Automatica 83 (2017) 220–225

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

N-diagnosability for active on-line diagnosis in discrete event systems[☆]

Feng Lin[a,1], Le Yi Wang[a], Wen Chen[b], Leitao Han[c], Bin Shen[c]

[a] Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202, USA
[b] Division of Engineering Technology, Wayne State University, Detroit, MI 48202, USA
[c] Zhejiang Provincial Center for Quality Inspection and Testing of Solar Energy Products, Haining, 314416, China

ARTICLE INFO

Article history:
Received 2 March 2016
Received in revised form 2 January 2017
Accepted 28 April 2017

Keywords:
Discrete event systems
Diagnosis
Diagnosability
Fault detection
Fault localization

ABSTRACT

In this paper, we investigate active on-line diagnosis in discrete event systems. Active diagnosis can be used for fault detection, fault localization, fault-tolerant control, among others. Discrete event systems are general models for complex manmade systems. For the active on-line diagnosis, we do not construct the entire diagnostic automaton off-line. Instead, we look N steps ahead to determine active diagnosability and calculate diagnostic strategies. Thus, we define active N-diagnosability and investigate the relation between active diagnosability and active N-diagnosability. We also develop an algorithm to check active N-diagnosability. If a system is actively N-diagnosable, the algorithm will also give the control that diagnoses the system. We show that there are significant computational advantages for using the on-line approach.

1. Introduction

Fault detection, fault localization, and fault-tolerant control are very important in modern engineering systems, as the sizes and complexities of systems increase dramatically (Blanke, Kinnaert, Lunze, & Staroswiecki, 2006; Korbicz, Koscielny, Kowalczuk, & Cholewa, 2003). For example, a networked battery system may consist of thousands of battery cells. When a cell fails, which is very likely, if the fault is not detected and located quickly, then it will cause other cells to fail as well. To detect and locate faults in a complex system, a systematic approach must be taken. Intuitions will not work. Ad-hoc methods will produce inconsistent results.

Many complex and networked systems can be modeled as discrete event systems (DES's) for fault detection and localization. There are several advantages in using discrete event system models. (1) DES models are general. They can represent different types of systems, from networked battery systems to power systems, from computer systems to manufacturing systems. (2) They can be used to solve a large class of diagnosis problems. By properly defining faulty events and/or faulty states, the DES approach can be used for both fault detection and fault localization. (3) DES models are modular. We can build DES models for components first and then use parallel composition.

Because of the advantages of using DES models, diagnosis and diagnosability have been investigated by DES researchers since the 1990s. The results can be divided into two groups: (1) event-based diagnosis and diagnosability, and (2) state-based diagnosis and diagnosability.

Event-based diagnosis and diagnosability are first proposed in Sampath, Sengupta, Lafortune, Sinnamohideen, and Teneketzis (1995), and then extended in Qiu and Kumar (2006), Sampath, Sengupta, Lafortune, Sinnamohideen, and Teneketzis (1996) and others. In this approach, a fault is modeled as an event. Here faults are general notions that may represent a failure, a partial failure, or an abnormality. Faulty events are not observable. Some other events may be observable or unobservable. The goal of diagnosis is to detect the occurrence of a faulty event after observing a finite number of observable events. If this can always be done for all trajectories of a system, then the system is called diagnosable. Methods to check if a system is diagnosable have been proposed. If a system is diagnosable, then a diagnoser can be constructed to diagnose the faults. While many faults can be identified by passively observing the occurrences of observable events, better results may be achieved by actively enforcing some events in the system. Hence, active diagnosis has also been investigated, for example, in Sampath, Lafortune, and Teneketzis (1998). On-line diagnosis is studied in Basile, Chiacchio, and De Tommasi (2009).

☆ The authors of this paper are supported in part by National Science Foundation of USA under Grant 1507096. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Christoforos Hadjicostis under the direction of Editor Christos G. Cassandras.

E-mail addresses: flin@ece.eng.wayne.edu (F. Lin), lywang@wayne.edu (L.Y. Wang), wchenc@wayne.edu (W. Chen), hanleitao@sina.com (L. Han), stcc@haining.gov.cn (B. Shen).

