Observability of Discrete Event Dynamic Systems

CÜNEYT M. ÖZVEREN, MEMBER, IEEE, AND ALAN S. WILLSKY, FELLOW, IEEE

Abstract—**A finite state automaton is adopted as a model for Discrete Event Dynamic Systems (DEDS). Observations are assumed to be a subset of the event alphabet. Observability is defined as having perfect knowledge of the current state at points in time separated by bounded numbers of transitions. A polynomial test for observability is given. It is shown that an observer may be constructed and implemented in polynomial time and space. A bound on the cardinality of the observer state space is also presented. A notion of resiliency is defined for observers, and a test for resilient observability and a procedure for the construction of a resilient observer are presented.**

I. Introduction

DISCRETE Event Dynamic Systems (DEDS) have received considerable attention in the control literature recently. Many large scale dynamic systems seem to have a DEDS structure, at least at some level of description. Some examples are manufacturing systems [7], [17], communication systems (such as data networks and distributed systems) [1], and expert systems (such as CPU design or air-traffic management) [2], [3], [18].

The notion of the control of a DEDS was, to our knowledge, first explicitly introduced in the work of the authors and their colleagues [5], [8], [15], [14], [20]. In this work, it is assumed that certain events in the system can be enabled or disabled. The control of the system is achieved by choice of control inputs that enable or disable these events. The objective is to have a closed-loop system, so that the event trajectory in this system is always in a given set of desired strings of events. This approach is generally classified as a linguistic approach, since the objective is defined in terms of the language generated by the closed-loop system, i.e., the set of possible strings of events.

This work has prompted a considerable response by other researchers in the field, and one of the principal characteristics of this research has been the exploration of alternate formulations and paradigms that provide the opportunity for new and important developments building on the foundations of both computer science (for example, building on the concepts in [4]) and control. The work presented here is very much in that spirit with, perhaps, closer ties to more standard control concepts. In particular, in our work, we have had in mind the development of the elements needed for a regulator theory for DEDS. In another paper [12], we develop notions of stability and stabilizability for DEDS which might, more correctly, be thought of as properties of resiliency or error recovery. In this paper, we focus on the output side of the problem, namely, on the questions of observability and state reconstruction.

Partial observation problems have been the subject of several investigations in the literature. In particular, Cieslak *et al.* [1] and Lin and Wonham [6] formulate a supervisor control problem that can be thought of as a dynamic output compensation problem. Ramadge [13], on the other hand, explicitly addresses the observability problem. In particular, as in this paper, Ramadge addresses the problem of determining the current state of the system. In his framework, partial observations may be available concerning both the system state and events. In this paper, we assume what might be thought of as an intermittent observation model: no direct measurements of the state are made, and we only observe a specified subset of possible events, i.e., if an event outside this subset occurs, we will not observe it and indeed will not even know that an event has occurred. The more substantive difference between [13] and the present paper is in the notion of observability that is adopted. In particular, Ramadge requires exact reconstruction of the current state after each system event, while in our work, we allow state ambiguities to develop (as they must if some events are unobserved) but require that these be resolvable after a bounded interval of events. While this difference in formulations is quite fundamental, we will see that the concept of indistinguishability introduced by Ramadge plays an important role in our work as well.

In addition to characterizing observability and constructing observers, we also introduce a notion of stability that we feel is of some importance more generally in characterizing desirable behavior in a DEDS. In particular, we introduce the notion of *resiliency* for an observer, corresponding to its ability to recover from a finite burst of errors.

In the next section, we introduce the mathematical framework considered in this paper, and we address the problem of observability. In particular, we characterize observability and related notions of always-observability and observability with a delay. We provide polynomial tests for these notions and algorithms to construct appropriate observers. In Section III, we turn our attention to complexity issues. We show that an observer may have an exponential number of states. Since the observer itself can be implemented in polynomial time, complexity is only important for stabilization by output feedback. In Section IV, we characterize resilient observability, and construct a resilient observer. Finally, in Section V, we summarize our results and discuss several directions for further work.

