Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 50-1 (2017) 13630–13635

Diagnosability and online diagnosis of discrete-event systems modeled by acyclic labeled Petri nets

Pedro R. R. Paiva * Lilian Kawakami Carvalho * João Carlos Basilio *

* *Universidade Federal do Rio de Janeiro, Departmental of Electrical Engineering, Rio de Janeiro, RJ, Brazil (e-mail: pedrorrpaiva@gmail.com, lilian@dee.ufrj.br, basilio@dee.ufrj.br).*

Abstract: We address in this paper the problems of online diagnosis and verification of language diagnosability of discrete event systems (DES) modeled by acyclic labeled Petri nets, in which, different transitions can be labeled by the same event (observable, unobservable and failure). The proposed diagnoser makes its decision regarding the failure occurrence by storing the sequence of observed events and, after each occurrence of observable event, it verifiers if two sets of inequalities are satisfied; the first set accounts for the normal whereas the second one accounts for the faulty behavior of the system. We also consider the problem of diagnosability verification by creating new sets of inequalities that, when satisfied, allow us to decide whether the language generated by the Petri net is diagnosable. Our method for online diagnosis has the advantage over previously ones for relying only on the verification of set of inequalities. Regarding language diagnosability, our verification algorithm does not require any knowledge of automaton theory, being self-contained within the Petri net formalism.

Keywords: Discrete event systems, Petri net, Fault diagnosis, Diagnosability.

1. INTRODUCTION

Fault diagnosis of discrete event system has been a very active area of research in recent decades (see Zaytoon and Lafortune (2013) for a fairly complete literature review on failure diagnosis of discrete event system), being mainly concerned with the problem of finding efficient and reliable ways to detect failure occurrences and its isolation. The research activity in this area has been driven by the need of many different application fields such as manufacturing, process control, control systems, transportation, communication networks, software engineering, etc. A failure or an abnormality is defined as any deviation of a system from its normal or desired behavior and are unavoidable in industrial environments currently. Failure diagnosis consists of checking a state for each sequence of observed events, and issue a verdict such as "normal" or "faulty" or "uncertain".

Initial works in this area used automata as modeling formalism. The rise of automaton, although suitable for modeling a system, brings problems of complexity to the diagnosis of large systems. In order to deal with the problem of state explosion, Petri nets have been increasingly used to model DES since it offers significant advantages because of its graphical and mathematical representation. Furthermore, the concept of states (markings) and actions (transitions) reduces the computational complexity involved in the resolution of a diagnostic problem.

Fault diagnosis using Petri nets can be divided in two different approaches (Zaytoon and Lafortune, 2013): the first one relies on reachability analysis of the marking of certain observable places (Hadjicostis and Verghese, 2002; Lefebvre and Delherm, 2007; Miyagi and Riascos, 2006; Ramírez-Treviño et al., 2007); the other approach is based on the set of observable transition (Basile et al., 2009; Benveniste et al., 2003; Cabasino et al., 2011; Dotoli et al., 2009; Genc and Lafortune, 2007; Manyari-Rivera et al., 2007). Our methodology fits in the second approach.

Dotoli et al. (2009) and Basile et al. (2009) consider the problem of online diagnosis by modeling the failures as unobservable transitions, and proposes an online diagnoser which observes sequences of observable events and issues a decision regarding the failure occurrence based on the solution of an integer linear programming problem. Both papers assume that two different transitions cannot be labeled by the same event. Cabasino et al. (2014) addresses failure diagnosability for bounded Petri nets and proposes a verification method inspired by the diagnosability approach for finite state automata proposed by Sampath et al. (1995) and gives necessary and sufficient conditions for diagnosability. Jiroveanu and Boel (2010) has obtained a similar result independently. Manyari-Rivera et al. (2007) proposes an online diagnoser based automaton for DES modeled by bounded Petri nets.

