Detectability of Discrete Event Systems

Shaolong Shu, Feng Lin, and Hao Ying

***Abstract*—In this note, we investigate the detectability problem in discrete event systems. We assume that we do not know initially which state the system is in. The problem is to determine the current and subsequent states of the system based on a sequence of observations. The observation includes partial event observation and/or partial state observation, which leads to four possible cases. We further define four types of detectabilities: strong detectability, (weak) detectability, strong periodic detectability, and (weak) periodic detectability. We derive necessary and sufficient conditions for these detectabilities. These conditions can be checked by constructing an observer, which models the estimation of states under different observations. The theory developed in this note can be used in feedback control and diagnosis. If the system is detectable, then the observer can be used as a diagnoser to diagnose the failure states of the system.**

***Index Terms*—Detectability, discrete event systems, observability, state estimation.**

I. Introduction

Discrete event systems have been studied for more than twenty years. Various problems have been investigated, especially those problems related to supervisory control [1], [3], [8], such as controllability, observability, coobservability, and normality. However, there is one problem that has not been fully investigated. This is the problem of how to estimate or determine the current and subsequent states of the system based on observations. The partial reason that this problem has not been fully investigated is that what is important in supervisory control is the information on sequences or traces of events. That is why observability of discrete event systems is defined on traces rather than on states [3]. We say that a language representing the desired behavior of a discrete event system is observable if for any two traces in the language that look the same to a controller (that is, they have the same projection), the control action following these two traces is consistent (that is, it cannot be the case that an event is desirable after one trace but not desirable after another). Obviously, the above definition of observability is unrelated to the estimation of states. Similarly, coobservability and normality (a stronger version of observability) are also unrelated to the estimation of states.

However, there are some applications of discrete event systems where the state estimation problem is important. We realize this especially in medical applications that we are investigating [4], [5]. In medical applications, it is important to know the state of a system (representing, for example, the disease stage of a patient). State estimation is also important in diagnosis. One approach to the diagnosis problem is to model failures as unobservable events and diagnosability requires that one can determine the occurrence of a failure event after finite number of event observations [9], [10]. However, we can also approach the diagnosability problem by defining some failure states [2], [11]. Then the diagnosability problem becomes a problem of estimating states. Yet another application of the state estimation problem is in remote and distributed systems, where it is desirable for a central station to be able to determine the state of a remote system under limited communications. For all these applications, in this note, we will investigate the state estimation problem.

State estimation is first studied using automata in an abstract sense in [12] and the idea is followed by many others, including us. In that sense, our note can be viewed as an extension of the work presented in [12]. However, our note is more specific and investigates state estimation in a general discrete event system framework. State estimation problem is also studied in [7] and [6]. Our work is different from [7] and [6] because we investigate the state estimation problem from all possible angles which includes the results of [7] and [6] as special cases. In particular, we define two types of detectabilities: detectability where we know the current and subsequent states of the system after some finite number of observations and periodic detectability where we know the state of the system periodically. One reason for defining periodical detectability is that when not all events are observable, knowing the current state does not imply knowing the subsequent states. We consider both strong detectability, where state can be determined for all possible trajectories of the system, and (weak) detectability, where state can be determined for some trajectories of the system. One reason for defining (weak) detectability is that if the system is weakly detectable, then we may be able to control it within certain trajectories so that it is strongly detectable.

In general, there are two types of outputs: event output and state output. For event output, we assume, as in supervisory control, that some events are observable and some are not. For state observation, we assume that there is a many-to-one output mapping from the state set to a state output set. Therefore, by observing state output, we know which subset of states the system is in, but we do not know exactly which state it is in because the output mapping is not one-to-one (otherwise the state estimation problem is trivial). Thus, there are four possible cases of event and state observations: (1) All events are observable and no state is observable. (2) All events are observable and some states are observable. (3) Some events are observable and no state is observable. (4) Some events are observable and some states are observable. The detectability when all events are observable and partial states are observable (Case 2) was partly investigated in [7] and the periodic detectability when some events are observable and no state is observable (Case 3) was partly investigated in [6], which they called observability. We investigate both detectability and periodic detectability for all the above four cases. To the best of our knowledge, our results have not been obtained before.

The note is organized as follows. In Section II, we present the model of discrete event systems and the output mechanisms. In Section III, we present our main results on four types of detectabilities for systems with partial event observation and some state observation. Conclusions are given in Section IV.

