This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2912712, IEEE Transactions on Automatic Control

Fault Diagnosis of Discrete Event Systems under Unknown Initial Conditions

Alejandro White, Ali Karimoddini, and Rong Su

This paper proposes a novel diagnosis technique for Discrete Event Systems (DESs) plant models. The developed diagnosis tool, so called diagnoser, is able to detect and isolate the occurrence of system's faults without the knowledge of the system's past behavior. This allows the diagnoser to asynchronously begin its diagnosis of a system's behavior at any time instance of system operation (including post fault occurrences); consequently removing the generally required synchronous initialization between a diagnoser and the system under diagnosis. The necessary and sufficient conditions are derived for the diagnosability of a given DES plant under this asynchronous situation. Several examples are provided to illustrate the details of the proposed diagnosis framework.

***Index Terms*—Fault Diagnosis; Discrete Event Systems; Automata; asynchronous Diagnosis; Uncertainty**

I. INTRODUCTION

DESPITE all efforts, almost all engineered systems are faulty or become faulty over time. This requires the development of systematic approaches that detect and compensate the faulty behavior(s) of a system [1]–[4]. A fault can be defined as a malfunction in system's component(s) (actuators, sensors, processors, mechanical parts, software, etc) that results in unacceptable or degraded system performance, and/or system instability. In this paper, we study the fault diagnosis problem within the Discrete Event Systems (DESs) framework [5]–[8] due to the fact that DES models naturally capture faults as abrupt changes (events) in the system, which facilitates the analysis of faulty behaviors of the system. There are different techniques for fault diagnosis of DESs. In [9], an off-line diagnosis technique was introduced. Later, in [10], an automated online diagnosis technique was developed for DESs and the diagnosability of DES systems. Diagnosability is a concept that allows the diagnoser (man or man-made) to determine if all system faults may indeed be detected within a finite number of post-fault transitions in the system. The approach later was extended to decentralized [11], [12], modular/distributed [13], [14], robust and safe [15]–[17] diagnosis structures. A comprehensive review of fault diagnosis techniques for discrete event systems can be found in [18]. In all of the aforementioned techniques, it is required to initialize and run the diagnoser synchronously with the plant. This allows the diagnoser to diagnose faults based on a rich set of information including both pre- and post-fault behaviours in the system. However, the requirement for synchronous initialization of the diagnoser and the system under diagnosis would not be practically easy to meet.

Therefore, to address this problem, this paper proposes a systematic and analytical approach to construct a diagnoser that can be asynchronously turned on at any time, even after the occurrence of a fault. The problem is that unlike conventional diagnosis techniques, the asynchronous diagnoser does not observe the past history of the system's event occurrences before the activation of the diagnoser, leaving the diagnoser with the challenge of diagnosing faults using only the future behaviors of the system, observed after the activation of the diagnoser. In contrast to existing methods, where the initial state of the system and correspondingly the initial state of the diagnoser are generally assumed to be non-faulty, upon its initialization, the asynchronous diagnoser is no longer able to assume that the current state of the system is normal. In [19], an asynchronous state-based diagnoser is introduced to detect permanent faults in a DES system. Compared to [19], this paper discusses the fault diagnosis problem within the context of event-based DES paradigm. In the proposed approach, unlike the state-based framework, the system under diagnosis is not restricted to be partitioned into disjoint and disconnected set of normal and faulty states. More importantly, compared to [19] and our preliminary results on developing an asynchronous diagnoser in [20], this paper goes beyond the development of the diagnoser by introducing a formal definition for asynchronous diagnosis and diagnosability, providing the necessary and sufficient conditions for asynchronous diagnosability, and comparing synchronous and asynchronous diagnosis and diagnosability.

The rest of the paper is organized as follows. Section II, provides the necessary background and notations for DES modeling of the original system and formulates the asynchronous fault diagnosis problem. In Section III, an algorithm for constructing the asynchronous diagnoser is presented, accompanied by an illustrative example detailing the steps of the proposed algorithm. In Section IV, we formally define asynchronous diagnosability and provide the necessary and sufficient conditions to check the asynchronous diagnosability. The proposed asynchronous diagnosis technique is compared with traditionally synchronous diagnosis techniques in Section V. Section VI concludes the paper.

A. White and A. Karimoddini are with the Department of Electrical and Computer Engineering, North Carolina Agricultural and Technical State University, Greensboro, NC 27411 USA, and S. Rong is with the School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798.

Corresponding author: A. Karimoddini. Address: 1601 East Market Street, Department of Electrical and Computer Engineering North Carolina A&T State University Greensboro, NC, US 27411. Email: akarimod@ncat.edu (Tel: +13362853847).

II. Problem Formulation

Consider a plant which is modeled as a non-deterministic finite-state Discrete-Event System (DES) represented by a four-tuple,

$$G = (X, \Sigma, \delta, x_0) \tag{1}$$

where X is the state space of the system, $x_0 \in X$ is the system's initial state, Σ is the finite set of events, and $\delta : X \times \Sigma \to 2^X$ (2^X is the power set of X) is the state transition relation; a partial relation that determines all feasible system state transitions caused by events.

The system's event set Σ can be partitioned into two disjoint sets: the observable event set (Σ_o) and the unobservable event set (Σ_u). A sequence of events is called a *string* or trace. For a string t, $|t|$ indicates its length. With the abuse of notation, we use $e \in s$ to say that the event e belongs to the string s, if e is one of the events forming the string s. Σ^* (the Kleene closure of Σ) is the set of all possible finite strings over the set Σ, including the zero-length string ε. A set of strings form a language. The concatenation of two strings s_1 and s_2 is shown by $s_1.s_2$. Extending the transition rule, δ, to the strings, it can be recursively defined as $\delta(x, s.e) = \bigcup\limits_{y \in \delta(x,s)} \delta(y, e)$.

The set of strings that can be generated by G from the state x is $\mathcal{L}_G(x) = \{s \in \Sigma^* \mid \delta(x, s) \neq \emptyset\}$. The language of the plant, $\mathcal{L}_G$, is the set of all sequences of strings that can be generated by the automaton G from the state x_0, which can be captured by $\mathcal{L}_G(x_0)$. The language $\mathcal{L}_{G/s} = \{t \in \Sigma^* \mid s.t \in \mathcal{L}_G\}$ is the set of all traces in $\mathcal{L}_G$ that occur immediately following $s \in \mathcal{L}_G$. The extension closure of the language $\mathcal{L}_G$, denoted by $ext(\mathcal{L}_G)$, can be defined as $ext(\mathcal{L}_G) := \{v \in \Sigma^* \mid \exists u \in \mathcal{L}_G : uv \in \mathcal{L}_G\}$.

We define unobservable reach $UR(x) = \{y \in X \mid \exists u \in \Sigma_u^*,\ y \in \delta(x, u)\}$, as the set of all of the system's states (with the inclusion of x itself) that are reachable from state x via strings solely consisting of unobservable events. Also, $UE(s, x) = \{s.t \mid t \in \Sigma_u^* \ and \ s.t \in \mathcal{L}_G(x)\}$ specifies the set of all unobservable extensions of s concatenated with the string s and generated from the state x.

