A Polynomial Time Algorithm for Diagnosability of Discrete Event Systems

A Polynomial Algorithm for Testing Diagnosability of Discrete Event Systems

Shengbing Jiang, Zhongdong Huang, Vigyan Chandra, and Ratnesh Kumar
Department of Electrical Engineering
University of Kentucky
Lexington, KY 40506-0046
{sjian0, zhdhuang, vigyan, kumar}@engr.uky.edu

Abstract

Failure diagnosis in large and complex systems is a critical task. In the realm of discrete event systems, Sampath *et al.* proposed a language based failure diagnosis approach. They introduced the diagnosability for discrete event systems and gave a method for testing the diagnosability by first constructing a diagnoser for the system. The complexity of this method of testing diagnosability is exponential in the number of states of the system and doubly exponential in the number of failure types. In this paper, we give an algorithm for testing diagnosability that does not construct a diagnoser for the system, and its complexity is of 4th order in the number of states of the system and linear in the number of the failure types.

Keywords: Discrete event system, failure diagnosis, diagnosability, complexity.

1 Introduction

Failure diagnosis is a critical task in large and complex systems. This problem has received considerable attention in the literature of various domains including the discrete event systems [1, 2, 3, 4, 5, 6]. In [4], Sampath *et al.* proposed a failure diagnosis approach for discrete event systems. They introduced the notion of diagnosability and gave a necessary and sufficient condition for testing it. Their condition is expressed as a property of the diagnoser of the system. In order to test the diagnosability, the diagnoser needs to be constructed first. The complexity of constructing the diagnoser and testing the diagnosability is exponential in the number of states of the system and doubly exponential in the number of failure types.

It is clear that if we could test more efficiently whether or not a system is diagnosable without having to construct a diagnoser, it would save us the time involved in constructing a diagnoser for the system which may not be diagnosable. In this paper, we give a method for testing the diagnosability without having to construct a diagnoser. The complexity of our method is polynomial in the number of states of the system and also in the number of failure types.

In the rest of the paper, we first introduce the notion of diagnosability of discrete event systems, then present our algorithm for testing it. Finally, an illustrative example is provided.

2 Diagnosability

We first give the system model and then define the diagnosability as introduced by [4].

2.1 System model

Let $G = (X, \Sigma, \delta, x_0)$ be a finite state machine model of the system to be diagnosed, where

- X is a finite set of states;
- Σ is a finite set of event labels;
- $\delta \subseteq X \times \Sigma \times X$ is a finite set of transitions;
- $x_0 \in X$ is the initial state.

We assume that all state machines are accessible (all states can be reached from the initial state), and otherwise we consider only the accessible part of the state machine. We let Σ^* denote the set of all finite length event sequences, including the zero length sequence denoted ϵ. An element of Σ^* is called a trace, and a subset of Σ^* is called a language. For a trace s

and an event σ, we write $\sigma \in s$ to imply that σ is an event contained in the trace s. A path in G is a sequence of transitions $(x_1, \sigma_1, x_2, \dots, \sigma_n, x_n)$ such that for each $i \in \{1, \dots, n-1\}$, $(x_i, \sigma_i, x_{i+1}) \in \delta$; this path is a cycle if $x_n = x_1$. We use $L(G) \subseteq \Sigma^*$ to denote the generated language of G, i.e., the set of traces that can be executed in G starting from its initial state. Then $L(G)$ is prefix-closed, i.e., $L(G) = pr(L(G))$, where $pr(L(G)) = \{u | \exists v \in \Sigma^*, uv \in L(G)\}$ is the set of prefixes of traces in $L(G)$. Let $\Sigma_o \subseteq \Sigma$ denote the set of observable events, $\Sigma_{uo} = \Sigma - \Sigma_o$ be the set of unobservable events, $M : \Sigma \rightarrow \Sigma_o \cup \{\epsilon\}$ be the observation mask, $\mathcal{F} = \{F_i, i = 1, 2, \dots, m\}$ be the set of failure types, $\psi : \Sigma \rightarrow \mathcal{F} \cup \{\emptyset\}$ be the failure assignment function for each event in Σ. The definition of M is extended from Σ to Σ^* inductively as follows: $M(\epsilon) = \epsilon$ and for each $s \in \Sigma^*, \sigma \in \Sigma : M(s\sigma) = M(s)M(\sigma)$.

We make the following assumptions as in [4] for the system studied in this paper.

A1 The language $L(G)$ generated by G is live. This means that there is a transition defined at each state x in X.