Due to page limitations, we only present Case 4 here. The results for other cases can be found in the full version of this note available at http://ece.eng.wayne.edu/~flin.

II. Discrete Event Systems

Discrete event systems are used to model systems with discrete states and events. States represent conditions and status of a system. For example, the states of a machine may consist of idle, working, and down; the states of a patient may include excellent, fair, and poor. Events represent changes in the system, action taken by external agents, and other

Manuscript received March 8, 2006; revised November 6, 2006; July 23, 2007; and August 17, 2007. Recommended by Associate Editor S. Haar. This work was supported in part by the NSF under Grants INT-0213651 and ECS-0624828, and by the NIH under Grant 1 R21 EB001529-01A1.

S. Shu is with the School of Electronics and Information Engineering, Tongji University, Shanghai, China.

F. Lin is the School of Electronics and Information Engineering, Tongji University, Shanghai, China and also with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202 USA.

H. Ying is with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2007.910713

activities of significance. For example, turning on or off a machine is an event; so is machine breaking down or being repaired. Similarly, improvement or deterioration of patient's conditions is an event; so is administrating a drug or treating a patient. To model a discrete event system, we often use an automaton (also called state machine or generator) [1]

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

where Q is the set of discrete states, Σ the set of events, and $\delta : Q \times \Sigma \to Q$ the transition function describing what event can occur at one state and the resulting new state. An equivalent way to define the transition function is to specify the set of all possible transitions: $\{(q, \sigma, q') : \delta(q, \sigma) = q'\}$. With a slight abuse of notation, we will also use δ to denote the set of all possible transitions and write $(q, \sigma, q') \in \delta$ if $\delta(q, \sigma) = q'$ is defined.

In many applications of discrete event systems, it is desirable to know the current state of the system. If we do not know the current state of G, we need to estimate it. The estimation is based on observation of some events and/or some states. The event observation is described by the projection $P : \Sigma^* \to \Sigma_o^*$, where Σ_o is the set of observable events. The state observation is described by the output map $h : Q \to Y$, where Y is a (finite) output set. The discrete event system with the event and state observation is described by

$$(G, P, h, \Sigma_o, Y)$$

where $G = (Q, \Sigma, \delta)$. The question is whether we can estimate the current and subsequent states based on the event and state observations.

To avoid unnecessarily complicated technicalities in our ensuing development, we will assume that $G = (Q, \Sigma, \delta)$ is deadlock free, that is, for any state of the system, at least one event is defined at that state $(\forall q \in Q)(\exists \sigma \in \Sigma)\delta(q, \sigma)$ is defined. This assumption is also made in [6] and can be relaxed at the expense of more complicated notations and proofs. We also make another assumption that no infinite strings exist whose events are all unobservable. In other words, no loops in G contain only unobservable events.

III. STATE ESTIMATION AND DETECTABILITIES

We consider the state estimation problem for systems with partial event observations and some state observations. The problem can be stated as follows.

State Estimation Problem: Given a discrete event system

$$(G, P, h, \Sigma_o, Y)$$

we do not know the initial state of $G = (Q, \Sigma, \delta)$. We have partial event observations and some state observations, that is, $\Sigma_o \subset \Sigma$ and $Y \neq$ emptyset. Can we determine the current state and the subsequent states of the system after a finite number of observations? To formalize the problem, let us define the following four properties.

Strong Detectability: A discrete event system (G, P, h, Σ_o, Y) is strongly detectable if we can determine the current state and the subsequent states of the system after a finite number of observations for all trajectories of the system.

(Weak) Detectability: A discrete event system (G, P, h, Σ_o, Y) is (weakly) detectable if we can determine the current state and the subsequent states of the system after a finite number of observations for some trajectories of the system.

Strong Periodic Detectability: A discrete event system (G, P, h, Σ_o, Y) is strongly periodically detectable if we can periodically determine the current state of the system for all trajectories of the system.

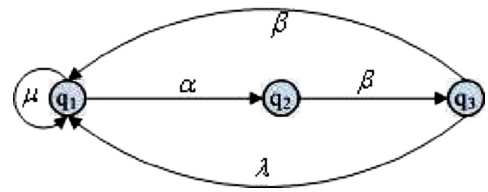

Fig. 1. System to illustrate various detectabilities.

(Weak) Periodic Detectability: A discrete event system (G, P, h, Σ_o, Y) is periodically detectable if we can periodically determine the current state of the system for some trajectories of the system.