The presented system model encompasses the system's normal and failed behavior, where the set of system faults, Σ_f, is a subset of the system's event set, Σ. Similar to [10], we partition the system's faults, Σ_f, as the union of m different types Σ_{f_1}, Σ_{f_2}, ..., Σ_{f_m}. We also consider the worst case scenario that faults are unobservable $\Sigma_f \subseteq \Sigma_u$. However, the paper's derivations are valid for the case that faults are observable as well.

Definition 1: We refer to a string $t \in \mathcal{L}_G$ as an "*F_i-faulty*" string if there exists an event $f \in \Sigma_{f_i}$, such that $f \in t$. A string $t \in \mathcal{L}_G$ is called "*non-F_i-faulty*" if for all $f \in \Sigma_{f_i}$, we have $f \notin t$. Finally, a string $t \in \mathcal{L}_G$ is referred to as a "*normal*" string if for all $f \in \Sigma_{f_i}$ and for all $i = 1, ..., m$, $f \notin t$.

Definition 2: State $x \in X$ in G is F_i-faulty if it is reachable by an F_i-faulty string, i.e., $\exists t \in \mathcal{L}_G$ and $\exists f \in \Sigma_{f_i}$ such that $x \in \delta(x_0, t)$ and $f \in t$. Similarly, state $x \in X$ is non-F_i-faulty, if it is reachable by a non-F_i-faulty string, i.e., $\exists s \in \mathcal{L}_G$ so that $\forall f \in \Sigma_{f_i}$, $f \notin s$ and $x \in \delta(x_0, s)$.

Remark 1: Since a state may be reached by different normal or faulty strings, a state $x \in X$ in G can be both F_i-faulty and non-F_i-faulty, if it is reachable by both a F_i-faulty string and a non-F_i-faulty string.

Our purpose is to diagnose the occurrence of (unobservable) faults from the observable behavior of the system. The system's observable behavior can be described by the natural projection of the system's language to the observable event set of the system. The natural projection onto observable event set, $P : \Sigma^* \to \Sigma_o^*$, can be defined as follows:

- $P(\varepsilon) = \varepsilon$,
- $P(e) = e$, if $e \in \Sigma_o$,
- $P(e) = \varepsilon$, if $e \notin \Sigma_o$,
- $P(s.e) = P(s)P(e)$, for $s \in \Sigma^*$ and $e \in \Sigma$.

This definition can be further extended to a language $\mathcal{L}_1$ as $P(\mathcal{L}_1) = \{P(s) \mid s \in \mathcal{L}_1\}$. The inverse projection of a string $w \in \Sigma_o^*$ into $\mathcal{L}_1 \subseteq \Sigma^*$ is $P_{\mathcal{L}_1}^{-1}(w) = \{s \in \mathcal{L}_1 \mid P(s) = w\}$, and the inverse projection of a language $\mathcal{L}_2$ into $\mathcal{L}_1$ is $P_{\mathcal{L}_1}^{-1}(\mathcal{L}_2) = \bigcup\limits_{s \in \mathcal{L}_2} P_{\mathcal{L}_1}^{-1}(s)$.

To analyze the faulty behaviors of a plant G, we assume that the model of the system containing both faulty and normal behaviors is given. Further, we assume that the language of the plant, $\mathcal{L}_G$, is live, i.e., $\forall x \in X, \exists \sigma \in \Sigma$ such that $\delta(x, \sigma)$ is defined. This ensures that after the occurrence of a fault there is ample time provided to monitor the system's behavior, and diagnose the fault occurrence. Furthermore, we assume that the lengths of unobservable strings in $\mathcal{L}_G$ are bounded by n_o, otherwise, the plant G may become trapped in a cycle of unobservable events, in turn making the diagnosis impossible. Given this information, fault diagnosis is the art of distinctively characterizing the system's behavior in order to detect, identify, and locate fault occurrences solely based upon external observations of the system. This can be formally formulated as follows:

Problem 1: Assume that a discrete event system G has been running and has generated a string s. Not knowing the past history of the system G and its generated string s, start the diagnosis process, i.e., for any successive string $t \in \mathcal{L}_{G/s}$, where t occurs upon starting the diagnosis process, only from the observation $P(t)$, determine if $\exists f \in \Sigma_f$ such that $f \in s.t$. If yes, identify the type of fault, Σ_{f_i}, where $f \in \Sigma_{f_i}$, and locate the fault by finding the system state $x \in X$ subsequently reached upon fault occurrence.

III. Constructing the Diagnoser

To address the fault diagnosis problem described in Problem 1, we propose to develop an asynchronous diagnoser, which provides diagnostics by extracting information from the original system's observable events in order to estimate the original system's current state location and current condition (faulty or non-faulty). The proposed asynchronous diagnoser is a deterministic finite-state DES represented by a four-tuple:

$$G_d = (Q_d, \Sigma_d, \delta_d, q_0) \tag{2}$$

where $Q_d \subseteq 2^{X \times L}$ is the state space of the diagnoser, $\Sigma_d = \Sigma_o$ is the event set, δ_d is the state transition rule, q_0 is the diagnoser's initial state, and L is the diagnostic label

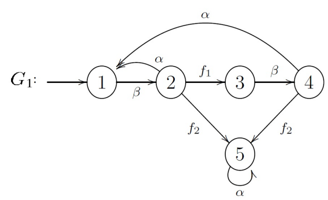

Fig. 1: The DES model of a UAV involved in a search mission in which the events α and β are for "*traveling back to the hangar*" and "*searching for a target*." The fault events f_1 and f_2 are for "*loss of the communication link*" and "*fuel leakage/low*."

set. The diagnostic labels are defined as $L = \{N\} \cup 2^F$, $F = \{F_1, F_2, \dots, F_m\}$, where F_i is a label representing the faults in Σ_{f_i}, $i = 1, ..., m$, and N is a label representing the condition of normal system operation. The diagnoser's states are in the form of $q_d = \{(x_1, \ell_1), \dots, (x_k, \ell_k)\}$, where $x_i \in X$ and $\ell_i \subseteq L$. In fact, the states of the diagnoser are sets of ordered pairs, (x_i, ℓ_i), consisting of the estimations of the original system state, x_i, and their corresponding fault indicator label sets, ℓ_i. From these ordered pairs, (x_i, ℓ_i), we can detect the occurrence of the faults and isolate fault types, as ℓ_i carries the information about the occurred faults, and x_i indicates the current state (location) of the system. Next, we will discuss the procedure to find the states of the diagnoser and the transition rules.

