Generalized Robust Diagnosability of Discrete Event Systems

Lilian K. Carvalho ∗ Marcos V. Moreira ∗ João C. Basilio ∗

Abstract: We address the problem of robust diagnosability of discrete event systems described by a class of automata, where each automaton in the class generates a distinct language. We introduce a new definition which generalizes all previous definitions of robust diagnosability: for this reason it is referred here to as generalized robust diagnosability. We also present a necessary and sufficient condition for the generalized robust diagnosability and propose a polynomial time algorithm for its verification.

Keywords: Discrete event systems, fault diagnosis, finite automata, robust diagnosis.

1. INTRODUCTION

The usual approach to fault diagnosis of discrete event systems (DES) modeled by automata assumes that the language generated is uniquely determined by the model; see e.g. Sampath et al. (1995), Debouk et al. (2000), Zad et al. (2003), Qiu and Kumar (2006) and the references therein. The notion of language uncertainty in fault diagnosis of DES has been introduced by Basilio and Lafortune (2009), initially, in the context of decentralized diagnosability, assuming unreliable communication between a local site and the coordinator. More recently, this notion has been extended to also encompass permanent and intermittent sensor failures (Lima et al., 2010; Carvalho et al., 2010), and model uncertainty due to inexact knowledge of the real system (Takai, 2010).

The definition of robust diagnosability of discrete event systems (DES) subject to permanent sensor failures proposed in Lima et al. (2010) deploys the redundancy that may exist in a set of diagnosis bases (set of events that guarantee fault diagnosability) with the view to verifying fault diagnosability even in the occurrence of permanent sensor failures. A robust diagnoser is obtained using several partial diagnosers, each one built for a particular diagnosis basis and constructed from the same plant \( G \) with a specific set of observable events that is related to the diagnoses basis of the partial diagnoser. The robust diagnosability problem posed in Lima et al. (2010) can be regarded as the problem of identifying the occurrence of an unobservable fault event with uncertainties in the set of observable events.

The definition of robust diagnosability introduced by Takai (2010) requires that the system be described by a set of possible models \( \{ G_i : i \in I_m \} \) over a common event set \( \Sigma \), where \( I_m := \{1, 2, \ldots, m\} \) and \( m \in \mathbb{N} \) denotes the number of system models. In this definition, each possible model has its own non-faulty specification and the language generated by each model is live. Differently from Lima et al. (2010), all models \( G_i \) in Takai (2010) are assumed to have the same observable event set. An algorithm for the verification of the robust diagnosability condition based on a previously proposed algorithm by Qiu and Kumar (2006) is also presented in Takai (2010).

We introduce here a new definition of robust diagnosability, called generalized robust diagnosability, that generalizes the robust diagnosability definitions of Lima et al. (2010) and Takai (2010), in the sense that uncertainties are considered both in the system model and in the observable event set. In our approach, the system model belongs to a set of possible models over a common event set \( \Sigma \) — as in Takai (2010) — with the difference that, here, each model may have a distinct set of observable events. This situation may happen, for instance, due to a sensor failure that changes the behavior of a system under supervision, leading to a completely new automaton model with distinct sets of states and observable events, and with a different transition function; the reader is referred to the works by Lin (1993), Young and Garg (1991), and Saboori and Hashtrudi-Zad (1993), which consider model uncertainties in the context of supervisory control. We also propose a polynomial time algorithm for the verification of the generalized robust diagnosability of a DES which has lower computational complexity than that proposed by Takai (2010).

This paper is organized as follows: in Section 2 we present the preliminary concepts, and in Section 3 we introduce the definition of generalized robust diagnosability; in Section 4, we propose a polynomial time algorithm for the verification of robust diagnosability and, in the sequel, we determine its computational complexity; we present an example in Section 5 to illustrate the algorithm proposed in the paper, and draw conclusions in Section 6.
2. PRELIMINARIES

Let $G = (X, \Sigma, \Gamma, f, x_0)$ denote the automaton model of a DES, where $X$ is the finite state space, $\Sigma$ is the set of events, $\Gamma$ is the feasible event function, $f$ is the transition function, and $x_0$ is the initial state. Let us denote the language generated by $G$ as $L(G) = L$, and let a path be defined as a sequence of states and events $(x_k, \sigma_1, x_{k+1}, \sigma_2, \ldots, \sigma_l, x_{k+l})$, for $l > 0$, such that $x_{k+l} = f(x_{k+l-1}, \sigma_l), \ i \in \{1,2,\ldots,l\}$. A path forms a cycle if $x_{k+l} = x_k$.