[1] Fax: +1 313 5771101.

http://dx.doi.org/10.1016/j.automatica.2017.06.004
0005-1098/

State-based diagnosis and diagnosability are investigated in Lin (1994) and Lin, Lin, and Lin (1997). In the state-based approach, the state set of a discrete event system is divided into several subsets. One subset represents normal states; and the other subsets represent several faults. Different partitions of states can then be specified depending on the requirements of diagnosis. Based on a given partition, both off-line diagnosability and on-line diagnosability are defined. Algorithms to check diagnosability are also presented in Lin (1994). An application to mixed-signal circuit testing is discussed in Lin et al. (1997).

Results related to diagnosability, including observability (Lin & Wonham, 1988), detectability (Shu & Lin, 2013a,b; Shu, Lin, & Ying, 2007), opacity (Lin, 2011; Paoli & Lin, 2012) are also investigated in discrete event systems.

Recently, we have investigated active diagnosis and diagnosability (Chen, Lin, Wang, Wang, & Xu, 2014). We model a discrete event system using an automaton with outputs. The observation is a mapping from the state set to the output set. Diagnosis requirements are generally specified by a partition on the state set. Assuming that the observation mapping does not provide enough information to determine which partition the system is in, a diagnoser will enforce some controllable events in the system to "drive" the system to some states to determine which partition the system is in. If this can always be done, then the system is said to be actively diagnosable. Active diagnosability depends on the observation mapping, the diagnosis partition, and the controllable events. An algorithm is developed in Chen et al. (2014) to check active diagnosability. If a system is actively diagnosable, the algorithm also provides a control that diagnoses the system. Unfortunately, the complexity of the algorithm is double exponential (Chen et al., 2014).

To decide control on-line and to reduce the computational complexity, we propose an on-line approach for active diagnosis in this paper. We construct an N-step lookahead tree of a system and check if a control exists in the tree that diagnoses the system. If a control exists, then the system is actively N-diagnosable. In the process, the nodes at level N need special attention. We show that if a system is actively N-diagnosable, then it is actively diagnosable. We also find a condition under which the other implication is true. We then develop an algorithm to check active N-diagnosability. The algorithm will also produce a control to actively diagnose the system if the system is actively N-diagnosable. The computational complexity of the algorithm depends on the depth N.

2. Active diagnosis

In this section, we briefly review the results of Chen et al. (2014) and introduce some necessary notations. We model a discrete event system to be diagnosed as an automaton with outputs:

$$G = (Q, \Sigma, \delta, Y, h),$$

where Q is the set of states; Σ is the set of events; Y is the output space; $\delta : Q \times \Sigma \to Q$ is the state transition function; and $h : Q \to Y$ is the output function. The observation is state observation, not event observation as in other papers in the literature (see Section 1). The state observation is to be interpreted as follows. When system G is in state q, $y = h(q)$ is observed.

In most papers on diagnosis of discrete event systems, faults are modeled as events. However, we model faults as states. To diagnose a fault is to identify which state or set of states the system G is in. Depending on the requirements on diagnostics, we partition the state space Q into disjoint subsets (cells) as *normal states*, *fault* 1 *states*, *fault* 2 *states*, etc. The resulting partition is denoted by T. We use $q{=_T}q'$ to denote that q and q' are in the same cell. The states in the same cell are viewed as equivalent as far as faults under consideration are concerned.