II. Observability

A. Background and Preliminaries

The class of systems we consider are nondeterministic finite-state automata with intermittent event observations. The basic object of interest is the triple:

$$G=(X, \Sigma, \Gamma) \qquad (2.1)$$

where X is the finite set of states, with $n = |X|$, Σ is the finite set of possible events, and $\Gamma \subset \Sigma$ is the set of observable events. The dynamics of the system are characterized by two functions f and d:

$$x[k+1] \in f(x[k], \sigma[k+1]) \qquad (2.2)$$

$$\sigma[k+1] \in d(x[k]). \qquad (2.3)$$

Here, $x[k] \in X$ is the state after the kth event, and $\sigma[k+1] \in$

Manuscript received March 30, 1989; revised November 8, 1989. Paper recommended by Associate Editor, A. Haurie. This work was supported by the Air Force Office of Scientific Research under Grant AFOSR-88-0032, and by the Army Research Office under Grant DAAL03-86-K0171. This research was partially done at Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA), Rennes, France, and the second author was also supported by IRISA.

The authors are with the Laboratory for Information and Decision Systems, M.I.T., Cambridge, MA 02139.

IEEE Log Number 9035377.

0018-9286/90/0700-0797$01.00 © 1990 IEEE

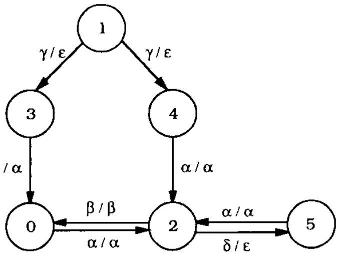


Fig. 1. A simple example.

Σ is the $(k + 1)$st event. The function $d:X \rightarrow 2^{\Sigma}$ is a set-valued function that specifies the set of possible events defined at each state (so that, in general, not all events are possible from each state), and the function $f:X \times \Sigma \rightarrow 2^{X}$ is also set-valued, so that the state following a particular event is not necessarily known with certainty. Note that f can be extended to act on strings over Σ by $f(x, \sigma_1 \cdots \sigma_n) = f(f \cdots f(x, \sigma_1), \cdots \sigma_{n-1}), \sigma_n)$. In calculating the complexity of algorithms that we present in this paper, we will assume that the number of transitions defined at each state, $|f(x, \Sigma)|$ for each $x \in X$, is small. It is otherwise straightforward to recompute the complexity of algorithms in order to account for $|f(x, \Sigma)|$. In the investigations of control of DEDS, one typically introduces control by allowing it to influence the set of possible events specified by d. We do not introduce it here as it is not needed for the present investigation.

Our model of the output process is quite simple: whenever an event in Γ occurs, we observe it; otherwise, we see nothing. Specifically, we define the output function $h:\Sigma \rightarrow \Gamma \cup \{\epsilon\}$, where ϵ is the "null transition," by

$$h(\sigma) = \begin{cases} \sigma & \text{if } \sigma \in \Gamma \\ \epsilon & \text{otherwise.} \end{cases} \tag{2.4}$$

Then, our output equation is

$$\gamma[k+1] = h(\sigma[k+1]). \tag{2.5}$$

Note that h can be thought of as a map from Σ^* to Γ^*, where Γ^* denotes the set of all strings of finite length with elements in Γ, including the empty string ϵ. In particular, $h(\sigma_1 \cdots \sigma_n) = h(\sigma_1) \cdots h(\sigma_n)$. The quadruple $A = (G, f, d, h)$ representing our system can also be visualized graphically as in Fig. 1. Here, circles denote states, and events are represented by arcs. The first symbol in each arc label denotes the event, while the symbol following "/" denotes the corresponding output. Thus, in this example, $X = \{0, 1, 2, 3, 4, 5\}$, $\Sigma = \{\alpha, \beta, \delta, \gamma\}$, and $\Gamma = \{\alpha, \beta\}$.

There are several basic notions that we will need in our investigation. The first is the notion of *liveliness*. Intuitively, a system is alive if it cannot reach a point at which no event is possible. That is, A is alive if $\forall x \in X$, $d(x) \neq \emptyset$. We will assume that this is the case. A second notion that we need is the composition of two automata, $A_i = (G_i, f_i, d_i, h_i)$ which share some common events. Specifically, let $S = \Sigma_1 \cap \Sigma_2$ and, for simplicity, assume that $\Gamma_1 \cap S = \Gamma_2 \cap S$ (i.e., any shared event observable in one system is also observable in the other). The dynamics of the composition are specified by allowing each automaton to operate as it would in isolation except that when a shared event occurs, it *must* occur in both systems. Mathematically, we denote the composition by $A_{12} = A_1 \| A_2 = (G_{12}, f_{12}, d_{12}, h_{12})$, where