Let us partition $\Sigma$ as $\Sigma = \Sigma_o \cup \Sigma_{uo}$, where $\Sigma_o$ and $\Sigma_{uo}$ are, respectively, the set of observable and unobservable events, and let $\Sigma_f \subseteq \Sigma_o$ denote the set of fault events. In addition, assume, without loss of generality, that $\Sigma_f = \{\sigma_f\}$.

Assume also that $G$ models the normal and the faulty behavior of the system and let $H$ be the subautomaton of $G$ that represents the non-faulty behavior of the system. Thus, the language generated by $H$, denoted as $K$, is a prefix-closed language formed with all traces of $L$ that do not have any event from the fault event set $\Sigma_f$.

The fault event $\sigma_f$ is said to be diagnosable if the occurrence of $\sigma_f$ can be detected within a finite number of transitions after its occurrence using only traces formed with events in $\Sigma_o$. Formally, language diagnosability is defined as follows (Sampath et al., 1995).

**Definition 1.** Let $L$ be the prefix-closed language generated by the system $G$ and let $K \subseteq L$ denote the prefix-closed language generated by $H$. Assume also that $L$ is live. Then, $L$ is diagnosable with respect to $P_o : \Sigma^* \rightarrow \Sigma^*_o$ and $\Sigma_f$ if and only if

$$\forall n \in \mathbb{N}(\exists s \in L \setminus K)(\forall st \in L \setminus K, |t| \geq n) \Rightarrow (\forall w \in K, P_o(st) \neq P_o(w)).$$

3. GENERALIZED ROBUST DIAGNOSABILITY

A definition of robust diagnosability of DES against permanent sensor failures is presented in Lima et al. (2010), under the following assumptions: (i) sensor failures can only occur before the first occurrence of the event recorded by the sensors whose failures are being investigated; (ii) the non-observation of the events does not change the language generated by the system; (iii) the robust diagnosability is stated in terms of distinct sets of observable events $\Sigma_o$, $i = 1, \ldots, m, m \in \mathbb{N}$, where $\Sigma_o$ is a basis for diagnosability, i.e., $L$ is diagnosable with respect to projections $P_o : \Sigma^* \rightarrow \Sigma^*_o, i = 1, \ldots, m$, and $\Sigma_f$, as follows.

**Definition 2.** (Robust diagnosability against permanent sensor failures) Let $L$ be the live prefix-closed language generated by automaton $G$ and let $K \subseteq L$ denote the prefix-closed language generated by $H$. In addition, assume that $\Sigma_o, i = 1, \ldots, m$, is a diagnosis basis for $L$. Then, $L$ is robustly diagnosable against permanent sensor failures with respect to projections $P_o : \Sigma^* \rightarrow \Sigma^*_o, i = 1, \ldots, m$, and $\Sigma_f = \{\sigma_f\}$, if and only if

$$\forall n \in \mathbb{N}(\forall s \in L \setminus K)(\forall st \in L \setminus K, |t| \geq n) \Rightarrow (\forall i, j \in \{1,2,\ldots,m\}, i \neq j)(\forall w \in K, P_o(st) \neq P_o(w)).$$

Instead of using a single model, Takai (2010) deploys a set of possible automata to model the system, and introduces another definition of robust diagnosability, similar to that presented in Lima et al. (2010), assuming that: (i) the real system model belongs to a set of possible models $G_i = (X_i, \Sigma_i, f_i, x_{0i}), i = 1, 2, \ldots, m$; (ii) each automaton model $G_i$ generates a different language $L_i$ and has a distinct non-faulty behavior, described by a nonempty closed sublanguage $K_i \subseteq L_i$; (iii) they all share the same observable event set $\Sigma_o$.