Upon the occurrence of each observable event, the diagnoser will update its estimations of the state of the original system. In addition, the diagnoser will append its estimation of the system's condition to the estimated states of the system in the form of a label set $\ell \subseteq L$. Assuming that the current condition of the system is captured by ℓ, followed by the occurrence of a string $t \in \Sigma^*$, the update of the label set ℓ to a new label set ℓ' is carried out by the *Label Updating Function*, $\nabla : L \times \Sigma^* \to L$, where $\ell' = \nabla(\ell, t) =:$

$$\begin{cases} \{N\}, \; if \; \ell = \{N\} \;\; and \; \forall f \in \Sigma_f, \; f \notin t, \\ \{F_i \in F | F_i \in \ell \; or \; \exists f \in \Sigma_{fi}, \; f \in t\}, \; Otherwise \end{cases} \tag{3}$$

The asynchronous diagnoser may be activated at any time instance, independent of the original system's operation. For this purpose, the asynchronous diagnoser starts wide and narrows down its estimate of the original system's state and condition as it receives more information from its observations. Because the diagnoser is not synchronously activated with respect to the original system, upon activation, the diagnoser is completely unknowing of the original system's current state and condition. Therefore, the diagnoser's initial state is as follows:

$$q_0 \;:=\; \{(y, \nabla(\{N\}, t)) | t \;\in\; \mathcal{L}_G(x_0), y \;\in\; \delta(x_0, t)\} \tag{4}$$

Following activation of the diagnoser, observance of $e \in \Sigma_o$ will cause the diagnoser to update its estimation of the original system's state and condition. Starting from q_0, we may now define the diagnoser's set of states, Q_d, and construct its transition relation, $\delta_d : Q_d \times \Sigma_o \to Q_d$, as follows:

$$\delta_d(q, e) = \{(y, \nabla(\ell, t)) | (x, \ell) \in q, t \in UE(e, x), y \in \delta(x, t)\} \tag{5}$$

Algorithm 1 summarizes the diagnoser construction process. Assuming that the original system, G, is initially normal (non-faulty), the algorithm starts with $q_0 = \{(x_0, \{N\})\}$ as the initial state of the diagnoser, and then, extends q_0 to $x \in UR(x_0)$ by $q_0 = q_0 \cup \{(x, \ell) | x \in \delta(x_0, u), \; u \in \Sigma_u^*, \; \ell = \nabla(\{N\}, u)\}$, and will continue the process by searching over all other possible strings in $\mathcal{L}(G)$.

Algorithm 1 Constructing an Asynchronous Diagnoser

<u>Initialization:</u>
$q_0 := \{(x_0, \{N\})\}$;
<u>Step 1: Constructing q_0</u>
$q_0 \;:=\; q_0 \bigcup \{(x, \ell) | x \;\in\; \delta(x_0, u), \;\; u \;\in\; \Sigma_u^*, \;\; \ell \;=\; \nabla(\{N\}, u)\}$;
repeat
  for $(x, \ell) \in q_0$ and $e \in \Sigma_o$ **do**
    if $\exists t \in UE(e, x)$ such that $\exists y \in \delta(x, e)$ and $(y, \nabla(\ell, t)) \notin q_0$ **then**
      $q_0 = q_0 \cup \{(y, \nabla(\ell, t))\}$;
    end if
  end for
until There is no new pair (x, l) in q_o.
<u>Step 2: Constructing Q_d</u>
$Q_d := q_0$;
repeat
  for $q \in Q_d$ and $e \in \Sigma_o$ **do**
    if $\delta_d(q, e)$ is defined and $\delta_d(q, e) \notin Q_d$ **then**
      Add $\delta_d(q, e)$ to Q_d;
    end if
  end for
until There is no new state $\delta_d(q, e)$ for all $e \in \Sigma_o$

In Step 1, the algorithm constructs q_0, and in Step 2, it constructs the remaining accessible diagnoser states $q \in Q_d$. The following example illustrates the implementation of the algorithm.

Example 1: Consider an unmanned aerial vehicle (UAV) involved in a search mission to find a particular target. A simple model for this search mission is the automaton G_1, shown in Fig. 1, with $\Sigma = \{\alpha, \beta, f_1, f_2\}$, $\Sigma_o = \{\alpha, \beta\}$, $\Sigma_u = \{f_1, f_2\}$, and $\Sigma_f = \{f_1, f_2\}$, $\Sigma_{f_1} = \{f_1\}$, and $\Sigma_{f_2} = \{f_2\}$. In this model, the event β is for "searching for a target," the event α is for "traveling back to the hangar," f_1 is a fault event that is activated in case of "loss of the communication link" and the fault event f_2 is for "fuel leakage/low". In case the UAV loses the communication link, it continues searching around (to possibly get connected again), and if there is a fuel leakage or low fuel level, the UAV quickly returns to the hangar.

Following Step 1 of Algorithm 1, we will have $q_0 = \{(1, \{N\}), (1, \{F_1\}), (2, \{N\}), (2, \{F_1\}), (3, \{F_1\}), (4, \{F_1\}), (5, \{F_2\}), (5, \{F_1, F_2\})\}$. Following Step 2 of the algorithm, other states of the diagnoser and the transition function δ_d can be found as shown in Fig. 2. In this figure,

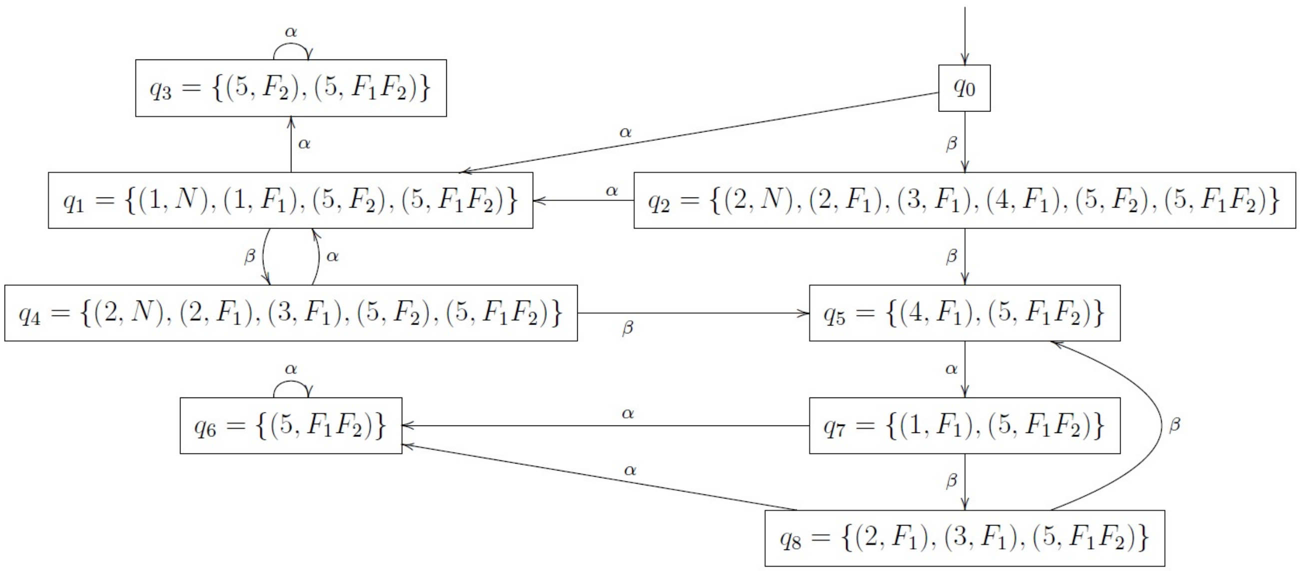

Fig. 2: The constructed diagnoser G_{d1} for the plant G_1 given in Fig. 1

instead of the pair $(5, \{F_1, F_2\})$, we have simply used $(5, F_1F_2)$. Similar notation is used for other pairs and other diagnosers' figures in this paper.