To illustrate the differences among the four types of detectabilities, let us consider an example of the system showed in Fig. 1. β is the unobservable event and the other events are all observable. We assume that there is not state output. Intuitively, if the system executes μ^*, we will observe μ^* and know that the system is in state q_1 at all time. If the system executes $(\alpha\beta\lambda)^*$, we will observe $(\alpha\lambda)^*$ and know that the system is in state q_1 periodically (after seeing λ). If the system executes $(\alpha\beta\beta)^*$, we will observe α^* and in this case we do not know where the system is. According to our definitions, the system in Fig. 1 is detectable but not strongly detectable. Furthermore, if we remove μ from the system, then the system is periodically detectable but not strongly periodically detectable.

Formally, the procedure to check the above four types of detectabilities is as follows.

Step 1, we add an initial state q_o to G. We extend the event set from Σ to $(\Sigma \cup \{\phi\}) \times Y \cup \{\varepsilon\}$. For all states $q \in Q$, we add transitions from q_o to q with label (ϕ, y), where $y = h(q)$. We relabel the transitions (q', σ, q) in G to take into account of observation. We need to consider two types of transitions: For observable transitions $\sigma \in \Sigma_o$, we relabel as $(q', (\sigma, h(q)), q)$. For unobservable transitions $\sigma \notin \Sigma_o$, we relabel (q', σ, q) as $(q', (\phi, h(q)), q)$ if $h(q') \neq h(q)$ and as (q', ε, q) if $h(q') = h(q)$. The reason for the above relabeling is that if $h(q') \neq h(q)$, then we know that some event has occurred because the system has changed states, although we do not know which event has occurred; if $h(q') = h(q)$, then nothing will be observed. Formally

$$G_{\rm ps,nd} = (Q \cup \{q_o\}, (\Sigma_o \cup \{\phi\}) \times Y \cup \{\varepsilon\}, \delta_{\rm ps,nd}, q_o)$$

where $\delta_{\rm ps,nd} = \{(q_o, (\phi, h(q)), q) : q \in Q\} \cup \{(q', (\sigma, h(q)), q) : (q', \sigma, q) \in \delta \wedge \sigma \in \Sigma_o\}$

$$\cup \{(q', (\phi, h(q)), q) : (q', \sigma, q) \in \delta \wedge \sigma \notin \Sigma_o \wedge h(q') \neq h(q)\}$$
$$\cup \{(q', \varepsilon, q) : (q', \sigma, q) \in \delta \wedge \sigma \notin \Sigma_o \wedge h(q') = h(q)\}.$$

In other words, $\delta_{\rm ps,nd}$ is a mapping $\delta_{\rm ps,nd} : (Q \cup \{q_o\}) \times ((\Sigma_o \cup \{\phi\}) \times Y \cup \{\varepsilon\}) \to 2^{Q \cup \{q_o\}}$, which can be easily extended to

$$\delta_{\rm ps,nd} : (Q \cup \{q_o\}) \times ((\Sigma_o \cup \{\phi\}) \times Y \cup \{\varepsilon\})^* \to 2^{Q \cup \{q_o\}}.$$

Step 2, we convert the nondeterministic automaton $G_{\rm ps,nd}$ into a deterministic automaton $G_{\rm ps,obs}$. Since $G_{\rm ps,nd}$ has unobservable transition (q', ε, q), we need first define the unobservable reach from a subset of states $x \subseteq Q \cup \{q_o\}$ as follows:

$${\rm UR}(x) = x \cup \{q \in Q \cup \{q_o\} : (\exists q' \in x) q \in \delta_{\rm ps,nd}(q', \varepsilon)\}.$$

Then we can define

$$G_{\rm ps,obs} = Ac(X, (\Sigma_o \cup \{\phi\}) \times Y, \xi_{\rm ps}, x_{\rm ps,o})$$

where Ac denotes the accessible (reachable) part of an automaton, $X = 2^{Q\cup\{q_o\}}, \xi_{\rm ps}(x,(\sigma,y)) = {\rm UR}(\{q \in Q\cup\{q_o\} : (\exists q' \in x)(q',(\sigma,y),q) \in \delta_{\rm ps,nd}\})$, and $x_{\rm ps,o} = \{q_o\}$.

The reason for constructing $G_{\rm ps,obs}$ is to describe the estimate of possible states of the system as shown in the following lemmas.