In the work by Lima et al. (2010), the system is modeled by a unique automaton $G$ and the uncertainties are due to loss of observable events. On the other hand, the robust diagnosability problem formulated in Takai (2010) is only suitable when the model is subject to change and cannot be used in a straightforward way when the model uncertainties are due to either permanent (Lima et al., 2010) or intermittent (Carvalho et al., 2010) loss of observable events. In order to encompass both, distinct observable event sets and different automaton models — therefore generalizing the robust diagnosability definitions presented in Lima et al. (2010) and Takai (2010) — we present the following definition.

**Definition 3.** (Generalized robust diagnosability) Let $L_i \subseteq \Sigma^*$ be the language generated by $G_i, i = 1, \ldots, m$, and assume that $L_i$ is live. In addition, assume that each model $G_i = (X_i, \Sigma_i, f_i, x_{0i}), i = 1, \ldots, m$, has a projection $P_o : \Sigma^* \rightarrow \Sigma^*_o$, and that $L_i$ is diagnosable with respect to $P_o$ and $\Sigma_f$. Let us denote $H_i$ as the subautomaton of $G_i$ that models the non-faulty behavior of the corresponding model, and $K_i \subseteq L_i$ the language generated by $H_i$. Then,

$$L = \{L_i : i \in I_m\},$$

the class of all possible languages generated by the class of automata

$$G = \{G_i : i \in I_m\},$$

is robustly diagnosable with respect to projections $P_o, i = 1, \ldots, m$, and $\Sigma_f = \{\sigma_f\}$, if and only if

$$\forall i \in I_m(\exists n_i \in \mathbb{N})(\forall s_i \in L_i \setminus K_i)(\forall s_t_i \in L_i \setminus K_i, |t_i| \geq n_i) \Rightarrow (\forall j \in I_m, j \neq i)(\forall w_j \in K_j, P_o(s_t_i) \neq P_o(w_j)).$$

4. VERIFICATION OF GENERALIZED ROBUST DIAGNOSABILITY

According to Definition 3, the problem of verifying the generalized robust diagnosability of a DES can be formulated as the problem of searching for traces $s_t_i \in L_i \setminus K_i$ and $w_j \in K_j, i \neq j$, where $s_i \in L_i \setminus K_i$ and $t_i$ is an arbitrarily long trace, such that $P_o(s_t_i) = P_o(w_j)$. If there exist traces $s_t_i, t_i,$ and $w_j$ that satisfy these conditions, then the language class $L$ will be non-robustly diagnosable with respect to projections $P_o, i = 1, 2, \ldots, m$, and $\Sigma_f$.

Let $G_i \in \mathbb{G}$ be a possible model for the system over the event set $\Sigma$ and assume that $G_i$ generates language $L_i \subseteq L$. Let us partition $\Sigma$ as $\Sigma = \Sigma_o \cup \Sigma_{uo}$, where $\Sigma_o$ and $\Sigma_{uo}$ are the observable and unobservable event sets for model $G_i$, respectively, and define $\Sigma_o = \bigcup_{i=1}^{m} \Sigma_o_i$. We make the following assumption.

**A1.** $L_i$ is diagnosable with respect to projection $P_o : \Sigma^* \rightarrow \Sigma^*_o$ and $\Sigma_f$, for $i = 1, 2, \ldots, m$. \qed
Let us now define the following one-to-one functions $R_i : \Sigma \rightarrow \Sigma_{R_i}$, for $i = 1, 2, \ldots, m$, as (Moreira et al., 2010, 2011)

$$R_i(\sigma) = \begin{cases} \sigma, & \text{if } \sigma \in \Sigma_0 \cup \Sigma_f \\ \sigma_{R_i}, & \text{if } \sigma \in \Sigma_{\omega_0} \setminus \Sigma_f. \end{cases} \quad (1)$$

It is worth remarking that function $R_i$ just renames the events in $\Sigma_{\omega_0} \setminus \Sigma_f$. Furthermore, notice that function $R_i$ can be extended to domain $\Sigma^*$ in the usual way, as follows: $R_i(\varepsilon) = \varepsilon$, and $R_i(\sigma \tau) = R_i(\sigma)R_i(\tau)$, $\forall \sigma \in \Sigma^* \text{ and } \forall \tau \in \Sigma$. As a consequence, $R_i$ can also be extended to a language $L \subseteq \Sigma^*$ by simply applying it to all strings in $L$.