A2 There does not exist in G any cycle of unobservable events, i.e., $(\exists k \in N)$ $(\forall ust \in L(G), s \in \Sigma_{uo}^*) \Rightarrow ||s|| \leq k$, where N denotes the set of natural numbers, and $||s||$ denotes the length of trace s.

A3 Every failure event is unobservable, i.e., $(\forall \sigma \in \Sigma,\ \psi(\sigma) \neq \emptyset) \Rightarrow M(\sigma) = \epsilon$.

2.2 Diagnosability

The diagnosability for discrete event systems defined in [4] is described as follows:

Definition 1 A prefix-closed language L is said to be diagnosable with respect to the observation mask M and the failure assignment function ψ if the following holds:

$$(\forall F_i \in \mathcal{F})\ (\exists n_i \in N)\ (\forall s \in L, \psi(s_f) = F_i)\ (\forall v = st \in L, ||t|| \geq n_i)$$
$$\Rightarrow (\forall w \in L, M(w) = M(v))\ (\exists u \in pr(\{w\}), \psi(u_f) = F_i),$$

where s_f and u_f denote the last events in traces s and u respectively, $pr(\{w\})$ is the set of all prefixes of w. A system G is said to be diagnosable if its language $L(G)$ is diagnosable.

The above definition states that if s is a trace in L ending with a F_i-type failure, and v is a sufficient long (at least n_i events longer) trace obtained by extending s in L, then every trace w in L that is observation equivalent to v, i.e., $M(w) = M(v)$, should contain in it a F_i-type failure.

3 Algorithm

We now present the algorithm for testing the diagnosability.

Algorithm 1 For a given system $G = (X, \Sigma, \delta, x_0)$ with an observation mask M and a failure assignment function ψ, do the following:

1. Obtain a nondeterministic finite state machine $G_o = (X_o, \Sigma_o, \delta_o, x_0^o)$ with language $L(G_o) = M(L(G))$ as follows:
    - $X_o = \{(x, f) \mid x \in X_1 \cup \{x_0\}, f \subseteq \mathcal{F}\}$ is the finite set of states, where $X_1 = \{x \in X \mid \exists (x', \sigma, x) \in \delta \text{ with } M(\sigma) \neq \epsilon\}$ is the set of states in G that can be reached through an observable transition, and f is the set of failure types along certain paths from x_0 to x.
    - Σ_o, the set of observable events, is the set of event labels for G_o.
    - $\delta_o \subseteq X_o \times \Sigma_o \times X_o$ is the set of transitions. $((x, f), \sigma, (x', f')) \in \delta_o$ if and only if there exists a path $(x, \sigma_1, x_1, \ldots, \sigma_n, x_n, \sigma, x')$ $(n \geq 0)$ in G such that $\forall i \in \{1, 2, \ldots, n\}, M(\sigma_i) = \epsilon$, $M(\sigma) = \sigma$, and $f' = \{\psi(\sigma_i) \mid \psi(\sigma_i) \neq \emptyset, 1 \leq i \leq n\} \cup f$.
    - $x_0^o = (x_0, \emptyset) \in X_o$ is the initial state.
2. Compute $G_d = (G_o || G_o)$, the strict composition of G_o with itself. Then
   $G_d = (X_d, \Sigma_o, \delta_d, x_0^d)$, where
    - $X_d = \{(x_1^o, x_2^o) \mid x_1^o, x_2^o \in X_o\}$ is the set of states.
    - Σ_o is the set of event labels for G_d.

- $\delta_d \subseteq X_d \times \Sigma_o \times X_d$ is the set of transitions. $((x_1^o, x_2^o), \sigma, (y_1^o, y_2^o)) \in \delta_d$ if and only if both (x_1^o, σ, y_1^o) and (x_2^o, σ, y_2^o) are in δ_o.
- $x_0^d = (x_0^o, x_0^o) \in X_d$ is the initial state.

3. Check whether there exists in G_d a cycle $cl = (x_1, \sigma_1, x_2, \ldots, x_n, \sigma_n, x_1)$, $n \geq 1$, $x_i = ((x_i^1, f_i^1), (x_i^2, f_i^2))$, $i = 1, 2, \ldots, n$, such that $f_1^1 \neq f_1^2$. If the answer is yes, then output that the system is not diagnosable; otherwise output that the system is diagnosable. This last step can be performed by first identifying states $((x^1, f^1), (x^2, f^2))$ in G_d for which $f^1 \neq f^2$, and deleting all the other states and the associated transitions; and next checking if the remainder graph contains a cycle.

