Control Theory Tech, Vol. 17, No. 3, pp. 265–275, August 2019

Control Theory and Technology

http://link.springer.com/journal/11768

Diagnosability of a class of discrete event systems based on observations

S. RESHMILA[1†], Devanathan RAJAGOPALAN[2]

1.*Sree Narayana Gurukulam College of Engineering, Kerala, India;*

2.*Hindustan Institute of Technology and Science, Chennai, India*

Received 22 December 2017; revised 27 December 2018; accepted 18 January 2019

Abstract

The diagnosability of discrete event systems has been a topic of interest to many researchers. The diagnosability conditions for various systems have evolved based on a regularity condition that is imposed on faulty traces with respect to their observable continuations. Improving upon this weak but necessary condition, a new model of diagnosability that is based on sensor outputs, which are called observations, upon a command input is proposed in this paper. Necessary and sufficient conditions are derived for the proposed diagnosability model. The search performance of the proposed diagnosability condition is of linear complexity in terms of the power set of the system events and observations, compared to the exponential complexity of the search with the existing diagnosability regularity condition. Moreover, a system that is not diagnosable according to the existing diagnosability condition may be diagnosable in the proposed diagnosability model, which includes observations.

Keywords: Discrete event system, diagnosability, fault diagnosis, mealy automata, finite state automata

DOI https://doi.org/10.1007/s11768-019-7298-3

1 Introduction

The sequential behavior of a system is modeled as a discrete event system (DES) using discrete states and events. DES models can be used for fault diagnosis, but only faults that are considered in the modeling of the system can be diagnosed. However, prior to the fault diagnosis, the diagnosability of the system must be verified.

According to the literature, in the field of fault diagnosis, there are two types of diagnoses: event-based diagnosis and state-based diagnosis. Event-based diagnosis is considered in [1, 2], and state-based diagnosis in [3, 4]. In event-based diagnosis, a fault is considered an unobservable event and is diagnosed using observable events. In state-based diagnosis, binary output vectors that are related to the system state are used to detect and isolate faults. In both methods, a diagnoser is used for the diagnosis. In event-based diagnosis, the diagnoser is initialized during the system initialization,

†Corresponding author.

E-mail: reshmilas@yahoo.com. Tel.: +91 9447231897; fax: +0484 2762541.

which may cause difficulties, whereas in state-based diagnosis, it is possible to initialize the diagnoser at a later stage.

There is a large body of literature based on the diagnosability condition that was put forward by Sampath [1]. However, Sampath's condition is more of a regularity requirement than a model. It states that, a prefix trace that contains a fault and another prefix trace that does not contain the same fault cannot have identical observable suffixes. We argue that such a weak but necessary condition cannot be the basis of an efficient diagnosis technique. In this context, we propose, a command event-based sensor output as a model for diagnosis. We demonstrate that the proposed model leads to a more efficient diagnosis technique than the existing diagnosability condition.

Lamperti et al. [5] used an event-based output model that is similar to ours but more elaborate in structure. Our work differs from that of Lamperti et al. [5] in that the former emphasizes diagnosability, while the latter emphasizes the diagnosis. While diagnosis is performed via a computational approach that involves the interpretation of event sequences, diagnosability requires an automata-theoretic approach, thereby leading to an analysis of the feasibility of the diagnosis prior to the diagnosis. The diagnosability of a DES is considered in this paper.

We briefly review the existing works on diagnosability. In [6], a diagnosability condition was proposed for a system that has an uncertain sequence of observable events. In [7], the prediction of the occurrence of a fault using an asymptotically almost sure predictability (AAS-predictability) for stochastic DESs was proposed, and a necessary and sufficient condition for AAS-predictability is defined. The fault diagnosis of complex communication networks was conducted using the DES approach in [8]. In [9], the diagnosability of transient faults was evaluated within a bounded time interval before the fault is cleared. The diagnosability of intermittent faults based on a twin plant construction was discussed in [10], together with the diagnosability definitions for the occurrence of and recovery from faults. I-diagnosability condition [1] that involves an indicator event simplifies the diagnosability condition that was presented in [1]. According to this condition, a system is said to be non-diagnosable if it is not diagnosable even after the occurrence of the indicator event. I^2-diagnosability condition was defined by including an empowering event in addition to the indicator event in [11]. An empowering event is an event that ensures that the indicator event will be successful in identifying the fault.

Developments in sensor networks have enabled the availability of event-based sensor outputs for fault detection upon the issuance of various commands. For example, in modern numerical relays in power system protection, sensors provide the status of each operation. After a relay has been energized, a sensor will output the status of that relay operation, namely, whether it has been executed correctly or if a fault has occurred. Our work is inspired by such new developments, and we redefine the diagnosability condition by taking advantage of the additional capability that is afforded to the system by these developments. In this paper, we propose a new diagnosability condition for a class of problems for which event-based outputs are utilized, as outlined above. Event-based outputs are associated with Mealy automata [12] in the transitions from one state to another. Although Mealy automata can be transformed to regular automata [12], we prefer to retain the former to emphasize the role of event outputs in the simplification of the diagnosability of a class of discrete event systems.

In Sampath's approach, a diagnoser is constructed, and in the process, fault mapping is conducted using associated event and state output combinations via the inspection of the generated cycles that are involved in the fault. However, under the proposed diagnosability condition, we approach the problem of fault mapping up front in terms of observations prior to building the diagnoser and generating the cycles. In other words, we apply the diagnosability condition deeper in the domain knowledge by considering the event input and sensor output that are associated with a fault, in contrast to leaving the diagnoser building process to generate event and state output combinations that lead to fault diagnosis, as in Sampath's case.

Moreover, a system that is not diagnosable according to Sampath's condition may be made diagnosable according to the proposed diagnosability condition via the inclusion of observations in the system. Moreover, under the proposed diagnosability condition, the search, which is limited to faulty traces that have a specified suffix, as a special case, is of linear complexity in the system event power set, compared to an exponential complexity [1] for faulty traces that have an arbitrary suffix, as in Sampath's case, or a polynomial complexity, as in [13].

