static if, 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 other words, whenever an attacker sees an event $\sigma \in \Sigma_o^a$, it will replace σ with some strings in the same A_{tr}. Hence, A_{tr} can also be written as A_σ.

A sensor attack is dynamic if the above is not true, 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 $\sigma_1 = \sigma_2$.

Our results on diagnosability under sensor attacks work for both static and dynamic attacks. However, different attacker models need to be used to implement the static or dynamic attacks.

For static attacks, the attacker models to implement sensor attacks are simple: whenever an attacker see an event $\sigma \in \Sigma_o^a$, it will replace σ with some strings in the same A_σ.

For dynamic attacks, the attacker models to implement sensor attacks are more complex: when an attacker see an event $\sigma \in \Sigma_o^a$, it needs to decide which A_{tr} to use, because the same σ may correspond to different $tr = (q, \sigma, q')$. This decision will depend on the string of events the attacker has observed so far. Since the number of all possible observed strings may be infinite, a finite model must be used.

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 $\Sigma_o^a \subseteq \Sigma_{ao}$, 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 attack model is based on a finite automaton on Σ_{ao}:

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

It is required that $P_{ao}(L(G)) \subseteq L(H)$. For example, we can use 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 transitions that can be attacked is denoted by $\zeta^a = \{(y, \sigma, y') \in \zeta : \sigma \in \Sigma_o^a\}$. For each transition $tr = (y, \sigma, y') \in \zeta^a$, the attacker can change the event σ to any string in a language $A_{tr}^H \subseteq \Sigma^*$.

Note that if H has only one state $Y = \{y_o\}$ and all transitions form self-loops at y_o, then the dynamic attack model reduces to the static attack model. Hence, static attacks are a special case of dynamic attacks.

We "embed" H into G so that the attacks can be implemented based on H as follows. Take the parallel composition of G and H (Cassandras and Lafortune, 2021):

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

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 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.

Example 1. Let us illustrate the above modeling procedure using an example of a protection relay and a circuit breaker in a power system, as shown in Fig. 1. 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. 1, then the protection relay will be triggered. When the protection relay is triggered and closed, the circuit breaker will open and cut the faulty power line from the power system. After the circuit breaker is open for a short period of time, it will automatically try to reclose. 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 circuit breaker will return to and remain open until the repair is made to the power line.

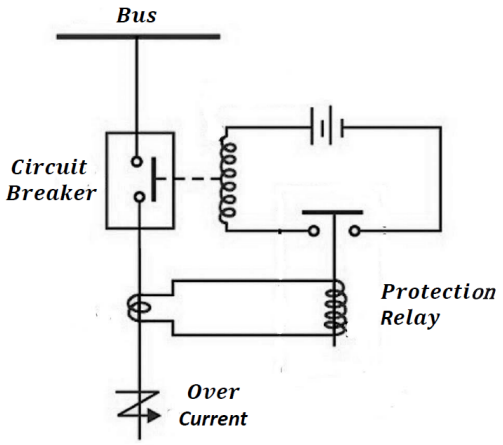

Fig. 1. Protection relay and circuit breaker

We model the system by the automaton G shown in Fig. 2. 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.

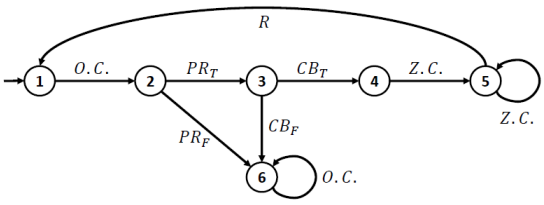

Fig. 2. Automaton G for protection relay and circuit breaker

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 (Sampath et al., 1995; Cassandras and Lafortune, 2021). 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. 3. Note that G is already embedded with the attacker model H shown in Fig. 4 because G is isomorphic to $G||H$.

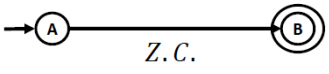

Fig. 3. Automaton F_{tr} describing the attack

For the automaton G shown in Fig. 2, we can construct the extended automaton G^e as shown in Fig. 5. Note that

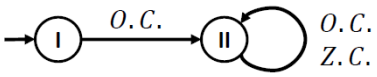

Fig. 4. Attacker model H

the extended automaton G^e in Fig. 5 is equivalent to the automaton in Fig. 6.