In the following, we give two Lemmas showing some properties of the state machines G_o and G_d derived in Algorithm 1. The proofs are omitted here because they follow directly from the definitions of G_o and G_d.

Lemma 1 For the state machine G_o the following holds:

1. $L(G_o) = M(L(G))$.
2. For every path tr in G_o ending with a cycle,

$$tr = ((x_0, \emptyset), \sigma_0, (x_1, f_1), \ldots, (x_k, f_k), \sigma_k, \ldots, (x_n, f_n), \sigma_n, (x_k, f_k)),$$

we have

- $f_i = f_j$ for any i and j in $\{k, k+1, \ldots, n\}$.
- $\exists uv^* \in L(G)$ such that $M(u) = \sigma_0 \ldots \sigma_{k-1}$, $M(v) = \sigma_k \ldots \sigma_n$, and $\{\psi(\sigma) \mid \sigma \in u, \psi(\sigma) \neq \emptyset\} = \{\psi(\sigma) \mid \sigma \in uv, \psi(\sigma) \neq \emptyset\} = f_k$.

Lemma 2 For every path tr in G_d ending with a cycle,

$$tr = (x_0^d, \sigma_0, x_1, \ldots, x_k, \sigma_k, \ldots, x_n, \sigma_n, x_k),$$

$x_i = ((x_i^1, f_i^1), (x_i^2, f_i^2))$, $i = 1, 2, \ldots, n$, we have

1. there exist two paths tr_1 and tr_2 in G_o ending with cycles, namely,

$$tr_1 = ((x_0, \emptyset), \sigma_0, (x_1^1, f_1^1), \ldots, (x_k^1, f_k^1), \sigma_k, \ldots, (x_n^1, f_n^1), \sigma_n, (x_k^1, f_k^1)),$$
$$tr_2 = ((x_0, \emptyset), \sigma_0, (x_1^2, f_1^2), \ldots, (x_k^2, f_k^2), \sigma_k, \ldots, (x_n^2, f_n^2), \sigma_n, (x_k^2, f_k^2)).$$

2. $f_i^1 = f_j^1$ and $f_i^2 = f_j^2$ for any i and j in $\{k, k+1, \ldots, n\}$.

Next we provide a theorem which guarantees the correctness of Algorithm 1.

Theorem 1 G is diagnosable if and only if for every cycle cl in G_d,

$$cl = (x_1, \sigma_1, x_2, \ldots, x_n, \sigma_n, x_1), n \geq 1, \quad x_i = ((x_i^1, f^1), (x_i^2, f^2)), i = 1, 2, \ldots, n,$$

we have $f^1 = f^2$.

Proof: For the necessity, suppose G is diagnosable, but there exists a cycle cl in G_d, $cl = (x_k, \sigma_k, x_{k+1}, \ldots, x_n, \sigma_n, x_k)$, $n \geq k$, $x_i = ((x_i^1, f^1), (x_i^2, f^2))$, $i = k, k+1, \ldots, n$, such that $f^1 \neq f^2$. Since G_d is accessible, there exists a path tr in G_d ending with the cycle cl, i.e., $tr = (x_0^d, \sigma_0, x_1, \ldots, x_k, \sigma_k, \ldots, x_n, \sigma_n, x_k)$. Then from Lemma 2 we know that there exist two paths tr_1 and tr_2 in G_o with

$$tr_1 = ((x_0, \emptyset), \sigma_0, (x_1^1, f_1^1), \ldots, (x_k^1, f^1), \sigma_k, \ldots, (x_n^1, f^1), \sigma_n, (x_k^1, f^1)),$$
$$tr_2 = ((x_0, \emptyset), \sigma_0, (x_1^2, f_1^2), \ldots, (x_k^2, f^2), \sigma_k, \ldots, (x_n^2, f^2), \sigma_n, (x_k^2, f^2)).$$