Example 2: For the plant G_1, imagine that f_2 has occurred and the plant G_1 is in state 5. Then, we turn on the diagnoser, and the diagnoser starts from q_0. When the event $\alpha \in \Sigma_o$ happens in the plant G_1, the diagnoser switches to $q_1 = \{(1, \{N\}),(1, \{F_1\}), (5, \{F_2\}), (5, \{F_1, F_2\})\}$, which is an F_2-uncertain state. But if we wait for another transition, the event $\alpha \in \Sigma_o$ happens again in the plant G_1, resulting in the diagnoser transiting to q_3, which is an F_2-certain state, implying that the fault of type F_2 has occurred during the system's operation.

The following two lemmas provide some properties of the developed diagnoser, which will be used in future derivations.

Lemma 1: If $\delta_d(q_k, e_k) = q_{k+1}$ then for any pair $(x_{k+1}, \ell_{k+1}) \in q_{k+1}$, there must exist at least one pair (x_k, ℓ_k) in q_k and $t_k \in P^{-1}_{ext(\mathcal{L}_G)}(e_k)$ such that $x_{k+1} \in \delta(x_k, t_k)$.

Proof: From Equation 5, we know that $q_{k+1} = \delta_d(q_k, e_k) = \bigcup\limits_{\substack{(x,\ell)\in q_k \\ t\in UE(e_k,x)}} \{(\delta(x, t), \nabla(\ell, t))\}$. Therefore, for any $(x_{k+1}, \ell_{k+1}) \in q_{k+1}$, there exists at least one pair (x_k, ℓ_k) in q_k such that $x_{k+1} \in \delta(x_k, t_k)$, where $t_k \in P^{-1}_{ext(\mathcal{L}_G)}(e_k)$. ■

Lemma 2: Consider $\delta_d(q_k, e_k) = q_{k+1}$, where (x_k, ℓ_k) in q_k, (x_{k+1}, ℓ_{k+1}) in q_{k+1}, $x_{k+1} \in \delta(x_k, t_k)$, $t_k \in P^{-1}_{ext(\mathcal{L}_G)}(e_k)$. If $F_i \notin \ell_{k+1}$, then $F_i \notin \ell_k$.

Proof: From Equation 3, we know that if $F_i \in \ell_k$, it will be propagated and $F_i \in \ell_{k+1} = \nabla(\ell, t)$. Conversely, If $F_i \notin \ell_{k+1}$, then $F_i \notin \ell_k$ ■

IV. Asynchronous Diagnosability

Once the diagnoser is constructed, an important question is: upon the diagnoser's asynchronous activation, is the diagnoser capable of definitively diagnosing system faults that occur pre and/or post diagnoser activation. This can be achieved if the system under diagnosis is asynchronously diagnosable, which can be formally defined as follows:

Definition 3: (F_i-Asynchronous Diagnosability) The DES system G with the live language $\mathcal{L}_G$, is said to be F_i-Asynchronously diagnosable with respect to the fault type F_i and the natural projection P, if for all $s \in \mathcal{L}_G$, $f \in \Sigma_{f_i}$, $f \in s$, there exist an upper bound $n_i \in \mathbb{N}$, such that for any string $t \in \mathcal{L}_{G/s}$ with $|t| \geq n_i$, the following condition holds:

$$\{\forall uv \in \mathcal{L}_G, v \in P^{-1}_{ext(\mathcal{L}_G)}(P(t))\} \Rightarrow f' \in uv, f' \in \Sigma_{f_i} \quad (6)$$

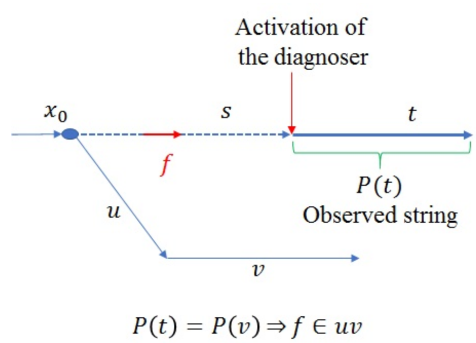

Fig. 3: Illustration of asynchronous diagnosability.

Definition 4: (Asynchronous Diagnosability) The plant G with the live language $\mathcal{L}_G$, is said to be asynchronously diagnosable with respect to the fault set Σ_f and the natural projection P, if it is F_i-asynchronously diagnosable with respect to all fault types F_i, $i = 1, ..., m$.

To graphically illustrate the concept of asynchronous diagnosability, Figure 3 shows an example of a faulty string s, which is succeeded by a string t. The string $P(t)$ represents all observations after the activation of the diagnoser. If there is any other string v in $\mathcal{L}_G$ with the same observations, $P(t) = P(v)$, then v and its predecessor string u should contain a fault event ($f \in u.v$). Otherwise, we should increase the number of observations (extend t to t' and v to v', where $t \leq t'$ and $v \leq v'$), until either $f \in uv'$ or $P(t') \neq P(v')$. If by finite number of observations we cannot distinguish whether a fault has occurred from the post-activation observations, the system becomes asynchronously undiagnosabile.

Remark 2: Unlike synchronous diagnosis [10] where all the system behavioral information (pre- and post-fault occurrence) is available to the diagnoser, in asynchronous diagnosis, the diagnoser only has access to the system behavioral information that occurs after the diagnoser's activation. This can be observed in Definitions 3 and 4 where the asynchronous diagnosis requires faulty strings to be distinguishable only from their extension closure.

Although Definitions 3 and 4 describe the asynchronous diagnosability, it is difficult to check the diagnosability condition given in (7) over all faulty strings in $\mathcal{L}_G$. Theorem 1, therefore, will derive the necessary and sufficient conditions to indirectly check the asynchronous diagnosability based on the structure of the diagnoser. For this purpose, we need to first provide a few definitions:

Definition 5: Consider $q = \{(x_1, \ell_1), \ldots, (x_M, \ell_M)\} \in Q_d$. Then, q is said to be

- Normal if $\ell_k = \{N\}$ for all $k = 1, ..., M$.
- F_i-certain if $F_i \in \ell_k$ for all $k = 1, ..., M$.
- F_i-uncertain if $\exists n, m$ such that $F_i \in \ell_n$, but $F_i \notin \ell_m$.

Definition 6: A cycle in G_d is called F_i-certain if all of its states are F_i-certain; otherwise, it is called a non-F_i-certain cycle. A cycle of F_i-uncertain states in G_d is called an F_i-uncertain cycle.