Lemma 1: If the current estimate of possible states of G is $x \in X$ (that is, $x \subseteq Q\cup\{q_o\}$), an event $\sigma \in \Sigma$ occurs, then either σ or nothing (ϕ) is observed and state output $y \in Y$ is also observed. The next estimate of possible states of G is $x' = {\rm UR}(\{q \in Q\cup\{q_o\} : (\exists q' \in x)(q',(\sigma,y),q) \in \delta_{\rm ps,nd}\})$, where σ could be ϕ.

Proof: If $\sigma \neq \phi$, then

$$\begin{aligned} x' &= {\rm UR}(\{q \in Q\cup\{q_o\} : (\exists q' \in x)(q',(\sigma,y),q) \in \delta_{\rm ps,nd}\}) \\ &= {\rm UR}(\{q \in Q\cup\{q_o\} : (\exists q' \in x)(q',\sigma,q) \\ &\qquad \in \delta \wedge \sigma \in \Sigma_o \wedge h(q) = y\}). \end{aligned}$$

That is, x' consists of all states in the unobservable reach of the set of states that can be reached from a state in x if event $\sigma \in \Sigma_o$ and $y \in Y$ is observed.

If $\sigma = \phi$, then

$$\begin{aligned} x' &= {\rm UR}(\{q \in Q\cup\{q_o\} : (\exists q' \in x)(q',(\phi,y),q) \in \delta_{\rm ps,nd}\}) \\ &= {\rm UR}(\{q \in Q\cup\{q_o\} : (\exists q' \in x) \\ &\qquad \times ((q',\sigma,q) \in \delta \wedge \sigma \notin \Sigma_o \wedge h(q) \\ &\qquad\quad = y \neq h(q')) \vee (q' = q_o \wedge h(q) = y)\}). \end{aligned}$$

That is, x' consists of all states in the unobservable reach of the set of states that can be reached from a state in x if no event is observed but a new state output $y \in Y$ is observed.

Lemma 2:

1) The initial estimate of possible states of G after state output $y \in Y$ is observed and before any event occurs is $\xi_{\rm ps}(x_{\rm ps,o},(\phi,y)) = {\rm UR}\{q \in Q\cup\{q_o\} : (q_o,(\phi,y),q) \in \delta_{\rm ps,nd}\}$.
2) The estimate of possible states of G after transitions $\sigma_0\sigma_1\sigma_2\ldots\sigma_k$ (σ_i could be ϕ) and state outputs $y_0y_1y_2\ldots y_k$ are observed is $\xi_{\rm ps}(x_{\rm ps,o},(\sigma_0,y_0)(\sigma_1,y_1)(\sigma_2,y_2)\ldots(\sigma_k,y_k))$.

Proof:

1) By the definition of $G_{\rm ps,nd}, \xi_{\rm ps}(x_{\rm ps,o},(\phi,y)) = {\rm UR}\{q \in Q\cup\{q_o\} : (q_o,(\phi,y),q) \in \delta_{\rm ps,nd}\}$ is the set of all state possible from q_o when state output $y = h(q)$ is observed before any event occurs. Hence, it is the initial estimate.
2) By the definition of $G_{\rm ps,nd}$ and repeated applications of Lemma 1.

Step 3, we mark the states in that contain singleton state and denote this set by $X_m = \{x \in X : |x| = 1\}$, where $|x|$ denotes the number of elements in x.

Step 4, we remove states in X_m from $G_{\rm ps,obs}$ and denote the automaton remained after removing the states in X_m as $G^{\rm rem}_{\rm ps,obs}$

$$G^{\rm rem}_{\rm ps,obs} = Ac(X - X_m, \Sigma_o, \xi_{\rm ps}|_{X-X_m}, x_{\rm ps,o}).$$

Step 5, since there is no deadlock state in the observer, we can now check detectability with partial event observation and partial state observation using the following criterions.

Theorem 1 (Criterion for Checking Strong Detectability): A discrete event system (G,P,h,Σ_o,Y) with $\Sigma_o \subset \Sigma$ and $Y \neq$ emptyset is strongly detectable if and only if all loops in $G_{\rm ps,obs}$ are entirely within X_m.

Proof: By Lemma 2, the observer $G_{\rm ps,obs}$ describes the estimate of states the system may be in. When $G_{\rm ps,obs}$ enters states in X_m, we know exactly which state the system is in. By our assumptions, the observer $G_{\rm ps,obs}$ is deadlock free. Since $G_{\rm ps,obs}$ is finite, after some finite observations, $G_{\rm ps,obs}$ must enter some loops. If all loops in $G_{\rm ps,obs}$ are entirely within X_m, then the current state and the subsequent states of the system are known no matter which trajectory the system follows, that is, the system is strongly detectable. On the other hand, if there exists some loops that are not entirely within X_m, then the system can follow those loops and hence is not strongly detectable.