Based on $R_i$, we can also define the inverse renaming function, as follows:

$$R_i^{-1} : \Sigma_{R_i} \rightarrow \Sigma$$

where $\sigma_{R_i} = R_i(\sigma)$, with the following extension to domain $\Sigma_{R_i}$: $R_i^{-1}(\sigma_{R_i}) = R_i^{-1}(\tau)$ for all $\sigma, \tau \in \Sigma_{R_i}$ and $\sigma_{R_i}, \tau_{R_i} \in \Sigma_{R_i}$, and $R_i^{-1}(\varepsilon) = \varepsilon$.

In order to verify the existence of traces that lead to a violation of the generalized robust diagnosability condition given in Definition 3, we will present a polynomial time algorithm and, in the sequel, a theorem that proves its correctness. We also carry out the computational complexity analysis of the algorithm.

### 4.1 Verification Algorithm

We now present an algorithm for the verification of the generalized robust diagnosability.

#### Algorithm 1.

**Step 1** For each model $G_i = (X_i, \Sigma, f_i, \Gamma_i, x_{i,0})$, build automaton $\bar{G}_i = (X_i, \Sigma, f_{\bar{i}}, x_{\bar{i},0})$, where $\Sigma_{\bar{i}} = R_i(\Sigma)$, $\Gamma_{i,0}(x) = R_i(\Gamma_i(x))$, and $f_{\bar{i}}(x, R_i(\sigma)) = f_i(x, \sigma)$ for all $x \in X_i$ and $\sigma \in \Gamma_i(x)$.

**Step 2** Compute the faulty behavior automaton $F_i$ as follows:

- **Step 2.1**: Build the faulty label automaton $A_i = (X_{A_i}, \Sigma_{A_i}, f_{A_i}, x_{A_i,0})$, where $X_{A_i} = (N, Y)$, $x_{A_i,0} = N$, and $f_{A_i}(N, \sigma_{A_i}) = Y$ and $f_{A_i}(Y, \sigma_{A_i}) = Y$.

- **Step 2.2**: Compute $\bar{G}_i = G_i \parallel A_i$ and mark all states of $\bar{G}_i$ that have the second component equal to $Y$.

- **Step 2.3**: Compute the coaccessible part of automaton $\bar{G}_i$ and define the faulty behavior automaton $F_i = (X_{F_i}, \Sigma_{F_i}, f_{F_i}, x_{F_i,0})$, by unmarking all marked states of $\bar{G}_i$.

- **Step 2.4**: Redefine the event set of $F_i$ as $\Sigma_{F_i} = \Sigma_{\bar{i}} \cup \Sigma_o$.

**Step 3** Build the non-faulty automaton $H_i$ as:

- **Step 3.1**: Define $\Sigma_{H_i} = \Sigma_{R_i} \setminus \Sigma_f$, and build automaton $Z_i = (\{N\}, \Sigma_{Z_i}, f_{Z_i}, x_{Z_i,0})$ composed of a single state with a self-loop labeled with all events in $\Sigma_{Z_i}$.

- **Step 3.2**: Construct $H_i = G_i \times Z_i = (X_{H_i}, \Sigma_{H_i}, f_{H_i}, x_{H_i,0})$.

- **Step 3.3**: Redefine the event set of each $H_i$ as $\Sigma_{H_i} = \Sigma_{H_i} \setminus \Sigma_f$.

**Step 4** Construct the augmented automaton $H_i = (X_{H_i}, \Sigma_{H_i}, f_{H_i}, x_{H_i,0})$ from automaton $H_i$ as follows:

- **Step 4.1**: Define $\Sigma_{H_i} = \Sigma_{H_i} \cup \Sigma_o$.

- **Step 4.2**: Define $x_{H_i,0} = x_{H_i,0}$.

- **Step 4.3**: Add a new state $D_i$ to the state space of $H_i$. Thus, $X_{H_i} = X_{H_i} \cup \{D_i\}$.