In earlier works [14–17], the authors demonstrated, through application examples, the role of event-based outputs, which are called observations, in improving

the diagnosability of systems. The present paper is an attempt at formalizing the use of observations and introducing a new diagnosability condition for a class of systems.

The main contribution of the paper is a new model of diagnosability, namely, O-diagnosability that is based on an event-triggered sensor output in a DES. The search for diagnosability verification is shown to be linear in the power set of events of the system in the proposed O-diagnosability, compared to the exponential complexity in the power set in the existing diagnosability. In addition, it is demonstrated that it is possible for a system that is not diagnosable in the existing diagnosability condition to be made O-diagnosable via the inclusion of observations.

The remainder of the paper is organized as follows: Section 2 briefly presents background on the DES and discusses the Mealy model. Section 3 introduces the proposed O-diagnosability model. Section 4 specifies the steps for building a diagnoser based on the new O-diagnosability model. A necessary and sufficient condition for O-diagnosability, together with its comparative search performance, is presented in Section 5. In Section 6, the O-diagnosability model is demonstrated on an example. The conclusions of the paper are presented in Section 7.

2 Discrete event system

2.1 Background

A DES is represented using states and events that cause transitions from one state to another state. The system is denoted by $S = (X, \Sigma, \delta, x_0)$, where X is the set of states, Σ is the set of events, δ is the partial transition function, and x_0 is the initial state of the system. It is assumed that all states are accessible since we cannot diagnose inaccessible states. The set of events consists of observable events and unobservable events. The prefix closed language, namely, $L(S)$, which is represented in short by L, is generated by the system and represents the behavior of the system. L is a subset of Σ^*, where Σ^* represents the Kleene closure of the set Σ. Σ^* is the set of all strings that are generated by concatenating elements of Σ, including the null event ϵ. A trace is an element of Σ^* and represents a sequence of events. The prefix of an event indicates the trace prior to the event, and the suffix indicates the trace after the event. $\sigma \in s$ denotes that event σ is contained in trace s. In the proposed system model, we use observations, or sensor outputs, along with events and transitions.

2.2 System model

The system to be diagnosed is modeled as a Mealy automaton, which is denoted by $G = (X, \Sigma, O, \delta_e, \delta_o, x_0)$, where

- X is the set of states, which includes normal states X_n and faulty states X_f, $X = X_n \cup X_f$;
- Σ is the set of events, which includes observable events (Σ_o) and unobservable events (Σ_{uo}) and satisfies $\Sigma = \Sigma_o \cup \Sigma_{uo}$;
- O is the set of sensor outputs;
- The input transition function $\delta_e : X \times \Sigma \rightarrow X$ is a partial transition function that defines the transition from one state to the next state upon an event in Σ;
- $\delta_o : X \times \Sigma \rightarrow O$ is the output transition function, which yields the output that is produced when an event in Σ acts on a state in X; and
- $x_0 \in X$ is the initial state of the system.

Typically, unobservable events are faults or events that are not recorded by sensors. Observable events are commands that are issued by controllers, and a sensor output is obtained upon the issuance of a command. Since not all command events result in sensor outputs, observable events can further be subdivided as $\Sigma_o = \Sigma_s \cup \Sigma_{ns}$, where Σ_s denotes the set of observable command events that result in sensor outputs and Σ_{ns} denotes the set of observable command events that do not result in sensor outputs.

Let $\Sigma_f \subseteq \Sigma$ denote the set of events that are faults (which are referred to as faulty events henceforth) and $\Sigma_f \subseteq \Sigma_{uo}$ because faulty events are unobservable. The faulty events can be partitioned into disjoint groups of faults $\Sigma_f = \Sigma_{f1} \cup \Sigma_{f2} \cup \Sigma_{f3} \cup \ldots \cup \Sigma_{fn}$. The states that have faults can be grouped according to the partition of the faults: $X_f = X_{f1} \cup X_{f2} \cup X_{f3} \cup \ldots \cup X_{fn}$, where $\{X_{fi}\}, i = 1, 2, \ldots, n$ are disjoint.

The sensor outputs can be state-based sensor outputs or event-based sensor outputs. A state-based sensor output depends on the state of the system, irrespective of the event due to which the state has been reached. An event-based sensor output is produced only upon the occurrence of an event; it indicates the successful/failed execution of the event. An event-based sensor output is called an observation henceforth. Hence, sensor outputs can be subdivided into $O = O_e \cup O_s$, where O_e denotes the set of event-based sensor outputs and O_s

the set of state-based sensor outputs. o_σ represents an observation that is due to event σ, and $o_\sigma \in O_e$. The set of event-based sensor outputs can be further expressed as $O_e = O_{en} \cup O_{ef}$, where O_{en} is the set of observations that indicate the successful (normal) operation of event e and O_{ef} is the set of observations that indicate the failed operation of event e due to fault f. Faulty observations can be associated with unobservable faulty events. We assume that the sensors are fault-free. If sensor faults are to be considered, they must be included in the system model.

2.3 Fault observation

We model the faults and their observations as follows:

Definition 1 (Mapping of faults and observations) The sensor outputs can be mapped to states by a function h, where h (state) $= o$ if o is produced when the state is reached, irrespective of the event that caused the transition. Faults and observations are related to each other in the form of triples $m_{ij} = (\sigma_{fij}, \sigma_i, o_{\sigma i})$, where $\sigma_{fij} \in \Sigma_{fi}, i = \{1, 2, \ldots, n\}, j = \{1, 2, \ldots, n_{fi}\}, \sigma_i \in \Sigma_s$, and $o_{\sigma i} \in O_{ef}$, such that after any fault event $\sigma_{fij} \in \Sigma_{fi}$, the command event σ_i can occur and produces observation $O_{\sigma i}$. The number of faults in fault group Σ_{fi} is denoted by n_{fi}, and the total number of faults in the system is $t_f = \sum_{i=1}^{n} n_{fi}$.