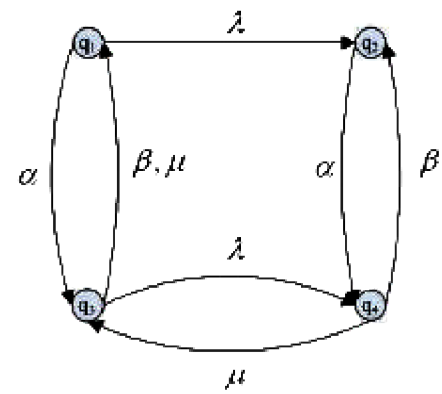

Fig. 2. System whose detectabilities are to be determined.

Theorem 2 (Criterion for Checking Detectability): A discrete event system (G,P,h,Σ_o,Y) with $\Sigma_o \subset \Sigma$ and $Y \neq$ emptyset is detectable if and only if there are loops in X_m.

Proof: If X_m is not empty then it is accessible from the initial state in $G_{\rm ps,obs}$. So for any state in X_m, there must exist some trajectories for which the system can enter that state. Furthermore, if there are loops in X_m, then any such loop can produce at least one infinite string by which the system can always stay within X_m. Hence, the system is detectable by Lemma 2. On the other hand, if there are no loops in X_m, the system will eventually leave X_m along any trajectory of the system. Hence, the system is not detectable.

Theorem 3 (Criterion for Checking Strongly Periodic Detectability): A discrete event system (G,P,h,Σ_o,Y) with $\Sigma_o \subset \Sigma$ and $Y \neq$ emptyset is strongly periodically detectable if and only if there are no loops in $G^{\rm rem}_{\rm ps,obs}$.

Proof: The condition of no loops in $G^{\rm rem}_{\rm ps,obs}$ ensures that the system cannot always stay in $X - X_m$. Therefore, the system must visit periodically. This implies that we can periodically determine the current state of the system for all trajectories of the system by Lemma 2. On the other hand, if there are loops in $G^{\rm rem}_{\rm ps,obs}$, then the system may stay in the loop forever. Hence, the system is not strongly periodically detectable.

Theorem 4 (Criterion for Checking Periodic Detectability): A discrete event system (G,P,h,Σ_o,Y) with $\Sigma_o \subset \Sigma$ and $Y \neq$ emptyset is periodically detectable if and only if there are loops in $G_{\rm ps,obs}$ which include at least one state belonging to X_m.

Proof: If there are loops in $G_{\rm ps,obs}$ which include at least one state belonging to X_m, then the system can stay in this loop, and we can periodically determine the current state of the system for some trajectories of the system by Lemma 2. If no such loops exist, then the system is not periodically detectable.

Let us apply the above theorems and determine detectabilities of the system in Fig. 2. We first assume that event μ is not observable and the state output is $h(q_1) = 1, h(q_2) = 2, h(q_3) = 3, h(q_4) = 2$. $G_{\rm ps,nd}$ and $G_{\rm ps,obs}$ are calculated as shown in Figs. 3 and 4, respectively. From Fig. 4, we can see that the system is strongly detectable (and hence detectable, strongly periodic detectable, and periodic detectable). Now we consider the second observation with μ observable but λ unobservable

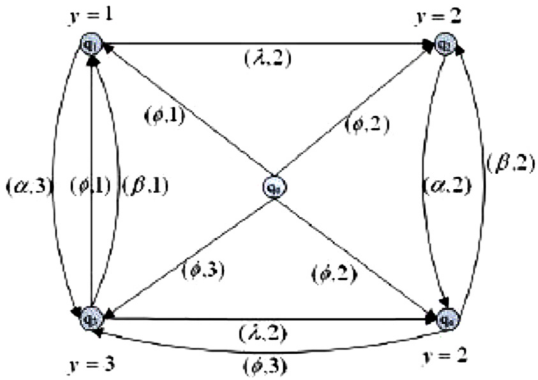

Fig. 3. $G_{\mathrm{ps,nd}}$ of the system in Fig. 2 with the first observation.