In this paper, we first address the problem of online diagnosis. In this regard, we extend the work by Al-Ajeli and Bordbar (2016) to consider acyclic labeled Petri nets, in which, different transitions can be labeled by the same event (observable, unobservable and fault). The proposed diagnoser makes its decision regarding the failure occurrence by verifying, after each observed event occurrence, if two sets of inequalities are satisfied; the first inequality set accounts for the normal whereas the second inequality set accounts for the faulty behavior of the system. We also consider the problem of diagnosability verification by creating new sets of inequalities that, when satisfied, allow us to decide whether the language generated by the Petri system is diagnosable. Our method for online diagnosis has the advantage over previously proposed ones since it relies

10.1016/j.ifacol.2017.08.2392

only on the verification of inequality sets. Regarding language diagnosability, our verification algorithm does not require any knowledge of automaton theory, being self-contained within the Petri net formalism.

This paper is organized as follows. In section 2, we present preliminary concepts on Petri nets and diagnosability of DES. In Section 3, we derive two sets of inequalities that are obtained from the state equation associated to the Petri net. In Section 4, we propose the online diagnoser that is based on the inequality sets developed in the previous section. In section 5 , we address the problem of language diagnosability, present necessary and sufficient conditions for diagnosability, and a verification algorithm. Finally, in Section 6, we present the conclusions.

2. BACKGROUND

2.1 Petri net

A Petri net is a 4-tuple $\mathscr{N} = (P,T,F,W)$ where, $P = \{p_1,\dots p_m\}$ is a finite set of places, $T = \{t_1,t_2,\dots,t_n\}$ is a finite set of transitions, $F \subseteq (P\times T)\cup(T\times P)$ is a set of arcs and $W : F \to \{1,2,3,\dots\}$ is the weighting function. A state of a Petri net is represented by the marking function $M : P \to \mathbb{N}^m$ that captures the number of tokens in each place; $(\mathscr{N},M_0)$ denotes a Petri net with initial marking M_0. If a marking M is reachable from M_0 through a sequence of transitions $\sigma = t_1 \dots t_k$, denoted as $M_0 \xrightarrow{\sigma} M$, then there exists a vector x such that the following state equation is satisfied:

$$M = M_0 + Ax, \tag{1}$$

where $A = [a_{ij}] \in \mathbb{Z}^{m\times n}$ is the incidence matrix, whose element $a_{ij} = W(j,i) - W(i,j)$, with $W(j,i)$ (resp. $W(i,j)$) being the arc weight from transition j (resp. place i) to place i (resp. transition j), and $x = [x_1 \dots x_n]^T \in \mathbb{N}^n$ is a firing count vector, where x_i represents the number of occurrence of transition $t_i \in T$ in the sequence σ.

A Petri net that has no directed circuits is called acyclic. It is well know that, for acyclic Petri nets, state equation (1) does not have spurious solutions (Murata, 1989).

A labeled Petri net with initial marking is defined as $\mathscr{N} = (P,T,F,W,E,\ell,M_0)$ where $(P,T,F,\ W)$ is a Petri net, E is the set of events, $\ell : T \to E$ is the transition labeling function and M_0 is the initial marking. The language generated by labeled Petri net $\mathscr{N}$ is $\mathscr{L}(\mathscr{N}) = \{\ell(\sigma) \in E^* : (\exists M \in \mathbb{N}^m)[M_0 \xrightarrow{\sigma} M]\}$, where M denotes state reachable from M_0 through Equation (1).

Let $E = E_o \dot\cup E_{uo}$ be a partition of E, where E_o and E_{uo} are the set of observable and unobservable events, respectively. Let T_o and T_{uo} be defined as the set of observable and unobservable transitions, respectively, *i.e.*, $T_o = \{t \in T : (\exists e \in E_o)[\ell(t) = e]\}$ and $T_{uo} = \{t \in T : (\exists e \in E_{uo})[\ell(t) = e]\}$. Two or more transitions $t_1,t_2,\dots,t_n \in T$ are called indistinguishable if they share the same label, i.e., $\ell(t_1) = \ell(t_2) = \dots = \ell(t_n) = e \in E$. An important language operation is the natural projection $P_o : E^* \to (E_o \cup \{\varepsilon\})^*$, that transforms unobservable sequences to the empty sequence ε, *i.e.*, $P_o(e) = \{\varepsilon\}$ for $e \in E_{uo}$, and $P_o(e) = e$, for $e \in E_o$, and $P_o(se) = P_o(s)P_o(e), s \in E^*$ and $e \in E$.