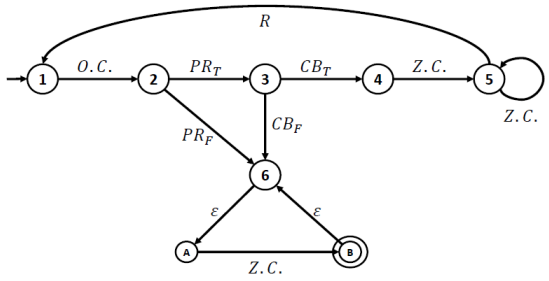

Fig. 5. The extended automaton G^e

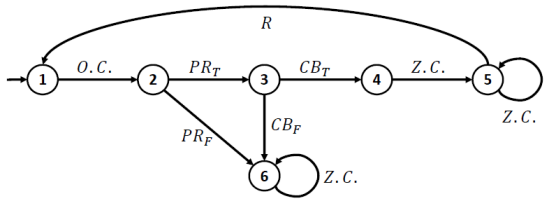

Fig. 6. Equivalent automaton of G^e in Fig. 5

In general, if for transition $tr = (q, \sigma, q') \in \delta^a$, F_{tr} has only one marked state, that is, $Q_{m,tr} = \{q_{m,tr}\}$, then we can eliminate ε-transitions in $G_{tr\to(q,F_{tr},q')}$ by combining state q with $q_{o,tr}$ and state q' with $q_{m,tr}$. In the rest of the paper, we will do this whenever possible.

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

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

Let us now state 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.

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 2. 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 (Sampath et al., 1995; Cassandras and Lafortune, 2021), we can check that G^e is not diagnosable with respect to P. Intuitively, this is because 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.

6. 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., (Zhang et al., 2018; Meira-Góes et al., 2020; Zhang et al., 2022).

Another method to detect sensor attacks is to translate the sensor attack detection problem into a diagnosis problem 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.

$$\overline{A}_{tr} = (A_{tr} - \{\sigma\})\{\rho\} \cup (\{\sigma\} \cap A_{tr}). \tag{10}$$

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,

$$(\forall tr = (q, \sigma, q') \in \delta^a)\sigma \in A_{tr}. \tag{11}$$

Note that Assumption A5 implies that $L(G) \subseteq L(\bar{G}^e)$.

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

Theorem 2. 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,

$$\begin{aligned}&(\forall s\rho \in L(\bar{G}^e))(\forall n \in \mathcal{N})(\exists u \in L(\bar{G}^e)/s\rho)\\ &|u| \geq n \wedge (\exists v \in L(\bar{G}^e))P(v) = P(s\rho u) \wedge \rho \notin v.\end{aligned} \tag{12}$$

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

Theorem 3. 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.

It is easy to see that Equation (12) is stronger than non-diagnosability. Intuitively, this is because Equation (12) requires that all attacks are not detectable, while non-diagnosability only requires that some attacks 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.

Example 3. Let us consider the system modeled by G shown in Fig. 7. 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 can be attacked, that is, $\Sigma_o^a = \{\alpha\}$.

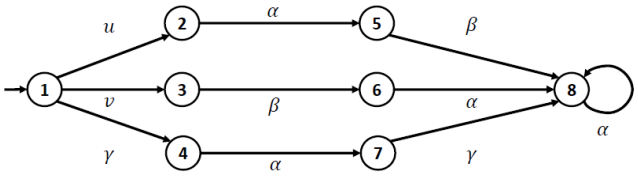

Fig. 7. Automaton G of the system in Example 3

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. 8.

The resulting extended automaton $\bar{G}^e$ is shown in Fig. 9. 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.

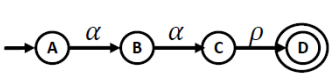

Fig. 8. Automaton $\bar{F}_{tr}$ of Example 3

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 $\alpha \notin A_{tr} = \{\alpha\alpha\}$.

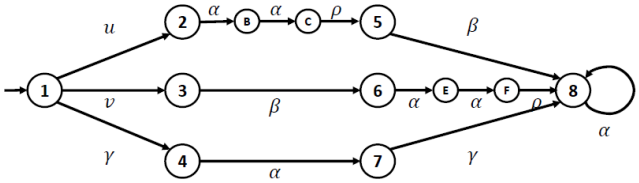

Fig. 9. Extended automaton $\bar{G}^e$ of Example 3

7. 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 CA-diagnosability was presented, based on model transformation. Furthermore, the detection of the attacker was also considered from the view point 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.