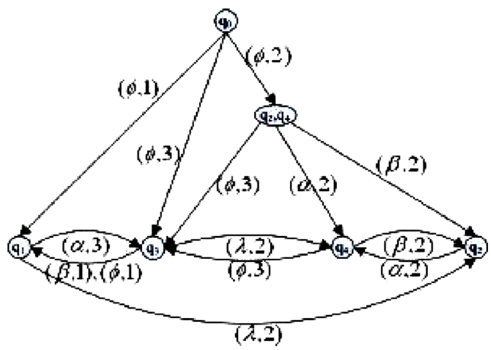

Fig. 4. $G_{\mathrm{ps,obs}}$ for the system in Fig. 3.

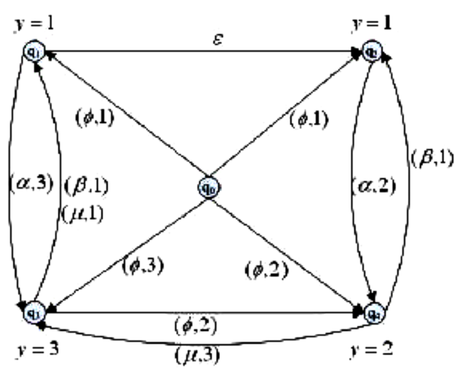

Fig. 5. $G_{\mathrm{ps,nd}}$ of the system in Fig. 2 with the second observation.

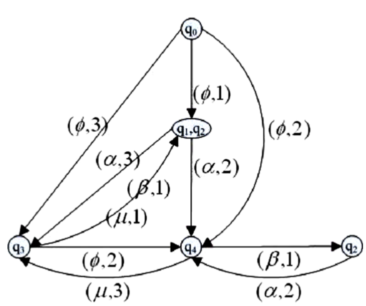

Fig. 6. $G_{\mathrm{ps,obs}}$ for the system in Fig. 5.

and state observation to $h(q_1) = 1, h(q_2) = 1, h(q_3) = 3, h(q_4) = 2$. $G_{\mathrm{ps,nd}}$ and $G_{\mathrm{ps,obs}}$ are now shown in Figs. 5 and 6 respectively. It can be checked that the system is strongly periodic detectable (and, hence, periodic detectable), detectable, but not strongly detectable.

IV. Conclusion

In this note, we considered both event observation and state observation. We defined detectability and periodic detectability, both in a strong sense and in a weak sense. We constructed an observer, whose roles are to estimate the states of a system after a sequence of observation. We derived computable criterions for checking necessary and sufficient conditions for various types of detectability. These definitions and criterions extend the results of [6] and [7] significantly and covered most cases encountered in practice.

References

[1] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. New York: Kluwer, 1999.

[2] F. Lin, "Diagnosability of discrete event systems and its applications," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 4, no. 1, pp. 197–212, 1994.

[3] F. Lin and W. M. Wonham, "On observability of discrete event systems," *Inf. Sci.*, vol. 44, no. 3, pp. 173–198, 1988.

[4] F. Lin and H. Ying, "Modeling and control of fuzzy discrete event systems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 32, pp. 408–415, 2002.

[5] F. Lin, H. Ying, R. D. MacArthur, J. A. Cohn, D. C. Barth-Jones, H. Ye, and L. R. Crane, "Theory for a control architecture of fuzzy discrete event systems for decision making," *Inf. Sci.*, vol. 177, pp. 3749–3763, 2007.

[6] C. M. Ozveren and A. S. Willsky, "Observability of discrete event dynamic systems," *IEEE Trans. Autom. Control*, vol. 35, no. 7, pp. 797–806, 1990.

[7] P. J. Ramadge, "Observability of discrete event systems," in *Proc. CDC*, Dec. 1986.

[8] R. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Control Optim.*, vol. 25, no. 1, pp. 206–230, 1987.

[9] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete event systems," *IEEE Trans. Autom. Control*, vol. 40, no. 9, Sep. 1995.

[10] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Failure diagnosis using descrete event models," *IEEE Trans. Control Syst. Technol.*, vol. 4, no. 2, Mar. 1996.

[11] S. Hashtrudi Zad, R. H. Kwong, and W. M. Wonham, "Fault diagnosis in discrete-event systems: Framework and model reduction," *IEEE Trans. Autom. Control*, vol. 48, no. 7, pp. 1199–1212, Jul. 2003.

[12] W. M. Wonham, "Towards an abstract internal model principle," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 11, pp. 735–740, 1976.