$$G_{12} = (X_1 \times X_2, \Sigma_1 \cup \Sigma_2, \Gamma_1 \cup \Gamma_2) \tag{2.6}$$

$$f_{12}(x, \sigma) = f_1(x_1, \sigma) \times f_2(x_2, \sigma) \tag{2.7}$$

$$d_{12}(x) = (d_1(x_1) \cap \bar{S}) \cup (d_2(x_2) \cap \bar{S}) \cup (d_1(x_1) \cap d_2(x_2)) \tag{2.8}$$

$$h_{12}(\sigma) = \begin{cases} h_1(\sigma) & \text{if } \sigma \in \Gamma_1 \\ h_2(\sigma) & \text{if } \sigma \in \Gamma_2 \\ \epsilon & \text{otherwise.} \end{cases} \tag{2.9}$$

Here we have extended each f_i to all of Σ_i in the trivial way, namely, $f_i(x_i, \sigma) = x_i$ if $\sigma \notin \Sigma_i$. Note also that h_{12} given by (2.9) is well defined.

Two issues often studied in computer science in the context of such compositions are liveness (i.e., the absence of deadlocks) and fairness. Such a composition is *fair* if it is impossible for an infinite number of transitions to occur in one system alone without any transitions occurring in the other. In our present context, in which we will be composing systems and observers, liveness will not be an issue and fairness will be guaranteed by assumption on our DEDS.

Another property we would like the DEDS under investigation to have is that observations occur with some regularity. Specifically, since we are only observing events in Γ in our automaton A, we will not want it to be possible for our DEDS to generate arbitrarily long sequences of unobservable events, i.e., events in $\bar{\Gamma}$, the complement of Γ. A necessary condition for this is that if we remove the observable events, the resulting automaton $A|\bar{\Gamma} = (G, f, d \cap \bar{\Gamma}, h)$ must *not* be alive. However, we actually want more than this, namely, that every trajectory in $A|\bar{\Gamma}$ is killed in finite time by being forced into a state x for which $d(x) \cap \bar{\Gamma} = \emptyset$. This condition can be stated in terms of the notion of stability introduced in [12] which we will also use in the next section to characterize the notion of observability introduced in this paper: our notion of stability is a notion of recovery from any possible error in a finite number of transitions. Specifically, we assume that we have identified a set of "good" states (the set E in the following definition), and we define this notion of recovery in two stages as follows.

Definition 2.1: Let E be a specified subset of X. A state $x \in X$ is *E-prestable* if every trajectory starting from x passes through E in a finite number of transitions. The state $x \in X$ is *E-stable* if every state reachable from x is E-prestable. The DEDS is *E-stable* if every $x \in X$ is E-stable. □

Note that if x is E-stable, then every trajectory from x visits E infinitely often and indeed at intervals separated by at most n events [12]. Also, as shown in [12], a necessary and sufficient condition for E-stability of A is the absence of cycles that do not pass through E. Here, a cycle is a finite sequence of states $x_1, x_2, \cdots, x_k$, with $x_k = x_1$, so that there exists an event sequence s that allows the system to follow this sequence of states. We refer the reader to [12] for a more complete discussion of this subject and for an $O(n^2)$ test for E-stability of a DEDS.

It is not difficult to see that an equivalent condition to our DEDS being unable to generate arbitrarily long sequences of unobservable events is that if we remove the observable events, the resulting automaton $A|\bar{\Gamma} = (G, f, d \cap \bar{\Gamma}, h)$ must be D-stable, where D is the set of states that only have observable transitions defined, i.e., $D = \{x \in X | d(x) \cap \bar{\Gamma} = \emptyset\}$.[1] This is not difficult to check and will be assumed.

Finally, let us introduce some notations that we will find useful.

- Let $x \rightarrow^{s} y$ denote the statement that state y is reached from x via the occurrence of event sequence s. Also, let $x \rightarrow^{*} y$ denote that x reaches y in any number of transitions, including none. We also define the reach of x in A as

$$R(A, x) = \{y \in X | x \rightarrow^{*} y\}. \tag{2.10}$$

Finally, given $X' \subset X$, we let $R(A, X') = \cup_{x \in X'} R(A, x)$.