Definition 7: (F_i-indeterminate cycle) A set of F_i-uncertain states q_1, q_2, ...,$q_n \in Q_d$ forms an F_i-indeterminate cycle if and only if

- The states q_1, q_2, ..., and q_n form a cycle in G_d, i.e., $\delta_d(q_k, e_k) = q_{k+1}$, for $k = 1, \ldots, n-1$, $\delta_d(q_n, e_n) = q_1$, and $e_k \in \Sigma_o$, $k = 1, \ldots, n$.
- The cycle q_1, q_2, ...,q_n in G_d can be inversely projected back to at least one cycle of non-F_i-faulty states and one cycle of F_i-faulty states in the original system G, i.e., each state of the cycle, q_k, contains (x_k, ℓ_k) and (x'_k, ℓ'_k) so that
  - $F_i \notin \ell_k$ and $F_i \in \ell'_k$.
  - x_1, x_2,..., x_n form a cycle in G so that $x_{k+1} \in \delta(x_k, t_k)$, $k = 1, \ldots, n-1$, and $x_1 \in \delta(x_n, t_n)$, where $t_k \in P^{-1}_{ext(\mathcal{L}_G)}(e_k)$ for $k = 1, \ldots, n$.
  - x'_1, x'_2,..., x'_n form a cycle in G so that $x'_{k+1} \in \delta(x'_k, t'_k)$, $k = 1, \ldots, n-1$, and $x'_1 \in \delta(x'_n, t'_n)$, where $t'_k \in P^{-1}_{ext(\mathcal{L}_G)}(e_k)$ for $k = 1, \ldots, n$.

In other words, an F_i-indeterminate cycle is a cycle of F_i-uncertain states in the diagnoser, of which there exists a corresponding cycle of non-F_i-faulty states, and a corresponding cycle of F_i-faulty states in the original system.

Upon entering an F_i-indeterminate cycle, the diagnoser will be unable to definitively detect and isolate faults of type F_i, as it will be discussed in Theorem 1. Before that, we need the following lemma.

Lemma 3: Consider the diagnoser G_d constructed for a plant G with a live language $\mathcal{L}_G$, that has a cycle of F_i-uncertain states q_1, q_2, ..., $q_n \in Q_d$ such that $\delta_d(q_k, e_k) = q_{k+1}$, for $k = 1, \ldots, n-1$, $\delta_d(q_n, e_n) = q_1$, $e_k \in \Sigma_o$, $k = 1, \ldots, n$. Assume that this cycle is not an F_i-indeterminate cycle. If before or after entering this cycle of F_i-uncertain states a fault of type F_i occurs, then the diagnoser will transit out of the cycle of F_i-uncertain states after a finite number of transitions.

Proof: The states q_1, q_2, ...,q_n are F_i-uncertain, therefore each state q_k contains at least two pairs of (x_k, ℓ_k) and (x'_k, ℓ'_k) so that $F_i \notin \ell_k$ and $F_i \in \ell'_k$, for $k = 1, \ldots, n$. Since q_k, $k = 1, \ldots, n$, do not form an F_i-indeterminate cycle, according to Definition 7, either there does not exist a cycle of non-F_i-faulty pairs (x_k, ℓ_k) (the corresponding non-F_i-faulty states x_k do not form a cycle in G), or there does not exist a cycle of F_i-faulty pairs (x'_k, ℓ'_k) (the corresponding F_i-faulty states x'_k do not form a cycle in G). The former case is impossible to happen. This can be proven by a backward reachability induction and by applying Lemmas 1 and 2. Now, consider the following two possible situations:

Case 1 (A fault $f \in \Sigma_{f_i}$ occurs before diagnoser G_d enters the cycle of F_i-uncertain states): If after the occurrence of the fault $f \in \Sigma_{f_i}$, the diagnoser enters this cycle of F_i-uncertain states by reaching the state q_k of the cycle, one of the states x'_k in G will be reached, which corresponds to $(x'_k, \ell'_k) \in q_k$ and $F_i \in \ell'_k$. This is due to the fact that after the occurrence of the fault $f \in \Sigma_{f_i}$, only F_i-faulty states in G are reachable.

Case 2 (A fault $f \in \Sigma_{f_i}$ occurs while the diagnoser G_d is in the cycle of F_i-uncertain states): If the diagnoser is in one of the states q_k of the aforementioned cycle of F_i-uncertain states, and the fault $f \in \Sigma_{f_i}$ occurs, upon observing the first observable event, the diagnoser will switch to the state q_{k+1} of the cycle, in which the faulty state x'_{k+1} in G will be reached, which corresponds to $(x'_{k+1}, \ell'_{k+1}) \in q_{k+1}$ and $F_i \in \ell'_{k+1}$.

In both cases, as the plant G is live and has no cycle of unobservable events, visiting F_i-faulty states will continue. However, since the finite set of F_i-faulty pairs (x'_k, ℓ'_k) do not form a cycle, each can be visited only once, and then, the diagnoser has to leave the cycle. ■

We now can prove Theorem 1, which explains the necessary and sufficient conditions for asynchronous diagnosability of a given DES plant G based on the structure of its diagnoser G_d.

Theorem 1: (Asynchronous Diagnosability Theorem) The plant G with the live language $\mathcal{L}_G$, and with the asynchronous diagnoser G_d, constructed in Section III, is F_i-asynchronously diagnosable if and only if, there does not exist an F_i-indeterminate cycle in G_d.

Proof of Necessity: We prove that if G is F_i-asynchronously diagnosable, then there is no F_i-indeterminate cycle in G_d. For this purpose, by contradiction, assume that there is an F_i-indeterminate cycle, q_1, q_2, ..., q_n, in G_d, such that $\delta_d(q_k, e_k) = q_{k+1}$, $\delta_d(q_n, e_n) = q_1$, $e_k \in \Sigma_o$, $k = 1, \ldots, n$. According to Definition 7, any F_i-indeterminate cycle in G_d corresponds to at least one cycle of F_i-faulty states x'_1, x'_2,..., x'_n in G such that $x'_{k+1} \in \delta(x'_k, t'_k)$, $x'_1 \in \delta(x'_n, t'_n)$, $t'_k \in P^{-1}_{ext(\mathcal{L}_G)}(e_k)$ for $k = 1, \ldots, n$; and one cycle of non-F_i

faulty states x_1, x_2,..., x_n in G such that $x_{k+1} \in \delta(x_k, t_k)$, $k = 1, \ldots, n-1$, $x_1 \in \delta(x_n, t_n)$, $t_k \in P^{-1}_{ext(\mathcal{L}_G)}(e_k)$ for $k = 1, \ldots, n$, for which (x'_k, ℓ'_k) and $(x_k, \ell_k) \in q_k$, $F_i \in \ell'_k$, and $F_i \notin \ell_k$, $k = 1$, ..., n.