- $M_i = \{m_{i1}, m_{i2}, \ldots, m_{in_{fi}}\}$ corresponds to fault group Σ_{fi}, for $i = 1, 2, \ldots, n$.
- $M = \{M_1 \cup M_2 \cup \ldots \cup M_n\}$ is the set of all such triples.

To illustrate the above definition, suppose that for a pump-valve system, a pressure sensor is available that outputs a positive pressure (PP) and no pressure (NP) upon the Start pump and Stop pump commands, respectively, under normal conditions. If the pump fails, the outputs will differ, and we can map the fault "Pump failed to start" to (Start pump, NP).

Definition 2 The mappings of two fault groups are equal if the (command event, observation) combination is the same for both, that is $M_i = M_j, i, j \in \{1, 2, \ldots, n\}$ iff $\sigma_i = \sigma_j$ and $o_{\sigma i} = o_{\sigma j}$.

3 Diagnosability

3.1 Sampath's diagnosability condition

Let L be the language that is generated by the system and s be a trace in the language. The empty trace is denoted by $\in$. The language after s is denoted by L/s, namely, $L/s = \{t | st \in L\}$. We define $\psi(\sigma) = s\sigma \in L : \sigma \in \Sigma$, i.e., $\psi(\sigma)$ represents the set of all strings of L that end in the event σ. The projection operator $P : \Sigma^* \to \Sigma_o^*$ is defined as in [1]:

$$
\begin{aligned}
&P(\varepsilon) = \varepsilon,\\
&P(\sigma) = \sigma \text{ if } \sigma \in \Sigma_o,\\
&P(\sigma) = \varepsilon \text{ if } \sigma \in \Sigma_{uo},\\
&P(s\sigma) = P(s)P(\sigma) \text{ if } \sigma \in \Sigma \text{ and } s \in \Sigma^*,
\end{aligned}
$$

where ε represents the null string and $s\sigma$ represents a language string that is generated by the system and ends in σ. Thus, projection P erases unobservable events from the event string. The inverse projection is defined as

$$P^{-1}(y) = \{s \in L : P(s) = y\}.$$

The diagnosability condition that was defined by Sampath [1] can be expressed as follows:

$$(\forall i \in \Sigma_f)(\exists n_i \in \mathbb{N})[\forall s \in \psi(\Sigma_{fi})](\forall t \in L/s)[\|t\| \geqslant n_i \Rightarrow D],$$

where D is $\omega \in P_L^{-1}[P(st)] \Rightarrow \Sigma_{fi} \in \omega$.

Remark 1 The Sampath diagnosability condition requires that every trace that produces the same observable string as st should contain the same fault as in s. The sufficient length requirement of t ensures the uniqueness of the suffix whose preimages contain the fault.

3.2 O-diagnosability

Due to the inclusion of observations, the projection operator must be redefined as $P : (\Sigma \cup O)^* \to (\Sigma_o \cup O)^*$. Moreover, $P(\sigma) = \sigma$ if $\sigma \in \Sigma_o \cup O$. The other properties of P remain the same.

Definition 3 (O-diagnosability condition) The O-diagnosability condition for a system requires that the following be satisfied:

C1) $(\forall \sigma_{fi})(\exists M_i)$ and $(\nexists M_i, M_j \in M | M_i = M_j$, where $i \neq j)$;

C2) $(\forall \sigma_{fij}, i = \{1, 2, \ldots, n\}, j = \{1, 2, \ldots, n_{fi}\}, n, n_{fi} \in \mathbb{N})(\forall s \in \psi(\sigma_{fij}))(\exists t_1 t_2 \in L/s | st_1 \in \psi(\sigma_i))$, $st_1 t_2 \in \psi(o_{\sigma i})$, where $\sigma_i \in m_{ij}, (\|t_1 t_2\|) \leqslant n_i \Rightarrow$ OD, where OD is $\omega \in P^{-1}[P(st_1 t_2)] \Rightarrow \sigma_{fij} \in \omega$.

Remark 2 According to condition C1, for all fault groups, there should exist a unique mapping that is not common to any other fault group. This mapping can be verified without taking into consideration the language that is generated by the system. Condition C2 is based

on the language that is generated by the system. According to condition C2, the system is O-diagnosable if for all faults σ_{fij} in every group of fault events Σ_{fi}, for every string "s" that ends with fault event σ_{fij}, there exists a finite continuation (under the assumptions that unobservable strings are bounded in length [1] and of fairness in the occurrence of observable events and observations) t_1t_2 of that string such that trace t_1 ends with command event σ_i and trace t_2 ends with fault observation o_{σ_i}. According to the diagnosability condition, which is denoted by OD, all the prefixes of a string that end in a command event and an observation should contain the fault that is mapped to it, which is a major difference between the diagnosability conditions of Sampath [1] and O-diagnosability. In the case of Sampath, all the strings that have the same suffix should contain the same fault. Therefore, all the strings are to be searched and grouped into sets that have the same suffix without any indication of the type of suffix to be searched. In contrast, under the O-diagnosability condition, the search space for a fault is limited to the strings that contain the mapped command event and observation in the suffix.

3.3 Algorithm for verification of conditions C1 and C2 of O-diagnosability

Steps 1 and 2 for verifying condition C1 and Step 3 for condition C2:

Input $G = (X, \Sigma, O, \delta_e, \delta_o, x_0)$,
$M = \{M_1 \cup M_2 \cup \ldots \cup M_n\}$,
$\Sigma_f = \Sigma_{f1} \cup \Sigma_{f2} \cup \Sigma_{f3} \cup \Sigma_{fn}$.

Step 1 **For** every fault Σ_{fi} do

Check whether there is a mapping M_i to Σ_{fi}, $i = 1, 2, \ldots, n$.