- Let

$$Y_0 = \{x \in X | \nexists y \in X, \gamma \in \Sigma, \quad \text{such that } x \in f(y, \gamma)\} \tag{2.11}$$

[1] In [12], we have defined stability for live systems. Although, $A|\bar{\Gamma}$ is not alive, its trajectories can only die in D, and thus, our results on stability will carry to this case.

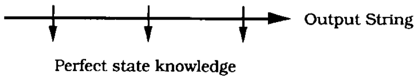


Fig. 2. Notion of observability. The state is known perfectly only at the indicated instants. Ambiguity may develop between these, but is resolved in a bounded number of steps.

$$Y_1 = \{x \in X \mid \exists y \in X, \gamma \in \Gamma, \quad \text{such that } x \in f(y, \gamma)\} \tag{2.12}$$

$$Y = Y_0 \cup Y_1. \tag{2.13}$$

Thus, Y is the set of states x such that either there exists an observable transition defined from some state y to x (as captured in Y_1) or x has no transitions defined to it (as captured in Y_0). Let $q = |Y|$.

- Let $L(A, x)$ denote the language generated by A, from the state $x \in X$, i.e., $L(A, x)$ is the set of all possible event trajectories of finite length that can be generated if the system is started from the state x. Given $s \in L(A, x)$ for some x, let s_f denote the final event in s and let

$$L_f(A, x) = \{s \in L(A, x) \quad \text{and } s_f \in \Gamma\} \tag{2.14}$$

be the set of strings in $L(A, x)$ that have an observable event as its final event. Similarly, $L_1(A, x)$ denotes the set of strings of $L_f(A, x)$ that contain one observable event, and given some $\gamma \in \Gamma$, $L_\gamma(A, x)$ denotes the set of strings of $L_1(A, x)$ that have γ as the observable event.

- Given $s \in L(A, x)$ such that $s = pr$, p is termed a *prefix* of s and we use s/p to denote the corresponding suffix r, i.e., the remaining part of s after p is taken out.

B. State Observability

As mentioned in the Introduction, and as illustrated in Fig. 2, we term a system observable if we can use the observation sequence $\gamma[k]$ to determine the current state exactly at intermittent (but not necessarily fixed) points in time separated by a bounded number of events. The precise definition is as follows.

Definition 2.2: A is observable if there exists some integer n_o such that $\forall x \in X$, $\forall s \in L(A, x)$ such that $|s| \geq n_o$, there exists a prefix of s, $p \in L_f(A, x)$, such that $|s/p| \leq n_o$, $f(x, p)$ is single valued, and $\forall y \in X$, $t \in L_f(A, y)$: $h(t) = h(p) \Rightarrow f(y, t) = f(x, p)$. □

This definition states the following. Take any sufficiently long string s that can be generated from any initial state x. For an observable system, we can then find a prefix p of s such that p takes x to a unique state, and the length of the remaining suffix is bounded by some integer n_o. Also, for any other string t, from some initial state y, such that t has the same output string as p, we require that t takes y to the same, unique state to which p takes x.

Let us note some very important implications of this definition. First, the string p need not be of length one. Thus, while from the definition we will know the state after p is observed, we may not know it at earlier points. Furthermore, since $p \in L_f(A, x)$, when we do know the state, that state will necessarily lie in Y. That is, since we only observe events in Γ, the only possible times at which we might know the state is at points at which events in Γ occur, i.e., points at which $x[k] \in Y$. Observability is in fact weaker, since in particular, in an observable system, we need not know the state every time it enters Y or even every time it visits a particular state in Y: all we can be sure of is that we will know the state at points separated by n or fewer events, and that when we know the state, it will be in Y.