Without loss of generality, assume that in this F_i-indeterminate cycle, the F_i-faulty pair (x'_1, ℓ'_1) in q_1, $F_i \in \ell'_1$, is the first F_i-faulty pair that is reachable from one of the pairs in q_0 by a string r, $q_1 = \delta_d(q_0, r)$. This means that the state x'_1 has to be reached from a pair $(y'_0, \ell'_0) \in q_0$, with a string $r'_y \in P^{-1}_{ext(\mathcal{L}_G)}(r)$, such that $\delta(y'_0, r'_y) = x'_1$. Correspondingly, since q_1 is an F_i-uncertain state, there exists a pair $(x_1, \ell_1) \in q_1$, $F_i \notin \ell_1$, which is reachable from a pair $(y_0, \ell_0) \in q_0$, $F_i \notin \ell_0$, with a non-F_i-faulty string $r_y \in P^{-1}_{ext(\mathcal{L}_G)}(r)$ meaning that $\delta(y_0, r_y) = x_1$, $f \notin r_y, \forall f \in \Sigma_{f_i}$. Also, according to the construction procedure of the diagnoser, for the states y_0 and y'_0, there exist strings $w, w' \in \mathcal{L}_G$ such that $\forall f \in \Sigma_{f_i}$, $f \notin w$, $\delta(x_0, w) = y_0$ and $\delta(x_0, w') = y'_0$. Figure 4 shows the graphical representation of the described situation. Now, since x'_1 is an F_i-faulty state, two cases may happen: there exists a $f \in \Sigma_{f_i}$ such that $f \in w'$ (if the fault occurs before the diagnoser activation) or $f \in r'_y$ (if the fault occurs after the activation of the diagnoser), based on which, we will have two following cases:

Case 1 (fault occurs before the diagnoser activation, $f \in w'$): Let $t = r'_y(t'_1 t'_2 ... t'_n)^K$, with arbitrary large K so that $||t|| > n_i$ for any $n_i \in \mathbb{N}$. Also, corresponding to the elements of Definition 4, set the strings $s = w'$, $u = w$, and $v = r_y(t_1 t_2 ... t_n)^K$.

Case 2 (fault occurs after the diagnoser activation, $f \in r'_y$): Let $t = (t'_1 t'_2 ... t'_n)^K$, with arbitrary large K so that $||t|| > n_i$ for any $n_i \in \mathbb{N}$. Also, corresponding to the elements of Definition 4, set the strings $s = w' r'_y$, $u = w r_y$, and $v = (t_1 t_2 ... t_n)^K$.

In both cases, the arbitrary long string t is in $\mathcal{L}_{G/s}$, $v \in P^{-1}_{ext(\mathcal{L}_G)}(P(t))$ and $uv \in \mathcal{L}_G$, where $f \in s$ but for all $f' \in \Sigma_{f_i}$, $f' \notin uv$, which violates the F_i-asynchronous diagnosability condition in Definition 4 and contradicts with the assumption which was made at the beginning of this proof.

Proof of Sufficiency: Next, we prove that if there is no F_i-indeterminate cycle in G_d, then G is F_i-asynchronously diagnosable. For this purpose, we show that upon its occurrence, fault $f \in \Sigma_{f_i}$, can be diagnosed by reaching an F_i-certain state within a finite number of transitions in G. Let's consider the case that a fault $f \in \Sigma_{f_i}$ has occurred before activating the diagnoser. When activated, the diagnoser enters q_0. Since G is live and has no cycle of unobservable events (the length of unobservable strings are bounded), the state of diagnoser keeps changing. This means that upon observing the first observable event, the diagnoser will update its estimation of the system state and condition, and transits to a new state. Since $f \in \Sigma_{f_i}$ has occurred in the past, the diagnoser will either transition to an F_i-certain state, which trivially diagnoses the fault occurrence, or transition to an F_i-uncertain state. For the latter case, however, we can prove that the diagnoser will eventually reach an F_i-certain state. For this purpose, we know that if

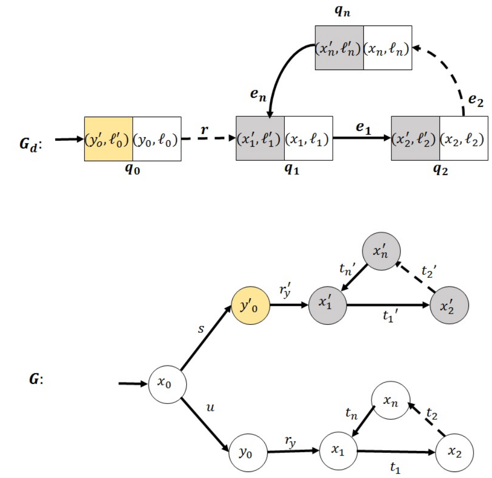

Fig. 4: An F_i-indeterminate cycle in G_d can be projected to an infinitely large F_i-faulty string and an infinitely large non-F_i-faulty string with the same observation, making the system F_i-asynchronously undiagnosable.

$f \in \Sigma_{f_i}$ has occurred in the past, an F_i-uncertain state may only reach either an F_i-uncertain state or an F_i-certain state. Also, if the diagnoser transitions to a cycle of F_i-uncertain states, based on the sufficiency part's assumption, it will not be a F_i-indeterminate cycle, and hence, based on Lemma 3, the diagnoser will leave the cycle. Since the diagnoser has a finite number of states, the state of diagnoser cannot remain F_i-uncertain and will eventually transition to an F_i-certain state, concluding that the fault type F_i has occurred in the past. If the fault $f \in \Sigma_{f_i}$ occurs after activation of the diagnoser, upon observing the first observable event, the diagnoser will transition to either an F_i-certain state or an F_i-uncertain state, and the same argument can be applied to show that the diagnoser will switch to an F_i-certain state and will diagnose the fault occurrence in a finite number of diagnoser state transitions. ■

Example 3: For the plant G_1 in Example 1, there are three cycles of F_i-uncertain states in G_{d1} in Fig. 2:

1) Cycle 1 consists of q_5, q_7 and q_8, which are F_2-uncertain.
2) Cycle 2 consists of only q_3 which is F_1-uncertain.
3) Cycle 3 consists of q_1 and q_4 which are both F_1-uncertain and F_2-uncertain.

Cycle 1 in G_{d1} corresponds to a cycle of non-F_2-faulty states in G_1: $\begin{tikzcd} 4 \arrow[r, "\alpha"] & 1 \arrow[r, "\beta"] & 2 \arrow[r, "f_1"] & 3 \arrow[lll, bend left, "\beta"] \end{tikzcd}$. However, for this cycle there does not exist any cycle of F_2-faulty states in G_1. Therefore, this cycle of F_2-uncertain states in G_{d1} is not F_2-indeterminate.

Cycle 2 in G_{d1} corresponds to a self-loop in G_1 at the state $x = 5$, which is both F_1-faulty and non-F_1-faulty, concluding Cycle 2 is F_1-indeterminate.

Cycle 3 in G_{d1} corresponds to a cycle of non-F_2-faulty states in G_1: $\begin{tikzcd} 1 \arrow[r, "\beta"] & 2 \arrow[l, bend left, "\alpha"] \end{tikzcd}$. However, for this cycle there does

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2912712, IEEE Transactions on Automatic Control

not exist any cycle of F_2-faulty states in G_1. Therefore, this cycle of F_2-uncertain states in G_{d1} is not an F_2-indeterminate. However, the states $x_1 = 1$ and $x_2 = 2$ in G_1 are both F_1-faulty and non-F_1-faulty, concluding that Cycle 3 is an F_1-indeterminate cycle.

All in all, Cycle 2 and Cycle 3 are F_1-indeterminate and there is no F_2-indeterminate cycle in this diagnoser. Hence, based on Theorem 1, we can conclude that the plant G_1 is F_2-asynchronously diagnosable but not F_1-asynchronously diagnosable. Then, based on Definition 4, the plant G_1 is not asynchronously diagnosable.