Step 2 **For** every mapping M_i do
Begin
Check ($\nexists M_i, M_j \in M | M_i = M_j$, where $i \neq j$)
If ($\exists M_i = M_j$ where $i \neq j$)
begin
declare not diagnosable
go to end of algorithm
end
end

Step 3 **For** every $(\sigma_i, o_{\sigma i}) \in m_{ij}$ do
Begin
For every $s \in \psi(o_{\sigma i})$ do
begin
check whether $\sigma_{fij} \in s$ and $\sigma_i \in s$
if condition is true, go to next string
if condition is false, declare σ_{fij} to be not diagnosable
end
end

If all faults are diagnosable, the system is diagnosable.

The mapping is evaluated in Step 1 with complexity O(n). The diagnosability is verified in Step 2, which has O(n) complexity if there are n mappings. Only if Step 2 is cleared is Step 3 performed. The worst-case complexity of Step 3 is O(n^2). Therefore, the worst-case complexity of the above algorithm is O(n^2), where n is the number of failure types.

4 Diagnoser

The diagnoser is an FSM that is constructed from the system model G and is used for O-diagnosability verification offline and diagnosis online. The method that is employed is a variation of Sampath's [1] diagnoser model that considers a command input and observations. The diagnoser is denoted by $G_D = (X_d, \Sigma_d, O, \delta_d, x_0)$, where X_d is the set of states; Σ_d is the set of events, which satisfies $\Sigma_d = \Sigma_o \cup O$; x_0 is the initial state; and δ_d is the transition function, which is defined as $\delta_d : X_d \times \Sigma_d \rightarrow X_d$. The set of states in the diagnoser consists of states and labels. The set of labels is denoted by $L = \{N\} \cup \{F_1, F_2, \ldots, F_n\} \cup \{C_e, O\}$, where N indicates the normal state, C_e indicates the occurrence of an event (with sensor output) and O indicates the observation state. Other labels, such as $F_1, F_2, \ldots, F_n$ indicate the fault group. The state is confirmed only via observation.

The states in the diagnoser are of the form $X_d = \{(x_1, l_1), (x_2, l_2), (x_3, l_3), \ldots, (x_k, l_k)\}$, where x_i is the state and l_i is the label. If the system G is initially normal, the initial state of the diagnoser is (x_0, N). The subsequent states are determined using the transition function δ_d; they are the states that are reachable from x_0 under δ_d. For the state (x, l), all possible states (x', l') that are reachable from x are identified:

$$x' = \delta_d(x, s\sigma'),$$

where $\sigma' \in \Sigma_o$ and $s \in \Sigma_{uo}^*$.

Definition 4 The label propagation function, which is denoted by LP, is defined as LP $(x, l, s\sigma') = l'$. For a state x that has label l, the label changes to l' due

to event $s\sigma'$. The label l' depends on s and σ'. When $\sigma' \in \Sigma_{\text{ns}}$, l' depends on s:

$l' = N$ if $\sigma_{\text{f}} \notin s$, $\sigma' \in \Sigma_{\text{ns}}$ and $l = N$,
$l' = F_i$ if $\sigma_{\text{f}i} \in s$, $\sigma' \in \Sigma_{\text{ns}}$ and $l = N$,
$l' = F_iF_j$ if $\sigma_{\text{f}j} \in s$, $\sigma' \in \Sigma_{\text{ns}}$ and $l = F_i$, $i \neq j$.

When the event has a sensor output, the labels are modified to include C_{e}, which indicates that a command event has occurred and an observation is expected. For events that lack a sensor output, the previous comments apply.

$l' = N$ if $\sigma_{\text{f}} \notin s$, $\sigma' \in \Sigma_{\text{s}}$ and $l = N$,
$l' = F_iC_{\text{e}}$ if $\sigma_{\text{f}} \notin s$, $\sigma' = \sigma_i$ and $l = F_i$,
$l' = F_iC_{\text{e}}$ if $\sigma_{\text{f}i} \in s$, $\sigma' = \sigma_i$ and $l = N$,
$l' = F_iF_jC_{\text{e}}$ if $\sigma_{\text{f}j} \in s$, $\sigma' = \sigma_j$ and $l = F_i$, $i \neq j$,
$l' = F_iC_{\text{e}}F_jC_{\text{e}}$ if $\sigma_{\text{f}j} \in s$, $\sigma' = \sigma_j$ and $l = F_iC_{\text{e}}$, $i \neq j$.

If the label contains C_{e} and the event lacks a sensor output, then the labels will be changed if string s is not a null string.

$l' = F_iC_{\text{e}}$ if $\sigma_{\text{f}} \notin s$, $\sigma' \in \Sigma_{\text{ns}}$ and $l = F_iC_{\text{e}}$,
$l' = F_iC_{\text{e}}F_j$ if $\sigma_{\text{f}j} \in s$, $\sigma' \in \Sigma_{\text{ns}}$ and $l = F_iC_{\text{e}}$, $i \neq j$.

When the event is an element of an observation, the label will become O, which indicates the observation state and confirms the state to be normal or faulty.

$l' = F_iC_{\text{e}}$ if $\sigma_{\text{f}} \notin s$, $\sigma' \in O_{\text{en}}$ and $l = F_iC_{\text{e}}$,
$l' = F_iO$ if $\sigma_{\text{f}} \notin s$, $\sigma' \in m_i$, $\sigma' \in O_{\text{ef}}$ and $l = F_iC_{\text{e}}$,
$l' = F_iOF_j$ if $\sigma_{\text{f}j} \in s$, $\sigma' \in \Sigma_{\text{ns}}$ and $l = F_iO$,
$l' = F_iOF_jC_{\text{e}}$ if $\sigma_{\text{f}} \notin s$, $\sigma' \in m_i$, $\sigma' \in O_{\text{ef}}$ and $l = F_iC_{\text{e}}F_j$ or $F_iC_{\text{e}}F_jC_{\text{e}}$,
$l' = F_iC_{\text{e}}F_jO$ if $\sigma_{\text{f}} \notin s$, $\sigma' \in m_j$, $\sigma' \in O_{\text{ef}}$ and $l = F_iF_jC_{\text{e}}$.