This suggests a straightforward design of an observer that produces "estimates" of the state of the system after each observation $\gamma[k] \in \Gamma$. Each such estimate is a subset of Y corresponding to the set of possible states into which A transitioned when the last observable event occurred. The state space for the observer is a subset Z of 2^Y, and the events and observable events are both Γ. What this observer must do is the following. Suppose that the present observer estimate is $\hat{x}[k] \in 2^Y$ and that the next output is $\gamma[k+1]$. The observer must then account for the possible occurrence of one or more unobservable events prior to $\gamma[k+1]$ and then the occurrence of $\gamma[k+1]$:

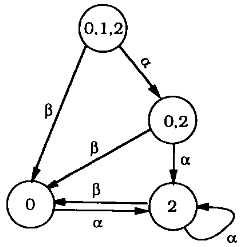


Fig. 3. Observer for the system in Fig. 1.

$$\hat{x}[k+1] = w(\hat{x}[k], \gamma[k+1]) \triangleq \cup_{x \in R(A|\bar{\Gamma}, \hat{x}[k])} f(x, \gamma[k+1]) \tag{2.15}$$

$$\gamma[k+1] \in v(\hat{x}[k]) \triangleq h(\cup_{x \in R(A|\bar{\Gamma}, \hat{x}[k])} d(x)). \tag{2.16}$$

The set Z is then in the reach of $\{Y\}$ using these dynamics. Note that once the first observable transition occurs, the state $\hat{x}[k]$ is in fact a subset of Y_1. However, before this point, we have no knowledge of the state. Thus, the choice of initial state is an issue that must be resolved. Note first that taking Y_1 as the initial state does not work, in general, as there may be states in Y_1 which can be reached by observable transitions only from *transient* states. Thus, we must augment Y_1 in order for the dynamics (2.15) and (2.16) to determine the correct state estimate sequence. It is easily shown that Y, as we have defined it, is the smallest subset of X that contains Y_1 and which, when used as the initial state of the observer, allows (2.15) and (2.16) to produce the correct estimate sequence.

Our observer then is the DEDS $O = (F, w, v, i)$, where $F = (Z, \Gamma, \Gamma)$ and i is the identity output function. The observer for the example in Fig. 1 is illustrated in Fig. 3. Note that the set of allowable events $v(\hat{x}[k])$ defined in (2.16) characterizes all possibilities for the next observable event given the set of possible states $\hat{x}[k]$. In general, $v(\hat{x}[k]) \not\equiv \Gamma$ for all $\hat{x}[k]$, i.e., not *all* sequences in Γ^* can actually occur in our system A. If such an unallowable sequence is observed, an error has obviously occurred. In Section IV, we will deal with this in order to define the composition of A and O in our treatment of resiliency. Observability, however, can be considered by examining O by itself. Specifically, let $E = (\cup_{x \in Y}\{x\}) \cap Z$ be the singleton states of O. The following result ties observability with stability.

Proposition 2.3: A is observable iff E is nonempty and O is E-stable.

Proof: Note first that E must necessarily be nonempty for the system to be observable. Thus, we assume that this is true, and focus then on necessity and sufficiency of E-stability. To prove necessity, assume the contrary. Then [12] there exists a cycle $\hat{x}_1 \hat{x}_2 \cdots \hat{x}_k = \hat{x}_1$ in O for which $|\hat{x}_i| > 1$ for all i. Let s denote the output sequence producing this cycle. Then, an arbitrarily long repetition of this sequence is a feasible output sequence for A. If this occurs, we will never know the current state exactly.

Now suppose that O is E-stable, and let $n_o = n|Z|$. Thanks to E-stability, the trajectories from all observer states go through E in at most $|Z|$ observations. Since we also assumed that A cannot generate arbitrarily long sequences of unobservable events, for any output that the system can generate, the observer goes through singleton states at intervals of at most n_o events. Let us now show that Definition 2.2 is satisfied. Given $x \in X$ and $s \in L(A, x)$ such that $|s| \geq n_o$, let $p \in L_f(A, x)$ be a prefix of s such that $|s/p| \leq n_o$ and $w(\{Y\}, h(p)) \triangleq \hat{x} \in E$. The existence of such a p

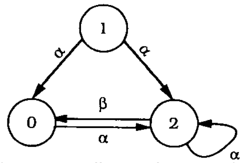


Fig. 4. A' corresponding to the example in Fig. 1.

is guaranteed thanks to E-stability. Furthermore, since $\hat{x}$ is a singleton, $f(x, p)$ is clearly single valued. Finally, to show that

$$\forall y \in X, t \in L_f(A, y) : h(t) = h(p) \Rightarrow f(y, t) = f(x, p),$$

let us assume the contrary, i.e., let us assume that there exists some $y \in X$ and $t \in L_f(A, y)$ such that $h(t) = h(p)$ and $f(y, t) \neq f(x, p)$. However, this implies that $\hat{x}$ cannot be a singleton, and we achieve a contradiction. Therefore, Definition 2.2 is satisfied and A is observable. □

Later in this section, we show that a generally tight upper bound on the interval between observer visits to singleton states is nq^2 in the worst case, and [9] illustrates a class of systems for which this bound is in fact tight. Note that the observer DEDS in Fig. 3 is stable with respect to $\{0, 2\}$ so that the system in Fig. 1 is observable.