V. Asynchronous Diagnosis vs Synchronous Diagnosis

In synchronous diagnosis [10], all the system behavioral information (pre- and post-fault occurrence) is available to the diagnoser, G_{sd}. Accordingly, the synchronous diagnosability can be defined as:

Definition 8: (F_i-synchronous Diagnosability [10]) The plant G with the live language $\mathcal{L}_G$, is said to be F_i-Synchronously diagnosable with respect to the fault type F_i and the natural projection P, if for all $s \in \mathcal{L}_G$, $f \in \Sigma_{f_i}$, $f \in s$, there exists an upper bound $n_i \in \mathbb{N}$, such that for any string $t \in \mathcal{L}_{G/s}$ with $|t| \geq n_i$, the following condition holds:

$$\{\forall w \in \mathcal{L}_G, w \in P^{-1}_{ext(\mathcal{L}_G)}(P(s.t))\} \Rightarrow f \in w \qquad (7)$$

The generic string $s.t$ in the above definition, represents both the past history of the system starting from x_0, s, and post fault observations, t. However, in the proposed asynchronous diagnosis in this paper, the diagnoser only has access to the system behavioral information that occurs after the diagnoser's activation. This can be observed in Definitions 3 and 4, where the asynchronous diagnosis requires faulty strings to be distinguishable only from their extension closure t instead of full observation of the system $s.t$.

An interesting question would be what is the relation between the synchronous and asynchronous diagnosability. Intuitively, asynchronous diagnosability implies synchronous diagnosability, as it is proven in the following theorem:

Theorem 2: If G is F_i-asynchronously diagnosable, then it is F_i-synchronously diagnosable.

Proof: By contradiction assume that the system G is not F_i-synchronously diagnosable. Therefore, based on Definition 8, there exists a string $s \in \mathcal{L}_G$, $f \in \Sigma_{f_i}$, $f \in s$, an (infinitely) large string $t \in \mathcal{L}_{G/s}$ and a string $w \in \mathcal{L}_G, w \in P^{-1}_{ext(\mathcal{L}_G)}(P(s.t))\}$, such that $f \notin w$. Now, let $u = \epsilon$ and $v = w$. Then, for the string $s \in \mathcal{L}_G$, $f \in \Sigma_{f_i}$, $f \in s$, with arbitrarily long string $t \in \mathcal{L}_{G/s}$, we have $uv \in \mathcal{L}_G, v \in P^{-1}_{ext(\mathcal{L}_G)}(P(t))\}$ but $f \notin uv$, violating the conditions in Definition 3, contradicting with the F_i-asynchronously diagnosablilty of G. ■

Synchronous diagnosability, however, does not always imply the asynchronous diagnosability. Figure 5, for example, shows the plant G_2 with $\Sigma_f = \Sigma_{f_1} = \{f\}$, which is synchronously diagnosable but is not asynchronously diagnosable due to the F_1-indeterminate cycle consisting of F_1-uncertain states q_1 and q_4 in G_{d_2} in Figure 5.c.

Since the synchronous diagnoser G_{sd} contains all projected strings, and their associated states, it is possible to use the structure of G_{sd} to indirectly determine if system G is asynchronously diagnosable. Theorem 3 describes the conditions that a synchronously diagnosable plant is asynchronously diagnosable, solely based on the synchronous diagnoser structure. Before that, we need the following definition:

Definition 9: Let states q_1, q_2, $\ldots$, q_n form a cycle in G_{sd} such that $\delta_d(q_k, e_k) = q_{k+1}$ $k = 1, \ldots, n-1$, $\delta_d(q_n, e_n) = q_1$. If there exists a separate cycle of states in G_{sd} with $\delta_d(q'_k, e_k) = q'_{k+1}$ $k = 1, \ldots, n-1$, $\delta_d(q'_n, e_n) = q'_1$, then the cycles q_1, q_2, $\ldots$, q_n and q'_1, q'_2, $\ldots$, q'_n in G_{sd} are called associated cycles.

Example 4: Cycle s_2, s_3 and cycle s_5, s_4 in G_{sd_3} in Figure 5.b are associated cycles.

Theorem 3: An F_i-synchronously diagnosable plant G with synchronous diagnoser G_{sd}, is F_i-asynchronously diagnosable if and only if for any F_i-certain cycle in G_{sd}, there does not exist an associated non-F_i-certain cycle.

Proof of Necessity: We prove that if G is asynchronously diagnosable, then for any F_i-certain cycle in G_{sd} there does not exist an associated cycle of non-F_i-faulty states in G_{sd}. For this purpose, assume that there exists an F_i-certain cycle in G_{sd}, q'_1, q'_2, ..., q'_n, in G_{sd}, where $\delta_d(q'_k, e_k) = q'_{k+1}$ $k = 1, \ldots, n-1$, $\delta_d(q'_n, e_n) = q'_1$. By contradiction, assume that for this F_i-certain cycle in G_{sd} there exists an associated cycle of non-F_i-faulty states in G_{sd}, q_1, q_2, ..., q_n, so that $\delta_d(q_k, e_k) = q_{k+1}$ $k = 1, \ldots, n-1$, $\delta_d(q_n, e_n) = q_1$. From the construction procedure of G_{sd}, we know that for the F_i-certain cycle in G_{sd}, q'_1, q'_2, ..., q'_n, there exists a cycle of F_i-faulty states in G such that $x'_{k+1} \in \delta(x'_k, t'_k)$, $k = 1, \ldots, n-1$, $x'_1 \in \delta(x'_n, t'_n)$ and $P(t'_1.t'_2 \ldots t'_n) = e_1.e_2 \ldots e_n$. On the other hand, by a backward induction it can be proven that for every cycle of non-F_i-certain states in asynchronous diagnoser G_d, there exist at least one corresponding cycle of non-F_i-faulty states in G. Therefore, for the non-F_i-faulty states q_1, q_2, ..., q_n in G_{sd}, there exists a corresponding cycle of non-F_i-faulty states in G such that $x_{k+1} \in \delta(x_k, t_k)$, $k = 1, \ldots, n-1$, $x_1 \in \delta(x_n, t_n)$ and $P(t_1.t_2 \ldots t_n) = e_1.e_2 \ldots e_n$. For these two cycles, therefore, we will have $P(t'_1.t'_2 \ldots t'_n) = P(t_1.t_2 \ldots t_n)$.

Assume that states x_1 and x'_1 are reachable from the initial state, x_0, in G, by the strings u and s, i.e., $x_1 \in \delta(x_0, u)$ and $x'_1 \in \delta(x_0, s)$. Consider the string $t = (t'_1.t'_2 \ldots t'_n)^K \mathcal{L}_{G/s}$ and $v = (t_1.t_2 \ldots t_n)^K \mathcal{L}_{G/u}$, where $K = kn_1n_2 \in \mathbb{N}$, $n_1 = |t'_1.t'_2 \ldots t'_n|$, $n_2 = |t_1.t_2 \ldots t_n|$, and k is an arbitrarily large number. With this setup, $f_i \in st$ but $f_i \notin uv$, $s.t, u.v \in \mathcal{L}_G$. However, from the infinitely large post-activation observations $P(t)$, one cannot distinctly detect the occurrence of the fault f_i, as $P(v) = P(t)$, $f_i \in st$, and $f_i \notin uv$, which contradictorily violates the F_i-asynchronous diagnosability condition in Definition 3 for the plant G.