By active diagnosis, we mean that a diagnoser can actively drive the system to be diagnosed to certain states by enforcing some controllable events.[2] The set of controllable events is denoted by $\Sigma_c \subseteq \Sigma$. Therefore, a control is a string of controllable events $u \in \Sigma_c^*$. u represents a diagnostic strategy. While the occurrences of controllable events is controlled by a diagnoser, the other events in $\Sigma - \Sigma_c$ can occur in G at any time as long as they are allowed by the state transition function δ. Hence, under a control $u \in \Sigma_c^*$, the set of all strings that can occur in G is

$$L(G, Q_0) \cap P^{-1}(u).$$

In this equation, $L(G, Q_0)$ denotes the language generated by G from the set of possible initial/current states $Q_0 \subseteq Q$, that is,

$$L(G, Q_0) = \{s \in \Sigma^* : (\exists q_o \in Q_0)\delta(q_o, s)!\},$$

where $(q_o, s)!$ means (q_o, s) is defined. $P^{-1}(.)$ is the inverse projection of the natural projection $P : \Sigma^* \to \Sigma_c^*$ (Lin & Wonham, 1988), that is,

$$P^{-1}(u) = \{s \in \Sigma^* : P(s) = u\}.$$

We call the set of possible states that the system G may be in currently the (current) state estimate. The current state estimate is denoted by Q_i ($Q_i \subseteq Q$) and the current output (observation) is denoted by y_i.[3] We update the state estimate if (1) a controllable event σ_{i+1} is enforced by the diagnoser, (2) a new output y_{i+1} is observed, or (3) both (1) and (2) occur. Hence, we use (σ_{i+1}, y_{i+1}) to denote a new control execution, a new output observation, or both as follows. If a new output y_{i+1} is observed without new control execution, then $\sigma_{i+1} = \epsilon$ (the empty string), that is, $(\sigma_{i+1}, y_{i+1}) = (\epsilon, y_{i+1})$. If a new control σ_{i+1} is enforced/executed but no change in the output, then $y_{i+1} = y_i$, that is, $(\sigma_{i+1}, y_{i+1}) = (\sigma_{i+1}, y_i)$. If a new control is executed and a new output is observed, then $\sigma_{i+1} \neq \epsilon$ and $y_{i+1} \neq y_i$. Using this notation, we describe an observed/controlled trajectory as a sequence

$$w = (\sigma_1, y_1)(\sigma_2, y_2)\cdots(\sigma_i, y_i)\cdots.$$

To find state estimates after all possible observed/ controlled trajectories of the system, we define a new diagnostic automaton as follows.

$$\begin{aligned}\tilde{G} &= (X, \tilde{\Sigma}, \xi, x_0)\\ &= Ac(2^Q \times Y, (\Sigma_c \cup \{\epsilon\}) \times Y, \xi, (Q_0, y_0)),\end{aligned}$$

where $Ac(.)$ denotes the accessible part. The event set is $\tilde{\Sigma} = (\Sigma_c \cup \{\epsilon\}) \times Y$. The state set is $X = 2^Q \times Y$. The initial/current state is $x_0 = (Q_0, y_0)$. The state transition function $\xi : X \times \tilde{\Sigma} \to X$ is defined as follows. For $x = (Q_i, y_i)$ and $\tilde{\sigma} = (\sigma_{i+1}, y_{i+1})$, if $\sigma_{i+1} = \epsilon \wedge y_{i+1} = y_i$, then $\xi(x, \tilde{\sigma})$ is undefined, otherwise

$$\xi(x, \tilde{\sigma}) = (SOR(NOR((Q_i, y_i), (\sigma_{i+1}, y_{i+1})), y_{i+1}), y_{i+1}).$$

In the above equation, SOR and NOR are defined as follows.

$$\begin{aligned}&NOR((Q_i, y_i), (\sigma_{i+1}, y_{i+1}))\\ &\quad = \{q \in Q : (\exists q' \in Q_i)\delta(q', \sigma_{i+1}) = q \wedge h(q) = y_{i+1}\}\\ &SOR((Q_{i+0.5}, y_{i+1}))\\ &\quad = \{q \in Q : (\exists q' \in Q_{i+0.5})(\exists s \in (\Sigma - \Sigma_c)^*)\delta(q', s) = q\\ &\qquad \wedge (\forall t \leq s)h(\delta(q', t)) = y_{i+1}\},\end{aligned}$$

[2] The definition of controllable events in this paper is different than supervisory control of discrete event systems, where an event is controllable if it can be disabled.

[3] The current state estimate must be consistent with the current observation, that is, $Q_i \subseteq h^{-1}(y_i)$.