Once a fault has been labeled with O, it will remain with this label for all subsequent states. Hence, the O label with a preceding fault indicates the confirmation of its occurrence. If multiple faults occur, the confirmed fault will have the O label, and the others will not.

5 Condition for O-diagnosability

5.1 Necessary and sufficient condition

Definition 5 A set of states $q_1, q_2, \ldots, q_y$ forms a loop if $\delta_{\text{d}}(q_1, \sigma_1) = q_2, \delta_{\text{d}}(q_2, \sigma_2) = q_3, \ldots, \delta_{\text{d}}(q_l, \sigma_l) = q_1$ and y is finite.

Definition 6 A loop is called a cycle if $\nexists \delta_{\text{d}}(q_i, \sigma) = z$, where $i \in \{1, 2 \ldots, y\}, \sigma \notin \{\sigma_j\}_{j=1,2,\ldots,y}$, $z \notin \{q_j\}_{j=1,2,\ldots,y}$. Hence, in a cycle, no event in the set of events can have a transition from a state that is in the loop to a state that is not in the loop.

To illustrate this, in Fig. 1 there are two cycles, namely, A and B, and a loop, C. There will not be any branching from the cycles, whereas loops can have branches.

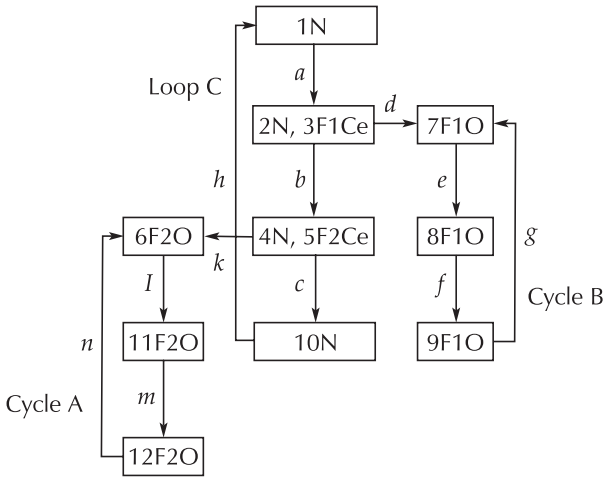


Fig. 1 Sample diagnoser.

Definition 7 A state q_i in a diagnoser is called an observation state if $\delta_{\text{d}}(q_{i-1}, o_{\sigma j}) = q_i$, where $o_\sigma j \in O_{\text{ef}}$, i.e., q_{i-1}can be any state in the diagnoser that transitions to q_i upon the occurrence of an observation. In Fig. 1, the states in cycle A are observation states and carry the label O.

Definition 8 A cycle is said to be an F_i observation cycle if it contains an observation state that has the F_iO label. In Fig. 1, cycle A is an observation cycle.

Definition 9 A state is said to be F_i observation uncertain if it contains labels l and l' such that $F_iO \in l$ and $F_iO \notin l'$. If a state has 13F2O and 12N as elements, then the state is an F2-uncertain observation, as we cannot determine whether the system is in the normal state or the F2 state.

Definition 10 A state is said to be F_i observation certain if all the labels in the state contain F_iO. If a state has 13F2O and 14F2O as elements, then it is F2-certain, as we can confirm that the system is in the fault F2 state.

Definition 11 A state is said to be ambiguous if it has different labels for the same state. If a state has elements 13F2O and 13N, then it is an ambiguous state.

Theorem 1 A system with language L is O-diagnosable if and only if each cycle that contains $F_i, i \in 1, 2, 3, \ldots, n$, in the diagnoser G_{d} is an F_i ob-

servation cycle.

Proof (Necessity) Proceed by contradiction. Suppose that L is O-diagnosable and assume that a fault F_i occurs and there exists a cycle of states, namely, $q_1, q_2, q_3, \ldots, q_m \in X_\mathrm{d}$, where X_d is the set of states in the diagnoser, such that it is not an F_i observation cycle. Hence, none of the states has the O label. It is possible that in the cycle, an observation never occurs in finite time. If an observation never occurs in finite time, it violates the diagnosability definition, which necessitates the presence of an observation after a faulty event in finite time, thereby resulting in a contradiction.

(Sufficiency) Assume that all the cycles that contain F_i in the diagnoser G_d are F_i observation cycles for $i \in 1, 2, \ldots, n$. Let $s \in L$ and $\delta(x_\mathrm{o}, s) = x$. Let t_1 be any string of observable events such that $\delta(x_\mathrm{o}, st_1) = x_1$. In the diagnoser, $q_1, q_2, q_3, \ldots, q_m \in X_\mathrm{d}$ is a cycle. By the assumption, the cycle is an observation cycle. Let $(x_1, l_1) \in q_1$ with $F_i \in l_1$ and $C_\mathrm{e} \in l_1$. This assumption can be made since the cycles are observation cycles, and each observation will be preceded by a command event. Since it is an observation cycle by assumption, there exists a string of observable events, which is denoted by t_2, such that $\|t_1 t_2\| \leqslant n_i < \infty$ and $\delta_\mathrm{d}(q_1, t_2) = q_2$, where $(x_2, l_2) \in q_2$ with $F_i \in l_2$ and $O \in l_2$. Assume that D is not implied; that is, $\exists w \in P^{-1}(P(st_1t_2)) \Rightarrow \sigma_{fi} \notin w$. Hence, under the assumption of the sensors being fault-free, $P(w) = P(st_1t_2)$ cannot have received a command and made observations for the i-th fault group in finite time, which contradicts the assumption of the F_i observation cycle. □

Theorem 2 Any system that is O-diagnosable satisfies the Sampath diagnosability condition.

Proof In the O-diagnosability condition C2 of Definition 3, set $t_1 t_2 = t$. Sampath's diagnosability condition follows immediately. □

Remark 3 According to Theorem 2, O-diagnosable systems (including observations) are diagnosable under the Sampath condition since the latter is a regularity issue. However, it is possible for a system (without the observations included) that is not diagnosable under the Sampath condition to become O-diagnosable when observations are included. This possibility is illustrated by example in the next section.