Further from Lemma 1, we have $\exists u_1 v_1^*, u_2 v_2^* \in L(G)$ such that $M(u_1) = M(u_2) = \sigma_0 \ldots \sigma_{k-1}$, $M(v_1) = M(v_2) = \sigma_k \ldots \sigma_n$, and $\{\psi(\sigma) \mid \sigma \in u_i, \psi(\sigma) \neq \emptyset\} = \{\psi(\sigma) \mid \sigma \in u_i v_i, \psi(\sigma) \neq \emptyset\} = f^i$, $i = 1, 2$. Since $f^1 \neq f^2$, we suppose $F_k \in f^1 - f^2 \neq \emptyset$. Then $\exists s \in L(G)$ such that $\psi(s_f) = F_k$ and $u_1 = st$ for some $t \in \Sigma^*$. For any integer n_k, we can choose another integer ℓ such that $||tv_1^\ell|| > n_k$. Now we have $M(u_2 v_2^\ell) = M(stv_1^\ell)$ and $\{\psi(\sigma) \mid \sigma \in u_2 v_2, \psi(\sigma) \neq \emptyset\} = f^2$, which means that no failure event of type F_k is contained in $u_2 v_2^\ell$. So from the definition of diagnosability, G is not diagnosable. A contradiction to the hypothesis. So the necessity holds.

For the sufficiency, suppose for every cycle cl in G_d, $cl = (x_1, \sigma_1, x_2, \ldots, x_n, \sigma_n, x_1)$, $n \geq 1$, $x_i = ((x_i^1, f^1), (x_i^2, f^2))$, $i = 1, 2, \ldots, n$, we have $f^1 = f^2$. From the second clause of Lemma 2, we know that the hypothesis implies that $\forall x = ((x^1, f^1), (x^2, f^2)) \in X_d$, if $f^1 \neq f^2$ then x is not contained in a loop. It further implies that for any state sequence $(x_1, x_2, \cdots, x_k)$ in G_d with $x_i = (x_i^1, f_i^1), (x_i^2, f_i^2))$ for $1 \leq i \leq k$, if $f_i^1 \neq f_i^2$ for all $i \in \{1, 2, \cdots, k\}$, then the length of the state sequence is bounded by the number of states in G_d, i.e., $k \leq |X_d|$.

Now let s be a trace in $L(G)$ ending with a F_k-type failure event, i.e., $\psi(s_f) = F_k$, we claim that $\forall v = st \in L(G)$ with $||t|| > |X_d| \times (|X| - 1)$, $\forall w \in L(G)$ with $M(w) = M(v)$, there is a F_k-type failure event contained in w. From above, for any state $x \in X_d$ that can be reached from x_0^d by executing $M(s)$ in G_d, we have that for any state sequence starting from x in G_d, a state $y = ((y^1, f^1), (y^2, f^2)) \in X_d$ with $f^1 = f^2$ can be reached within $|X_d| - 1$ steps. This implies that $\forall v = st \in L(G)$ with $||M(t)|| > |X_d| - 1$, $\forall w \in L(G)$ with $M(w) = M(v)$, there is a F_k-type failure event contained in w. Further from the assumption that no unobservable cycle exists in G, each "observed event" in $M(t)$ can be preceded/followed by at most $|X| - 1$ unobserved events. It follows that for the trace t above, $||t|| \leq (||M(t)|| + 1) \times (|X| - 1)$, i.e., $||M(t)|| \geq \frac{||t||}{|X|-1} - 1$. So if $||t|| > |X_d| \times (|X| - 1)$, then $||M(t)|| \geq \frac{||t||}{|X|-1} - 1 > \frac{|X_d| \times (|X|-1)}{|X|-1} - 1 = |X_d| - 1$, establishing our claim. (Note that we have assumed implicitly that $|X| > 1$; otherwise if $|X| = 1$, then from the assumption of no unobservable loops, no transition labeled by a failure event exists, so that the system is trivially diagnosable.) It follows from Definition 1 that G is diagnosable. So the sufficiency also holds. ■

Remark 1 From Algorithm 1, we know that the number of states in G_o is at most $|X| \times 2^{|\mathcal{F}|}$, the number of transitions in G_o is at most $|X|^2 \times 2^{2|\mathcal{F}|} \times |\Sigma_o|$. Since $G_d = G_o || G_o$, the number of states in G_d is at most $|X|^2 \times 2^{2|\mathcal{F}|}$, and the number of transitions in G_d is at most $|X|^4 \times 2^{4|\mathcal{F}|} \times |\Sigma_o|$.