- **Step 4.4**: For each $x_{H_i} \in X_{H_i}$ define:

$$f_{H_i}(x_{H_i}, \sigma) = \begin{cases} f_{H_i}(x_{H_i}, \sigma), & \text{if } \sigma \in \Gamma_{H_i}(x_{H_i}) \\ D_i, & \text{if } \sigma \in \Sigma_{\omega_0} \setminus \Gamma_{H_i}(x_{H_i}), \text{ undefined, otherwise} \end{cases}$$

and for $x_{H_i} = D_i$ define:

$$f_{H_i}(x_{H_i}, \sigma) = \begin{cases} D_i, & \text{for all } \sigma \in \Sigma_{\omega_0} \setminus \Gamma_{H_i}(x_{H_i}), \text{ undefined, otherwise} \end{cases}$$

**Step 5** For $i = 1, 2, \ldots, m$, compute verifier $V_i$ whose $j$th state $x_{V_i,j} \in X_{F_i} \times (\{x_{i,0}\} \cup \{x_{i,0}\})$ and the $j$th state of $X_{F_i}$ is $x_{F_i,j} \in X_{F_i} \times X_{A_i}$, by making a composition of $F_i$, $H_1, \ldots, H_{i-1}, H_{i+1}, \ldots, H_m$, following the same procedure as for the parallel composition $F_i \parallel (\prod_{\{i\} \neq \{j\}} H_j)$, except that if state $(x_{F_i,j}, D_1, \ldots, D_{i-1}, D_{i+1}, \ldots, D_m)$ is reached, where $x_{F_i,j} \in X_{F_i}$, then its feasible event set is forced to be the empty set $\emptyset$, i.e.,

$$\Gamma_{V_i}(x_{F_i,j}, D_1, \ldots, D_{i-1}, D_{i+1}, \ldots, D_m) = \emptyset.$$  \[ \square \]

The following result provides a necessary and sufficient condition for generalized robust diagnosability.

**Theorem 1.** The class $\mathbb{L}$ is not robustly diagnosable with respect to $P_{\omega_0}$, $i = 1, 2, \ldots, m$, and $\Sigma$ if and only if there exists a cycle $c_i = (x_{V_i,j}, \sigma_k, x_{V_i,k+1}, \sigma_{k+1}, \ldots, x_{V_i,k}, \sigma_k, \ldots, x_{V_i,j})$, where $l \geq k > 0$, in at least one verifier $V_i$, $i \in I_m$, satisfying the following condition:

$$\exists j \in \{k, k+1, \ldots, l\} \text{ s.t. } (\sigma_j \in \Sigma_{R_i}) \land (x_{F_i,j} = \{x_{i,j}Y\}). \quad (4)$$

**Proof.** ($\Leftarrow$) Suppose that there exists a cycle $c_i = (x_{V_i,j}, \sigma_k, x_{V_i,k+1}, \sigma_{k+1}, \ldots, x_{V_i,k}, \sigma_k, \ldots, x_{V_i,j})$, where $l \geq k > 0$, in verifier $V_i$ that satisfies condition (4). Since $x_{F_i,j} = \{x_{i,j}Y\}$ for all $j \in \{k, k+1, \ldots, l\}$, then, by construction of $V_i$, we have that $x_{F_i,j} = \{x_{i,j}Y\}$ for all $j \in \{k, k+1, \ldots, l\}$. Thus, there exists a trace $s' \in \mathcal{L}(V_i)$, such that $s'$ contains the fault event $\sigma_f$, and $t' = (\sigma_k \sigma_{k+1} \ldots \sigma_{l})^n$, $\forall n \in \mathbb{N}$.

According to Algorithm 1, the states of $V_i$ that are equal to $(x_{F_i,j}, D_1, \ldots, D_{i-1}, D_{i+1}, \ldots, D_m)$ are deadlock states. Therefore, if verifier $V_i$ possesses a cycle $c_i$ that satisfies condition (4), then there will exist $q \in I_m = \{1, \ldots, i-1, i+1, \ldots, m\}$ such that $x_{H_{i,j}} \neq D_q$ for all $j \in \{k, k+1, \ldots, l\}$.