2.2 Diagnosability

Let $E_f \subseteq E_{uo}$ denote the set of failure events, and assume, for the sake of simplicity, that there is only one failure event, *i.e.*, $E_f = \{\ f\}$. In addition, let $\Psi(E_f) = \{s \in \mathscr{L} : (\exists u \in (E \setminus f)^*)[s = uf]\}$ denote the set of all finite traces of L that end with the failure event f. Then, language $\mathscr{L}$ is said to be diagnosable if the occurrence of f can be detected within a finite number of transitions after its occurrence using traces formed with observable events only. Formally, language diagnosability is defined as follows (Sampath et al., 1995).

Definition 1. A prefix-closed and live language $\mathscr{L}$ is diagnosable with respect to projection P_o and $E_f = \{f\}$ if the following holds true:

$$(\exists n \in \mathbb{N})(\forall s \in \Psi(E_f))(\forall t \in \mathscr{L}/s)(\|t\| \geq n \Rightarrow D),$$

where the diagnosability condition D is

$$(\nexists \omega \in \mathscr{L})[(P_o(st) = P_o(\omega)) \wedge (f \notin \omega)] \tag{2}$$

Example 1. Consider the labeled Petri net $\mathscr{N} = (P,T,F,W,E,\ell,M_0)$ shown in Figure 1, where $E = \{a,b,c,d,f\}$ and $M_0 = [10000000000]^T$. The transition labeling function ℓ is defined as follows: $\ell(t_1) = f$, $\ell(t_2) = \ell(t_3) = \ell(t_{10}) = a$, $\ell(t_5) = \ell(t_6) = \ell(t_9) = b$, $\ell(t_7) = c$ $\ell(t_4) = d$ and $\ell(t_8) = e$. Let $E_o = \{a,b\}$ and $E_f = \{f\}$ be the set of observable events and the set of failure events, respectively. Transitions associated

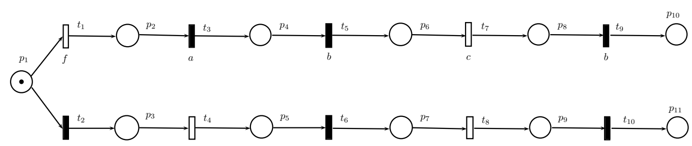

Fig. 1. acyclic labeled Petri Net $(\mathscr{N},M_0)$

with observable events are represented by solid boxes, whereas empty boxes represent transitions associated with unobservable events. These transitions will be referred throughout the text to as observable an unobservable transitions, respectively. The state equation for the Petri net of Figure 1 is given by:

$$M = \begin{bmatrix}1\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix} + \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix}x_1\\x_2\\x_3\\x_4\\x_5\\x_6\\x_7\\x_8\\x_9\\x_{10}\end{bmatrix}$$

Notice that observable transitions t_2, t_3 and t_{10} are indistinguishable because they are all labeled by event a and t_5, t_6, t_9 are also indistinguishable since they are all labeled by event b.

In the example, the occurrence of the sequences of events $s_1 = fab$ or $s_2 = fabc$, or $s_3 = adb$ or $s_4 = adbe$ all produce the sequence ab of observable events, thus we can not be sure if the failure has occurred or not. The occurrence of sequence $s_5 = adbea$, on the other hand, produces the sequence aba of observable events, and, in this case, we are sure that the failure has not occurred because there is no other sequences of events that contain the failure with the same sequence of observable events. The occurrence of sequence $s_6 = fabcb$ produces sequence abb of observable events, and, in this case, we are sure that the failure has occurred because there is no other sequence of events that does not contain the failure and has the same sequence of observable events.