The complexity of performing step 1 of Algorithm 1, which construct G_o, is thus $O(|X|^2 \times 2^{2|\mathcal{F}|} \times |\Sigma_o|)$, whereas that of step 2 of Algorithm 1, which construct G_d, is thus $O(|X|^4 \times 2^{4|\mathcal{F}|} \times |\Sigma_o|)$. The complexity of performing step 3 of Algorithm 1, which detects the presence of a certain "offending" cycle in an appropriately pruned subgraph of G_d (see the last sentence of step 3 of Algorithm 1), is linear in the number of states and transitions of the subgraph, i.e., it is $O(|X|^4 \times 2^{4|\mathcal{F}|})$. Note that while detecting the presence of a "offending" cycle, the transition labels are irrelevant.

So the complexity of Algorithm 1 is $O(|X|^4 \times 2^{4|\mathcal{F}|} \times |\Sigma_o|)$ which is polynomial in the number of states in G and exponential in the number of failure types in G.

In [4], another necessary and sufficient condition was given for diagnosability. The condition was expressed as a property of a certain diagnoser of the system. So in order to check the diagnosability we needed to first construct the diagnoser, then check the property on the diagnoser. The complexity to construct the diagnoser as well as the complexity to check the property on the diagnoser is exponential in the number of states of the system and doubly exponential in the number of failure types of the system. In Algorithm 1, no diagnoser is needed for checking the diagnosability.

Remark 2 The complexity of testing diagnosability can be made polynomial in the number of fault types as well by noting that a system is diagnosable with respect to the fault types $\mathcal{F} = \{F_i, i = 1, 2, \cdots, m\}$ if and only if it is diagnosable with respect to the each individual fault type $F_i, i = 1, 2, \cdots, m$. In other words, one can apply Algorithm 1 m different times for testing diagnosability with respect the individual failure type sets $\{F_1\}, \cdots, \{F_m\}$. Since now each failure type set is a singleton, from Remark 1 it follows that the complexity of each such test is $O(|X|^4 \times 2^{4|1|} \times |\Sigma_o|) = O(|X|^4 \times |\Sigma_o|)$. So, the overall complexity of testing diagnosability is $O(|X|^4 \times |\Sigma_o| \times |\mathcal{F}|)$.

Example 1 Consider the system $G = (X, \Sigma, \delta, x_0)$:

- $X = \{x_0, x_1, x_2, x_3, x_4\}$
- $\Sigma = \{\sigma_1, \sigma_2, \sigma_3, \sigma_{uo}, \sigma_{f1}, \sigma_{f2}, \sigma_{f3}\}$
- $\delta = \{(x_0, \sigma_1, x_1), (x_1, \sigma_{f1}, x_2), (x_1, \sigma_{uo}, x_2), (x_2, \sigma_{f2}, x_3),$
  $(x_3, \sigma_2, x_3), (x_2, \sigma_{f1}, x_4), (x_4, \sigma_3, x_4)\}$

with the observable event set $\Sigma_o = \{\sigma_1, \sigma_2, \sigma_3\}$. The system is shown in Figure 1. Let $\mathcal{F} = \{F_1, F_2\}$ be the set of failure types and ψ be the failure assignment function with

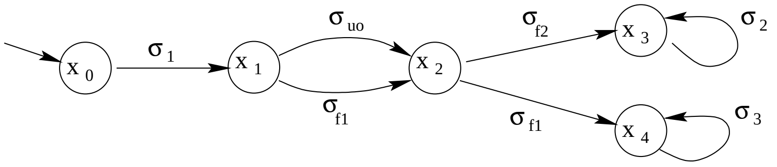

Figure 1: Diagram of the system G

$\psi(\sigma_{uo}) = \psi(\sigma_i) = \emptyset$, $i = 1, 2, 3$, $\psi(\sigma_{f1}) = F_1$, $\psi(\sigma_{f2}) = F_2$. From the first step in Algorithm 1, we can derive G_o from G, which is shown in Figure 2. The strict composition of G_o with itself,

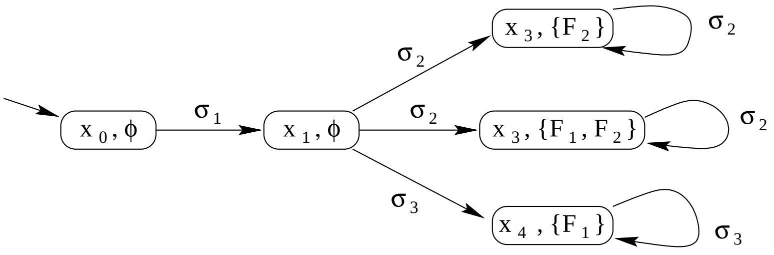

Figure 2: Diagram of G_o

$G_d = G_o||G_o$, is derived from the second step in Algorithm 1, which is shown in Figure 3. In