Proof of Sufficiency: To have the system G F_i-asynchronously diagnosable, it is sufficient to have no F_i-indeterminate cycle in G_d (Theorem 1). To check this indirectly from G_{sd}, since G_{sd} contains all projected strings and their associated states, any F_i-certain cycle in G_{sd} with an associated cycle of non-F_i-faulty states can create an F_i-indeterminate cycle in G_d.

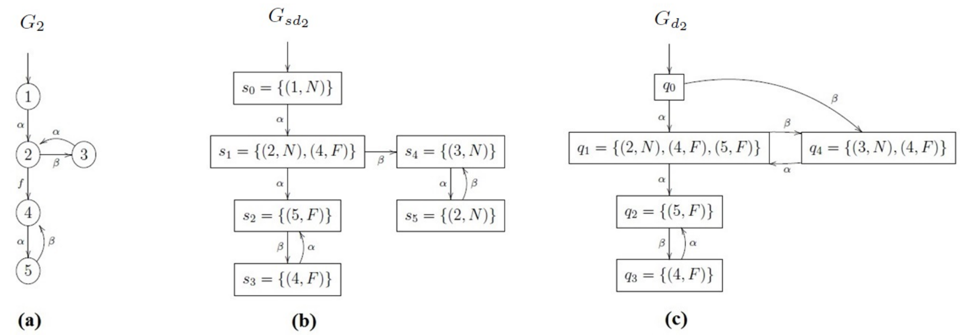

Fig. 5: (a) The DES plant G_2, (b) the synchronous diagnoser G_{sd_2} for the plant G_2, (c) the asynchronous diagnoser G_{d_2} for the plant G_2.

Since G_{sd} is F_i-synchronously diagnosable, there is no F_i-indeterminate cycle in G_{sd}. Therefore, the only remaining sufficient condition is to have no F_i-certain cycle in G_{sd} with an associated cycle of non-F_i-faulty states. Otherwise, the F_i-certain cycle in G_{sd} with an associated non-F_i-faulty cycle, form an F_i-indeterminate cycle in G_d, whose F_i-certain states are rooted in an F_i-certain cycle in G_{sd} and whose non-F_i-certain states are rooted in a non-F_i-certain cycle in G_{sd}. ■

Example 5: In Figure 5.b, the cycle of s_2, s_3 in G_{sd_2} is F_1-certain and cycle of s_5, s_4 in G_{sd_2} is Normal (so is non-F_1-faulty). This creates an F_1-indeterminate cycle in G_{d_2} (Figure 5c), thus G_2 is not asynchronously diagnosable, conforming Theorem 3.

VI. CONCLUSION

Through this paper, we introduced the new concept of asynchronous diagnosability and developed a systematic and analytical approach to construct a diagnoser that is able to detect and isolate faults in a DES plant without having access to the history of the past behaviors of the system. This allows the diagnoser to be asynchronously activated anytime, even after the occurrence of the faults. Moreover, we provided the necessary and sufficient conditions to check the asynchronous diagnosability of a given DES plant based on the structure of the constructed diagnoser. Lastly, the necessary and sufficient conditions were derived for asynchronous diagnosability based upon the structure of the synchronous diagnoser.

ACKNOWLEDGMENT

The authors would like to acknowledge the support from NSF under the award number 1832110, Air Force Research Laboratory and OSD for sponsoring this research under agreement number FA8750-15-2-0116, and Singapore Ministry of Education Academic Research Grant RG91/18-(S)-SU RONG (VP). The first author would like to acknowledge the support from Title III HBGI grant from the U.S. Department of Education.

REFERENCES

[1] R. M. Murray, K. J. Astrom, S. P. Boyd, R. W. Brockett, and G. Stein, "Future directions in control in an information-rich world," *IEEE Control Systems Magazine*, vol. 23, no. 2, pp. 20–33, 2003.

[2] J. Carreno, G. Galdorisi, S. Koepenick, and R. Volner, "Autonomous systems: Challenges and opportunities," DTIC Document, Tech. Rep., 2010.

[3] A. Pouliezos and G. S. Stavrakakis, *Real time fault monitoring of industrial processes*. Springer Science & Business Media, 2013, vol. 12.

[4] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results," *Automatica*, vol. 26, no. 3, pp. 459–474, 1990.

[5] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*. Springer Science & Business Media, 2009.

[6] P. J. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 81–98, 1989.

[7] N. Ran, H. Su, A. Giua, and C. Seatzu, "Codiagnosability analysis of bounded petri nets," *IEEE Transactions on Automatic Control*, vol. 63, no. 4, pp. 1192–1199, 2018.

[8] A. Boussif, M. Ghazel, and K. Klai, "Dpn-sog: A software tool for fault diagnosis of labeled petri nets using the semi-symbolic diagnoser," in *11ème Colloque sur la Modélisation des Systèmes Réactifs (MSR 2017)*, 2017.

[9] F. Lin, "Diagnosability of discrete event systems and its applications," *Discrete Event Dynamic Systems*, vol. 4, no. 2, pp. 197–212, 1994.

[10] 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.

[11] Y. Wang, T.-S. Yoo, and S. Lafortune, "Diagnosis of discrete event systems using decentralized architectures," *Discrete Event Dynamic Systems*, vol. 17, no. 2, pp. 233–263, 2007.

[12] W. Qiu and R. Kumar, "Decentralized failure diagnosis of discrete event systems," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 36, no. 2, pp. 384–395, 2006.

[13] O. Contant, S. Lafortune, and D. Teneketzis, "Diagnosability of discrete event systems with modular structure," *Discrete Event Dynamic Systems*, vol. 16, no. 1, pp. 9–37, 2006.

[14] R. Su and W. M. Wonham, "Global and local consistencies in distributed fault diagnosis for discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 50, no. 12, pp. 1923–1935, 2005.

[15] A. Paoli and S. Lafortune, "Safe diagnosability for fault-tolerant supervision of discrete-event systems," *Automatica*, vol. 41, no. 8, pp. 1335–1347, 2005.

[16] L. K. Carvalho, J. C. Basilio, and M. V. Moreira, "Robust diagnosis of discrete event systems against intermittent loss of observations," *Automatica*, vol. 48, no. 9, pp. 2068–2078, 2012.

[17] S. T. S. Lima, J. C. Basilio, S. Lafortune, and M. V. Moreira, "Robust diagnosis of discrete-event systems subject to permanent sensor failures." in *WODES*, 2010, pp. 90–97.

[18] J. Zaytoon and S. Lafortune, "Overview of fault diagnosis methods for discrete event systems," *Annual Reviews in Control*, vol. 37, no. 2, pp. 308–320, 2013.

[19] S. H. Zad, R. H. Kwong, and W. M. Wonham, "Fault diagnosis in discrete-event systems: framework and model reduction," *IEEE Transactions on Automatic Control*, vol. 48, no. 7, pp. 1199–1212, 2003.

[20] A. White and A. Karimoddini, "Asynchronous fault diagnosis of discrete event systems," in *2017 American Control Conference (ACC)*, May 2017, pp. 3224–3229.