5.2 Performance of the O-diagnosability condition

5.2.1 Comparison in terms of the search complexity

For the O-diagnosability case, let $N_v = \sum_{p=1}^{v} 2^p$ be the power set of a string of v symbols, which includes unobservable and observable events and observations. The complexity of the search for an O-diagnosability case will be on the order of $\mathrm{O}(N_v)$, since for a specified suffix that contains the command input and the corresponding observation, the condition D must be checked for all possible prefixes. However, in the Sampath case, the suffix can be any of the elements of the power set N_v. Hence, the complexity is $(N_v^{N_v})$ in the Sampath case. In terms of the power set N_v of v symbols, the complexity of the search for O-diagnosability is linear, whereas in Sampath's case, it is exponential. The search for the O-diagnosability condition is less complex than under Sampath's diagnosability condition.

5.2.2 Complexity of diagnosability verification

Sampath [1] defines X_o to be the set of states in G that have an observable event. In addition, Δ is used to refer to the labels of fault types F_i : Normal and Ambiguous states. Sampath's diagnoser has states that correspond to the power set $2^{X_0 \times \Delta}$. Jiang et al. [13] define (x, f) for characterizing the states of G_o, where $x \in X_1$, which is the set of states in G that are reachable via observable transitions, and f belongs to the set of fault types, which is denoted by Σ_f. The states of G_o in [13] also correspond to the power set $2^{X_1 \times \Sigma_\mathrm{f}}$, following Sampath [1]. Jiang et al. [13] formulate a composition, namely, $G_\mathrm{o} \| G_\mathrm{o}$, and propose an algorithm that is claimed to have complexity $\mathrm{O}(|X|^4 \times 2^{4|\Sigma_\mathrm{f}|})$, where $X = \{x, f\}$. Since X_1 of Jiang et al. [13] and X_o of Sampath [1] are identical, $|X| = |2^{X_1 \times \Sigma_\mathrm{f}}|$. Hence, the algorithm of Jiang et al. [13] is of polynomial complexity in the power set of states X of G_o and exponential in the number of fault types. In contrast, the complexity of the proposed O-diagnosability verification method is linear in the power set of the number of alphabets, including events and observations, as discussed in Section 5.2.1 above.

5.2.3 Complexity in terms of the fault type

The complexity of the algorithm for verifying the diagnosability in terms of the fault type is exponential in O-diagnosability, as in the existing diagnosability case. However, for verification, if the diagnoser is run repeatedly for every fault type, the complexity will reduce to linear in the number of fault types. The algorithm for di-

agnosability verification using the diagnoser is presented in the appendix.

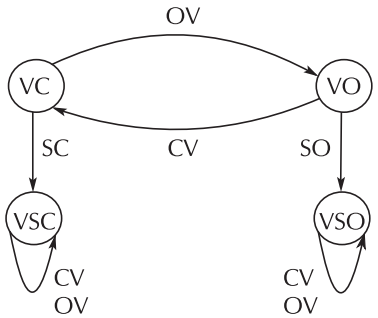
5.2.4 Bound on the number of events before reaching an F_i-certain observation state

The following provides a bound on the number of events that occur before a diagnosable system reaches an F_i-certain observation state. Let there be an F_i-uncertain observation cycle $q_1, q_2, q_3, \ldots, q_m \in G_d$. Hence, the states in the cycle have different labels. Let us assume, without loss of generality, that N and F_i are the labels of the states in the system. When a diagnosable system is in the N state and a fault occurs, the system will not loop indefinitely in this cycle; rather, the system will reach an F_i observation state within n_i events. We can obtain a bound on n_i for $\forall i \in 1, 2, \ldots, n$. For a fault of type i, the length of $t_1 t_2$ in st_1t_2, where s contains a fault, is bounded by $n_i \leqslant n_0 + c_i * n_0$, where n_0 is the longest unobservable string before an observable event occurs and $c_i = \sum_{q \in X_d;\, q \text{ is } F_i \text{ observation uncertain}} \#F_i$ states in q.

6 Fluid flow examples

We consider the fluid flow example with a pump, valve and controller that was used in Sampath's [2] paper. However, for the simplicity of the system model and the diagnoser, we assume that the pump is fault-free and the valve has faulty states. The valve has two normal states, namely, valve open (VO) and valve closed (VC), and two faulty states, namely, valve stuck closed (VSC) and valve stuck open (VSO). The pump has two normal states: pump on (PON) and pump off (POFF). The component models are illustrated in Fig. 2.

The state-based sensors that are considered are a flow sensor, which outputs PF (positive flow) and NF (no flow), and a pressure sensor, which outputs PP and NP. The diagnoser in the Sampath system is illustrated in Fig. 3. The system is not diagnosable, as the fault VSO and the normal states have the same trace of events.



(a)

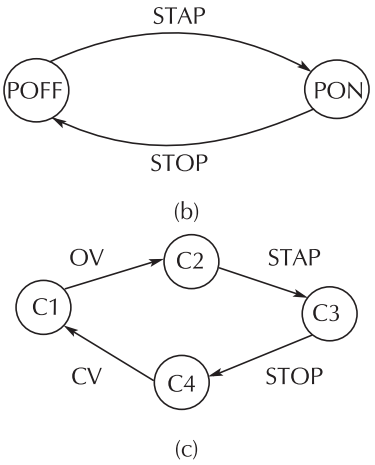


Fig. 2 Component models of a valve, pump and controller.