It is interesting to contrast our notion of observability with that used in [13]. In particular, in [13] it is required that the state is known at all times. Therefore, it must be that $E = X$ and that once the observer enters E, it is trapped there forever. In contrast, we may have E substantially smaller than X and, furthermore, we allow the observer state to leave E, as long as it returns in the future.

Let us also first make a few comments about computational complexity. Note that the cardinality of Z, the observer state space, is bounded by 2^q. Thus, using the stability test in [12], we immediately have an $O(2^{2q})$ test for observability. In Section III, we will provide tighter bounds on the size of Z. Independently of this, however, we can devise an observability test that is polynomial in q. In particular, the reason for the apparent complexity of the test for observability is the size of the observer state space. An important point to note is that the observer is a deterministic automaton, i.e., it tells us *exactly* the set of possible current states given the observed output. To test for observability, however, all we really want to know is if there are recurring points in time at which all ambiguity in the current state vanishes. Fortunately, it is possible to construct a *nondeterministic* automaton that captures this with a dramatically smaller state space. Specifically, given A, construct A', a nondeterministic automaton with state space Y and event set Γ such that A' generates the same output language as A (see Fig. 4 for A' corresponding to the example in Fig. 1).

Let $P = Y \times Y$, and construct an automaton O_P with state space P and event set Γ such that

$$f_{O_P}(p, \gamma) = (f'(x, \gamma) \cup f'(y, \gamma)) \times (f'(x, \gamma) \cup f'(y, \gamma)) \tag{2.17}$$

$$d_{O_P}(p) = d'(x) \cup d'(y) \tag{2.18}$$

where f' is the transition map of A', $p = (x, y) \in P$, $\gamma \in \Gamma$, and we define $f'(x, \gamma)$ as $\emptyset$ if $\gamma \notin d'(x)$. Note that since it is nondeterministic, O_P is certainly not an observer for A. However, if its state ever evolves deterministically to a state of the form (x, x), the automaton A must be in state x. Thus, we have the following.

Proposition 2.4: A is observable iff O_P is E_P-stable where $E_P = \{(x, x) | x \in Y\}$.

Proof: Straightforward by assuming contrary in each direction. □

Since $|P| = q^2$, this gives us a test for observability that has complexity $O(q^4)$. This also leads to an upper bound on the maximum number of transitions it takes to reach a singleton state n_o (see Definition 2.2).

Corollary 2.5: If A is observable, then $n_o < nq^2$.

Proof: If A is observable, then all trajectories from an observer state reach a singleton state in at most q^2 transitions, since otherwise O_P is not E_P-stable. In addition, between each observable transition, there can be at most n unobservable transitions. Therefore, an upper bound for n_o is nq^2. □

C. Persistent States and Always-Observability

In this subsection, we address the problem of finding a set of always-observable states, in the sense that, except perhaps for a finite number of transitions in the beginning, the observer has perfect knowledge of the current state *every time* the system goes through always-observable states. We characterize this notion as follows.

Definition 2.6: A state $x \in X$ is *always-observable* iff there exists an integer n_a such that for all $y \in X$ and $s \in L(A, y)$ such that $x \in f(y, s)$ and $|s| \geq n_a$, $w(\{Y\}, h(s)) = \{x\}$. □

Note that an always-observable state has to be a singleton state in the observer. Furthermore, it should not be an element of any other persistent state of the observer which is not a singleton, where a persistent state is one that may be visited after an arbitrarily long string of events. States that are on a cycle are certainly persistent. The following definition also characterizes as persistent those states that are in between cycles, since these states, although they may be visited at most once, may have this visit occur after an arbitrarily long sequence of transitions. For this reason, they must also be accounted for in characterizing always-observability.