Let $\Sigma_{R} = \cup_{i=0}^{m} \Sigma_{R_i}$, and define the following projections:

$$P_{F_i} : \Sigma_{R} \rightarrow \Sigma_{F_i}, \quad (5)$$

$$P_{H_i} : \Sigma_{R} \rightarrow \Sigma_{H_i}, \quad (6)$$

for $p \in I_m$. According to Algorithm 1, $V_i$ is obtained through an automaton operation that is performed in the same manner as the parallel composition except for the deadlock state. Therefore

$$L(V_i) \subset P_{F_i}^{-1}[L(F_i)] \cap \left(\bigcap_{p=1, p \neq i}^{m} P_{H_p}^{-1}[L(H_p)]\right),$$

1 This is equivalent to saying that state $(x_{F_i,j}, D_1, \ldots, D_{i-1}, D_{i+1}, \ldots, D_m)$ is forced to be a deadlock state.
which implies that \( s't' \in P_{F_i}^{-1}[L(F_i)] \) since, by assumption, \( s't' \in L(V_i) \).

Let \( \bar{s}t' = P_{F_i}(s't') \), where \( \bar{s} = P_{F_i}(s') \) and \( \bar{t} = P_{F_i}(t') \). Since \( P_{F_i} \left[ P_{F_i}^{-1}[L(F_i)] \right] = L(F_i) \), it is not difficult to see that \( \bar{s}t' \in L(F_i) \). In addition, since \( t' = (\sigma_1 \rho_1, \ldots, \sigma_l \rho_l) \), \( \forall n \in \mathbb{N} \), and, by assumption, there exists an event \( \sigma_j \in \Sigma_{R_i} \), for some \( j \in \{k, k + 1, \ldots, l\} \), we may conclude that trace \( \bar{t} \in P_{F_i}(t') \) can be made arbitrarily long. At this point, it is worth reminding that \( L(G_{R_i}) = L(G_{R_i}) \) and \( L(F_i) \subseteq L(G_{R_i}) \). As a consequence, \( \bar{s}t' \in L(G_{R_i}) \). Therefore, since \( G_{R_i} \) is obtained from \( G_i \) by renaming its events in \( \Sigma \) through the renaming function \( R_i \), defined in Equation (1), it is clear that, associated with \( \bar{s}t' \in L(G_{R_i}) \), there exists a sequence \( st \in L_i \), which is obtained as follows:

\[
st = R_i^{-1}(\bar{s}t).
\]

Let \( \bar{q} = P_{H_i}(s't') \). Then \( s't' \in P_{H_i}^{-1}[L(H_i)] \) since \( \bar{q} \in L(H_i) \). In addition, using the fact that \( P_{H_i} \left[ P_{H_i}^{-1}[L(H_i)] \right] = L(H_i) \), then \( \bar{q} \in L(H_i) \). By assumption, state \( x_{H_i} \), is different from \( D_q \) for all \( j \in \{k, k + 1, \ldots, l\} \), and thus \( \bar{q} \in L(H_i) \). Notice that \( H_{R_i} \) is obtained from \( H_q \) after renaming the events of the set \( \Sigma \) using function \( R_q \). Thus, associated with \( \bar{q} \), there exists a non-faulty trace \( s_q \in L(G_q) \) which is given as:

\[
s_q = R^{-1}_q(\bar{q}).
\]

To conclude the if part of the proof, notice that due to the construction of \( F_i, H_i \), \( G_i \) and \( V_i \) — Steps 2, 4 and 5, respectively, of Algorithm 1 — and based on the fact that \( x_{H_i} \neq D_q \) for all \( j \in \{k, k + 1, \ldots, l\} \), in cycle \( c_l \), then all events in \( \Sigma_q \) that are used to form trace \( s't' \) — and consequently, \( st \) and \( sq \) — are in \( \Sigma_{R_i} \cap \Sigma_{o_q} \). Therefore, \( P_{o_q}(st) = P_{o_q}(sq) \), which leads to a violation of the robust diagnosability condition in Definition 3.

(\( \rightarrow \)) Assume that the class \( L \) is not robustly diagnosable with respect to projections \( P_{o_q}, i = 1, \ldots, m, \) and \( \Sigma_f = \{\sigma_i\} \). There exists a pair \((i, q) \in I_m \times M_q \), such that for some \( s_i \in L_i \setminus K_i \), with \( s_i \in L_i \setminus K_i \) and \( |t_i| = n_i, \forall n_i \in \mathbb{N} \), and for some \( w_q \in K_q \) \( (w_q \) not necessarily arbitrarily long), \( P_{o_q}(s_i t_i) = P_{o_q}(w_q) \).