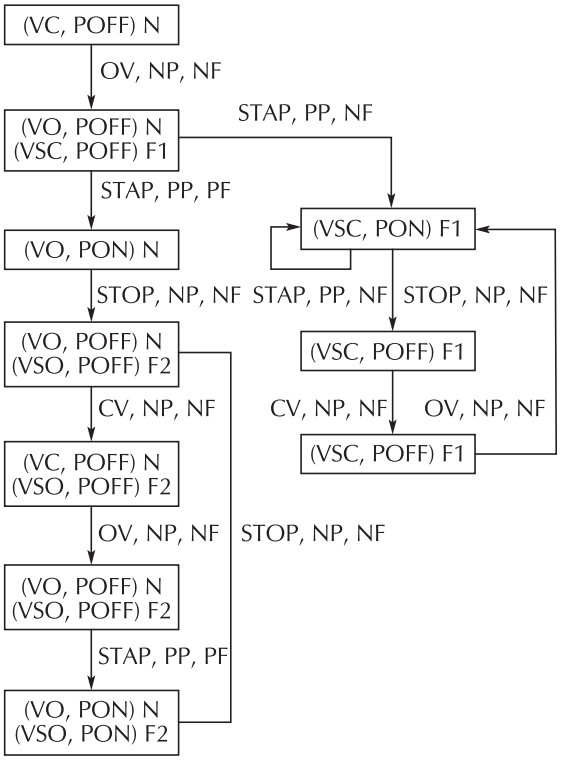


Fig. 3 Diagnoser.

In some cases, event-based sensors are available to indicate the status of valve operations. Such systems can be modeled using the proposed system. Suppose the system has valve sensors that indicate when the output valve is opened (VIO), when the valve is not opened (VNO), when the valve is closed (VIC) and when the valve is not closed (VNC). The model of the components is illustrated in Fig. 4. In the proposed system, the diagnosability is verified using conditions C1 and C2. To check condition C1, faults are mapped with

events/observations, as listed in Table 1.

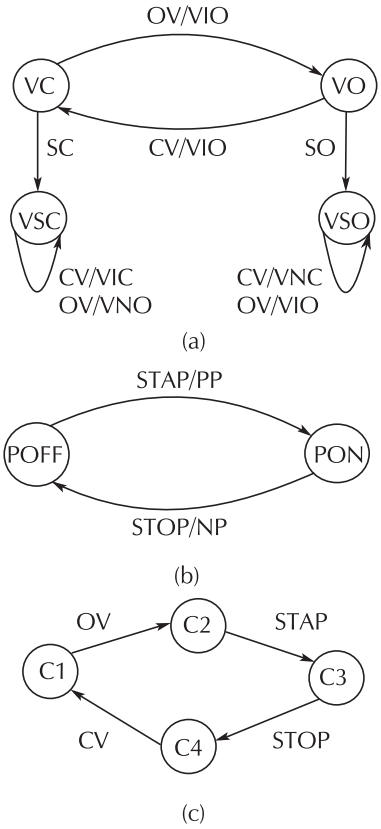


Fig. 4 Component models with observations.

Table 1 Mapping of faults and events/observations.

Fault	Event/Observation
VSC	OV/VNO
VSO	CV/VNC

Faults VSC and VSO are mapped; hence, they satisfy condition C1. We can check condition C2 to confirm the diagnosability. There will be many strings of events in the system. Of these strings, only one will be of normal operation; all others will contain faults. To diagnose a fault, the prefixes of all strings need not be checked. Only the prefixes of strings that contain a mapped event/observation must be checked for the presence of a fault. Therefore, to check, for example, whether VSC is O-diagnosable, we must check whether the prefix of each string that contains OV/VNO contains fault VSC. Similarly, for VSO, we must check the prefixes of strings that contain CV/VNC. OV/VNO is present in all the strings that contain fault VSC, and CV/VNC is present in all the strings that contain fault VSO. Hence, the system satisfies condition C2. The diagnoser is illustrated in Fig. 5.

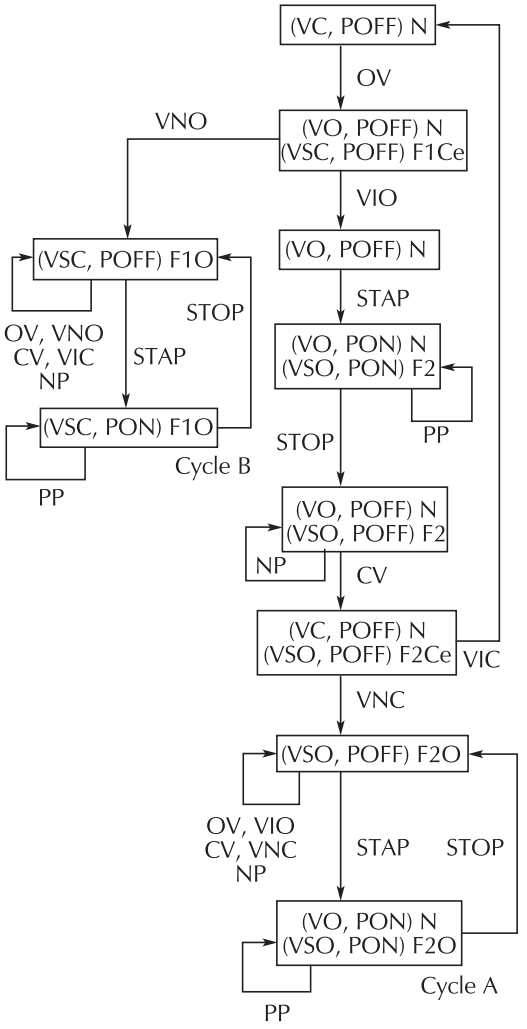


Fig. 5 Diagnoser in the proposed system.

The cycles A and B are observation cycles that confirm the O-diagnosability of the system.

7 Conclusions

A new diagnosability condition (which is called O-diagnosability) of a DES model that is based on event-based outputs (which are called observations) for diagnosis is proposed in this paper. A diagnoser is defined based on the proposed O-diagnosability condition. A necessary and sufficient condition for a system to be O-diagnosable is derived. The search for O-diagnosability verification, being a special case, is shown to be linear in the power set of all the events in the system, compared to exponential complexity in the power set in the case of the existing diagnosability. In addition, an upper bound on the number of events that occur before a diagnosable system satisfies the O-diagnosability condition

is presented. A system that is not diagnosable according to the existing diagnosability condition may become O-diagnosable with the inclusion of observations.