REFERENCES

Afzalian, A.A., Niaki, S.A.N., Iravani, M.R., and Wonham, W. (2008). Discrete-event systems supervisory control for a dynamic flow controller. *IEEE Transactions on Power Delivery*, 24(1), 219–230.

Alves, M.V., Barcelos, R.J., Carvalho, L.K., and Basilio, J.C. (2022). Robust decentralized diagnosability of networked discrete event systems against dos and deception attacks. *Nonlinear Analysis: Hybrid Systems*, 44, 101162.

Basilio, J.C., Hadjicostis, C.N., and Su, R. (2021). Analysis and control for resilience of discrete event systems: Fault diagnosis, opacity and cyber security. *Foundations and Trends in Systems and Control*, 8(4), 285–443.

Biswas, T., Davari, A., and Feliachi, A. (2004). Application of discrete event systems theory for modeling and analysis of a power transmission network. In *IEEE PES Power Systems Conference and Exposition*, 1024–1029. IEEE.

Carvalho, L.K., Moreira, M.V., and Basilio, J.C. (2021). Comparative analysis of related notions of robust diagnosability of discrete-event systems. *Annual Reviews in Control*, 51, 23–36.

Cassandras, C.G. and Lafortune, S. (2021). *Introduction to Discrete Event Systems*. Springer Nature, 3rd edition.

Dibaji, S.M., Pirani, M., Flamholz, D.B., Annaswamy, A.M., Johansson, K.H., and Chakrabortty, A. (2019). A systems and control perspective of CPS security. *Annual Reviews in Control*, 47, 394–411.

Duo, W., Zhou, M., and Abusorrah, A. (2022). A survey of cyber attacks on cyber physical systems: Recent advances and challenges. *IEEE/CAA Journal of Automatica Sinica*, 9(5), 784–800.

Ghasaei, A., Zhang, Z.J., Wonham, W.M., and Iravani, R. (2020). A discrete-event supervisory control for the AC microgrid. *IEEE Transactions on Power Delivery*, 36(2), 663–675.

Hadjicostis, C.N. (2021). *Estimation and Inference in Discrete Event Systems*. Springer.

Hadjicostis, C.N., Lafortune, S., Lin, F., and Su, R. (2022). 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)*, 3020–3040. IEEE.

Kharrazi, A., Mishra, Y., and Sreeram, V. (2017). Discrete-event systems supervisory control for a custom power park. *IEEE Transactions on Smart Grid*, 10(1), 483–492.

Li, Y., Hadjicostis, C.N., and Wu, N. (2022). Tamper-tolerant diagnosability under bounded or unbounded attacks. *IFAC-PapersOnLine*, 55(28), 52–57.

Lin, L., Zhu, Y., and Su, R. (2020). Synthesis of covert actuator attackers for free. *Discrete Event Dynamic Systems: Theory and Applications*, 30, 561–577.

Meira-Góes, R., Kang, E., Kwong, R.H., and Lafortune, S. (2020). Synthesis of sensor deception attacks at the supervisory layer of cyber–physical systems. *Automatica*, 121, 109172.

Porche III, I.R. (2019). *Cyberwarfare: An Introduction to Information-Age Conflict*. Artech House.

Reshmila, S. and Devanathan, R. (2016). Diagnosis of power system failures using observer based discrete event system. In *2016 IEEE First International Conference on Control, Measurement and Instrumentation (CMI)*, 131–135. IEEE.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., and Teneketzis, D. (1995). Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, 40(9), 1555–1575.

Takai, S. (2021). A general framework for diagnosis of discrete event systems subject to sensor failures. *Automatica*, 129, 109669.

Tong, Y., Wang, Y., and Giua, A. (2022). A polynomial approach to verifying the existence of a threatening sensor attacker. *IEEE Control Systems Letters*, 6, 2930–2935.

Zhang, Q., Li, Z., Seatzu, C., and Giua, A. (2018). Stealthy attacks for partially-observed discrete event systems. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, 1161–1164. IEEE.

Zhang, Q., Seatzu, C., Li, Z., and Giua, A. (2022). Selection of a stealthy and harmful attack function in discrete event systems. *Scientific Reports*, 12.

Zhao, J., Chen, Y.L., Chen, Z., Lin, F., Wang, C., and Zhang, H. (2012). Modeling and control of discrete event systems using finite state machines with variables and their applications in power grids. *Systems & Control Letters*, 61(1), 212–222.

Zheng, S., Shu, S., and Lin, F. (2021). 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)*, 1–8. IEEE.