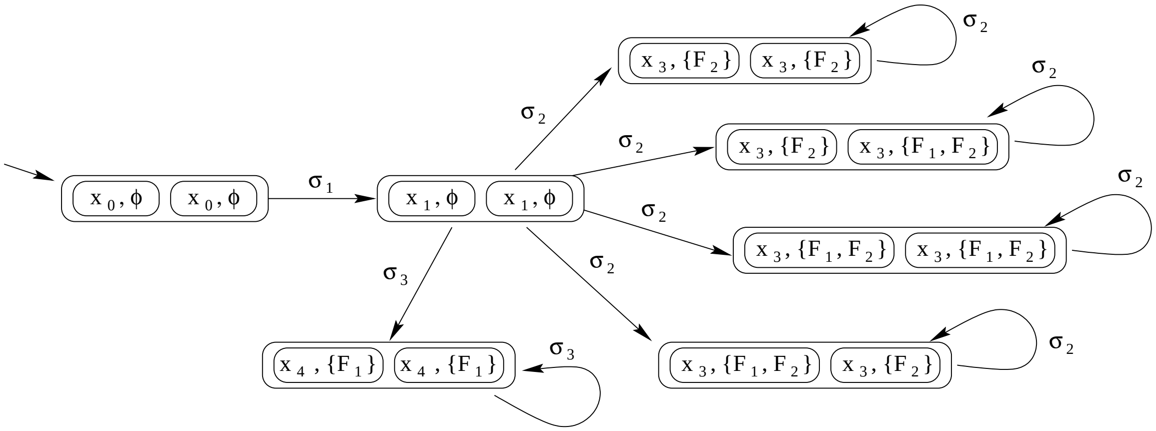
σ_2
$x_3, \{F_2\}$ $x_3, \{F_2\}$
σ_2
σ_2
σ_2
$x_3, \{F_2\}$ $x_3, \{F_1, F_2\}$
σ_2
x_0, ϕ x_0, ϕ
σ_1
x_1, ϕ x_1, ϕ
σ_2
$x_3, \{F_1, F_2\}$ $x_3, \{F_1, F_2\}$
σ_3
σ_2
σ_3
$x_4, \{F_1\}$ $x_4, \{F_1\}$
σ_2
$x_3, \{F_1, F_2\}$ $x_3, \{F_2\}$

Figure 3: Diagram of G_d

Figure 3, there is a self loop at the state $((x_3, \{F_2\}), (x_3, \{F_1, F_2\}))$. So from the last step in Algorithm 1 we know the system G is not diagnosable.

Now suppose we need not distinguish the failure type F_1 from the type F_2. Then by letting $F_2 = F_1$ in Figure 3 and deleting some redundant states, we can obtain the corresponding

G_d for the modified system. The resulting G_d is omitted here. In the modified G_d, there does not exist any cycle as stated in step 3 of Algorithm 1. So we know the modified system is diagnosable.

4 Conclusion

In this paper, an algorithm is provided for testing the diagnosability of discrete event systems. Compared to the existing testing method in [4], our algorithm does not require the construction of a diagnoser for the system. The complexity of our algorithm is of 4th order in the number of states of the system and linear in the number of failure types of the system, whereas the complexity of the testing method in [4] is exponential in the number of states of the system and doubly exponential in the number of failure types of the system.

References

[1] Y. L. Chen and G. Provan. Modeling and diagnosis of timed discrete event systems–a factory automation example. In *Proc. of the American Control Conference*, 31-36, Albuquerque, New Mexico, June 1997.

[2] L. Holloway and S. Chand. Time templates for discrete event fault monitoring in manufacturing systems. In *Proc. of 1994 American Control Conference*, 701-706, 1994.

[3] F. Lin. Diagnosability of discrete event systems and its applications. *J. Discrete Event Dynamic Systems: Theory and Applications*, 4(2):197-212, May 1994.

[4] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 40(9):1555-1575, Sept. 1995.

[5] S. H. Zad, R. H. Kwong, and W. M. Wonham. Fault diagnosis in timed discrete-event systems. In *Proc. 38th IEEE Conf. Decision Contr.*, 1756-1761, Phoenix, Arizona, Dec. 1999.

[6] G. Westerman, R. Kumar, C. Stroud, and J. R. Heath. Discrete event systems approach for delay fault analysis in digital circuits. In *Proceedings of 1998 American Control Conference*, Philadelphia, PA, 1998.