The presence of observations that correspond to faults upon a command event opens up the possibility of the direct observation of faults at the component level without having to deduce the faults based on a combination of event and sensor outputs in various components by employing a synchronous composition of component FSMs. We plan to extend the system diagnosability to component-level diagnosability using event-based observations in our future work.

References

[1] M. Sampath, R. Sengupta, S. Lafortune, et al. Diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 1995, 40(9): 1555 – 1575.

[2] M. Sampath, R. Sengupta, S. Lafortune, et al. Failure diagnosis using discrete-event models. *IEEE Transactions on Control Systems Technology*, 1996, 4(2): 105 – 124.

[3] F. Lin. Diagnosability of discrete event systems and its application. *Discrete Event Dynamic Systems*, 1994. 4(2): 197 – 212.

[4] S. H. Zad, R. H. Kwong, W. M. Wonham. Fault diagnosis in discrete-event systems: framework and model reduction. *IEEE Transactions on Automatic Control*, 2003, 48(7): 1199 – 1212.

[5] P. Baroni, G. Lamperti, P. Pogliano, et al. Diagnosis of large active systems. *Artificial Intelligence*, 1999, 110(1): 135 – 183.

[6] X. Su, M. Zanella, A. Grastien. Diagnosability of discrete-event systems with uncertain observations. *Journal of Software*, 2017, 28(5): 1091 – 1106.

[7] M. Chang, W. Dong, Y. Ji, et al. On fault predictability in stochastic discrete event systems. *Asian Journal of Control*, 2013, 15(5): 1458 – 1467.

[8] S. Bhattacharyya, R. Kumar, Z. Huang. A discrete event systems approach to network fault management: detection and diagnosis of faults. *Asian Journal of Control*, 2011, 13(4): 471 – 479.

[9] E. Fabre, L. Helouet, E. Lefaucheux, et al . Diagnosability of repairable faults. *Proceedings of the 13th International Workshop on Discrete Event Systems* (*WODES*), Xi'an: Springer, 2016: 183 – 213.

[10] A. Boussif, B. Liu, M. Ghazel. A twin-plant based approach for diagnosability analysis of intermittent failures. *Proceedings of the 13th International Workshop on Discrete Event Systems*, Xi'an: Springer, 2016: 237 – 244.

[11] M. Agarwal, S. Biswas, S. Nandi. I^2-diagnosability framework for detection of advanced stealth man in the middle attack in Wi-Fi networks. *Proceedings of the 23rd Mediterranean Conference on Control and Automation*, Torremolinos, Spain: IEEE, 2015: 349 – 356.

[12] C. Cassandras, S. G. Lafortune. *Introduction to Discrete Event Systems*. 2nd ed. New York: Springer, 2009.

[13] S. Jiang, Z. Huang, V. Chandra, et al. A polynomial algorithm for testing diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 2001, 46(8): 1318 – 1321.

[14] S. Reshmila, R. Devanathan. Robust diagnosis of power system failures using discrete event system approach. *IEEE Region 10 Conference*, Macao: IEEE, 2015: DOI 10.1109/TENCON.2015.7373116.

[15] S. Reshmila, R. Devanathan. Modeling a system using observations in discrete event system for failure diagnosis. *Proceedings of the IEEE Recent Advances in Intelligent Computational Systems*, Thiruvananthapuram: IEEE, 2015: 280 – 284.

[16] S. Reshmila, R. Devanathan. Modeling and robust diagnosis of power system protection failures using observations in discrete event system. *Indian Control Conference*, Hyderabad: IEEE, 2016: 170 – 175.

[17] S. Reshmila, R. Devanathan. Diagnosis of power system failures using observer based discrete event system. *IEEE First International Conference on Control, Measurement and Instrumentation*, Kolkatta: IEEE, 2016: 131 – 135.

Appendix

Algorithm for diagnosability verification using the diagnoser

Step 1 Compute diagnoser G_{d} as follows:

1) Define the set of labels as $L = \{N\} \cup \{F_1, F_2, \ldots, F_n\} \cup \{C_{\mathrm{e}}, O\}$.

2) Define the initial state with label N.

3) Identify the subsequent states that are reachable from x_0 using the transition function δ_{d}.

4) Find the labels of the subsequent states using the label propagation function, namely, LP.

Step 2 Verify the diagnosability of the system:

1) Identify the F_i cycles in G_{d}.

2) Check the states $x'_{\mathrm{d}} = (x_i F_i l, x_j F_j l)$ in the cycles.

3) Check whether $l = O$ and $F_i = F_j$ in all the states in the cycles.

a) If true, the system is diagnosable.

b) If false, the system is not diagnosable.

S. RESHMILA received the B.Tech degree in Electrical and Electronics Engineering from the Mahatma Gandhi University in 1998 and the M.Tech degree in Embedded Systems from the Calicut University in 2007. She is currently a Ph.D. candidate in Electrical Engineering at Hindustan Institute of Technology and Science, Chennai. Her research interests are in model-based fault diagnosis and discrete event system. E-mail: reshmilas@yahoo.com.

Devanathan RAJAGOPALAN received his Ph.D. and M.Sc. (Eng.) from Queen's University, Kingston, Ontario, Canada and his B.E. and M.E. degrees from Indian Institute of Science, Bangalore, India. Dr. Devanathan has taught at Nanyang Technological University (NTU), Singapore for over two decades. He has published over 160 papers in international and national conference proceedings and journals, and has received awards from IEEE Education Society and NTU. He has chaired and co-chaired international conferences organized by IEEE as well as NTU. Currently he is Professor Emeritus in Electrical & Electronic Engineering in Hindustan Institute of Technology and Science, Chennai, India. His interests include control systems, nonlinear systems, robotics, image processing, discrete event systems and quantitative linguistics. He is a life senior member of IEEE. E-mail: devanathanr@hindustanuniv.ac.in.