According to Algorithm 1, \( G_{R_i} \) and \( G_{R_q} \) are obtained by renaming the unobservable events of \( G_i \) and \( G_q \), respectively, through the renaming functions \( R_i \) and \( R_q \). Therefore, the following conclusions can be drawn:

\section*{4.2 Computational complexity of algorithm 1}

Table 1 shows the maximum number of states and transitions of all automata that must be computed in order to obtain \( V_q \) according to Algorithm 1, assuming that there are \( m \) possible models \( G_i \in G \). It can be checked that the number of states and transitions of \( V_q \), in the worst case, equal to \( 2[X_i] \prod_{j=1,j \neq i}^{m}[X_j] + 1 \) and \( 2[X_i] \prod_{j=1,j \neq i}^{m}[X_j] + 1 \) \([m(|\Sigma_q| - |\Sigma_f|) + |\Sigma_f|] \), respectively. Therefore, the complexity of Algorithm 1 is \( O(m[X_i] \prod_{j=1,j \neq i}^{m}[X_j](|\Sigma_q| - |\Sigma_f|)) \). The intermediate steps that lead to the worst case bound above are presented in Carvalho (2011).

Remark 1. Notice that the computational complexity of Algorithm 1 is \( O(m[X_i] \prod_{j=1,j \neq i}^{m}[X_j](|\Sigma_q| - |\Sigma_f|)) \), which is smaller than the complexity of the algorithm proposed by Takai (2010), considering the number of states of \( H_{R_i} \) and \( G_i \) are equal, which is \( O(|X_i| \prod_{j=1,j \neq i}^{m}[X_j]|\Sigma|^{m+1}) \).

It is also important to remark that the size of the verifier automaton \( V_q \) is, in general, smaller than the worst case presented in Table 1 since the algorithm searches only for those traces in \( L_i \setminus K_i \) and \( K_j, i \neq j \), that may lead to a violation of the robust diagnosability condition. □
5. EXAMPLE

Let $G = \{G_1, G_2, G_3\}$ be the class of automata shown in Figure 1, and assume that $\Sigma = \{a, b, c, d, e, \sigma_a, \sigma_f\}$ is the set of all events used in the modeling of the system. It follows that $G_1$, $G_2$ and $G_3$. The objective here is to verify if the class of languages generated by the automata in $G$ is robustly diagnosable with respect to $P_{\alpha_i}$, $i = 1, 2, \ldots, m$, and $\Sigma_f = \{\sigma_f\}$.

Initially, note that $\Sigma_o = \bigcup_{i=1}^{m} \Sigma_{o_i} = \{a, b, c, d, e\}$. Now, according to Algorithm 1, the first step is to obtain the automaton $G_{R_i}$, $i = 1, 2, 3$, by renaming the events in $\Sigma_{o_i} \setminus \Sigma_f$. Therefore events $d, e$ and $\sigma_a$ should be renamed, respectively, as $d_{R_i}$, $e_{R_i}$, and $\sigma_{a_{R_i}}$ in $G_1$; and event $c$ as $c_{R_i}$ in $G_3$. Notice that no event needs to be renamed in $G_2$. The state transition diagram of the automata $G_{R_i}$, $i = 1, 2, 3$, are not shown since they are identical to those of $G_i$, $i = 1, 2, 3$, except for the above renaming.

The next step of Algorithm 1 is to compute the faulty automaton $F_{i}$, $i = 1, 2, 3$. Following Steps 2.1–2.4, the automata $F_1$, $F_2$ and $F_3$, shown in Figure 2, are obtained. Notice that, although only events $b$, $d_{R_i}$, $e_{R_i}$ and $\sigma_f$ appear in the state transition diagram of $F_1$, its event set is $\Sigma_{f_1} = \Sigma_{R_1} \cup \Sigma_o = \{a, b, c, d, e, d_{R_1}, e_{R_1}, \sigma_{a_{R_1}}, \sigma_f\}$. The same analysis can be carried out for $F_2$ and $F_3$, leading to $\Sigma_{f_2} = \Sigma_{a_{R_2}} \setminus \{a, b, c, d, e, d_{R_2}, \sigma_{a_{R_2}}, \sigma_f\}$ and $\Sigma_{f_3} = \{a, b, c, d, e, c_{R_3}, \sigma_{a_{R_3}}, \sigma_f\}$.

The next step of Algorithm 1 is to obtain the non-faulty automata $H_{R_1}$, $H_{R_2}$ and $H_{R_3}$ that accounts for the non-faulty behavior of $G_{R_1}$, $G_{R_2}$ and $G_{R_3}$, respectively, and, in the sequel to obtain the augmented automata $H_1$, $H_2$ and $H_3$, whose state transition diagrams are depicted in Figure 3. It is worth remarking that $\Sigma_{H_1} = \Sigma_{f_1} \setminus \{\sigma_f\}$, $\Sigma_{H_2} = \Sigma_{f_2} \setminus \{\sigma_f\}$ and $\Sigma_{H_3} = \Sigma_{f_3} \setminus \{\sigma_f\}$.

Fig. 1. Class of automata $G = \{G_1, G_2, G_3\}$: (a) $G_1$ with $\Sigma_{o_1} = \{a, b, c\}$; (b) $G_2$ with $\Sigma_{o_2} = \{a, b, c, e\}$; (c) $G_3$ with $\Sigma_{o_3} = \{a, b, d, e\}$.

Fig. 2. Faulty automata $F_1$ (a), $F_2$ (b), and $F_3$ (c).

Proceeding in accordance with Step 5 of Algorithm 1, the verifier automata $V_1$, $V_2$ and $V_3$ must be computed. Figures 4(a) and 4(b) show the state transition diagram of verifiers $V_1$ and $V_3$, respectively; the state transition diagram of verifier $V_2$ has been omitted since it leads to a conclusion similar to that drawn from $V_3$.

The verification of robust diagnosability of $L$ with respect to $P_{\alpha_i}$, $i = 1, 2, \ldots, m$, and $\Sigma_f = \{\sigma_f\}$ is carried out in accordance with Theorem 1. We first consider verifier $V_1$ shown in Figure 4(a). Notice that since cycle $(4YD_26N, e_{R_1}, 4YD_26N)$ is formed with an event in $\Sigma_{R_1}$, we may conclude that $L$ is not robustly diagnosable with respect to $P_{\alpha_i}$, $i = 1, 2, \ldots, m$, and $\Sigma_f = \{\sigma_f\}$, in spite of the fact that verifiers $V_2$ (not shown in the paper) and $V_3$ (depicted in Figure 4(b)) do not have any cycles with events in $\Sigma_{R_2}$ and $\Sigma_{R_3}$, respectively, whose first components of their states have fault labels.

A close examination of verifier $V_1$ reveals the traces responsible for the non-robust diagnosability. Notice that trace $s_{V_1} = d_{R_1}\sigma_f c_{R_1} e^n_{R_1}$, where $n$ can be arbitrarily large, takes $V_1$ from its initial state to state $4YD_26N$ and cycles over this state. Since $s_{V_1} = d_{R_1}\sigma_f c_{R_1} e^n_{R_1}$ and $s_{H_1} = P_{H_1}(s_{V_1}) = c_{R_1}b$, then after inverse rename we obtain $s_1 = d_{R_1}\sigma_f c_{R_1} e^n_{R_1}$, where $s_1 = c_{R_1}b$, $s_3 = cb$ and $s_4 = cb$. Furthermore, $P_{R_1}(s_1) = P_{R_2}(s_3) = b$, which implies that when trace $s_1$ occurs, it is not possible to conclude that the system is either in state $4$ of $G_1$, which is after the occurrence of fault event $\sigma_f$, or in state $6$ of $G_3$, which is in a normal path.

6. CONCLUSION

We have proposed a new definition of robust diagnosability that encompasses previously introduced definitions of robust diagnosability. Instead of a single language, we have considered a class of languages generated by a class of automata that model the system behavior. We have given necessary and sufficient conditions for robust diagnosability and developed a polynomial-time algorithm for the verification of the robust diagnosability of a class of languages.

REFERENCES

Fig. 3. Augmented non-faulty automata $H_1$ (a), $H_2$ (b), and $H_3$ (c).