Definition 2.7: A state $x \in X$ is a *persistent state* if there exists some $y \in X$, $s \in L(A, y)$, $|s| \geq n$, such that $x \in f(y, s)$. A subset Q of X is termed a *persistent set* if all $x \in Q$ are persistent states. □

Clearly, the class of persistent sets are closed under unions and intersections. Thus, a maximal persistent set exists, and let X_R denote this set. In order to compute X_R, we compute $\overline{X_R}$ which, by the following result, is the maximal set of states stable (in fact, just prestable [12]) with respect to the dead states in A^{-1}, where A^{-1} denotes A with the transitions reversed, i.e., $A^{-1} = (G, f^{-1}, d^{-1})$ where

$$f^{-1}(x, \sigma) = \{y \in X | x \in f(y, \sigma)\} \tag{2.19}$$

$$d^{-1}(x) = \{\sigma \in \Sigma | \exists y \in X \quad \text{such that } x \in f(y, \sigma)\} \tag{2.20}$$

and the dead states in A^{-1}, D_i, are those states x such that $d^{-1}(x) = \emptyset$.

Proof:

($\subset$) Straightforward since all trajectories from $\overline{X_R}$ in A^{-1} are killed in a finite number of transitions.

($\supset$) Suppose x is D_i-stable, then all trajectories from x in A^{-1} are killed in a finite number of transitions. Therefore, $x \in \overline{X_R}$. □

The following proposition provides a mathematical characterization of always-observability.

Proposition 2.9: A persistent state $x \in X$ is an always-observable state iff

- x only has observable transitions defined to it, i.e., $d^{-1}(x) \subset \Gamma$, and
- for all $y \in X$, $s \in L_f(A, y)$ such that $|s| \geq nq^2$ and $x \in f(y, s)$, any string with the same output as s only goes to x, i.e., for all $z \in X$, $t \in L_f(A, z)$ such that $h(t) = h(s)$, $f(z, t) = x$. □

A subset Q of X is termed an *always-observable set* if all $x \in Q$ are always-observable states. A system A is termed *a-observable* if all trajectories in A visit always-observable states infinitely often. Note that this notion of a-observability is stronger than our notion of observability, but still weaker than the usual system-theoretic notion of observability which corresponds to requiring *all* persistent states to be always-observable.

Clearly, the class of always-observable sets are closed under unions and intersections. Thus, a maximal always-observable set X_A exists. As explained above, an always-observable state x should only have observable transitions defined to it, and the only persistent state of the observer that x is in should be the singleton state $\{x\}$.

Corollary 2.10: A persistent state x is always-observable iff $d^{-1}(x) \subset \Gamma$ and if $\hat{x}$ is a persistent observer state and $x \in \hat{x}$ then $\hat{x}$ is the singleton state $\{x\}$.

Proof:

($\rightarrow$) The proof for the first statement is obvious. To prove the second statement just assume the contrary.

($\leftarrow$) Straightforward. □

As we did before, we can use O_P to check if a state is always observable.

Proposition 2.11: A persistent state x is always-observable iff $d^{-1}(x) \subset \Gamma$ and if (x, y) for some y is a persistent state of O_P, then $y = x$.

Proof: Straightforward by assuming the contrary in each direction. □

Thus, X_A can simply be computed by performing this $O(q^4)$ test for each persistent state x such that $d^{-1}(x) \subset \Gamma$. Then, a test for a-observability is just a test for X_A-stability.

Proposition 2.12: A system A is a-observable iff it is X_A-stable.

Proof: Straightforward. □

D. Indistinguishability

Ramadge, in [13], introduces a notion of indistinguishability which he refers to as "possible indistinguishability." This turns out to be an extremely useful notion in our context as well. In this section, we reformulate his definition, present an algorithm for it in our framework, and use it, in the next subsection, to study observability with delay, and in Section III to analyze the complexity of the observer O.

A pair of states (x, y) is termed to be an indistinguishable pair if they share an infinite length output sequence. Since the observer uses the states in Y, for notational simplicity, we will define indistinguishability for states in Y.

Definition 2.13: Given $x \in X$, let $L_\infty(A, x)$ denote the set of infinite length event trajectories generated from x, and $h(L_\infty(A, x))$ the corresponding set of output trajectories. The pair $(x, y) \in Y \times Y$ is an *indistinguishable pair* if $h(L_\infty(A, x)) \cap h(L_\infty(A, y)) \neq \emptyset$, i.e., if there is an infinite length output sequence that could have been generated starting from either x or y. □

As an example, note that in Fig. 1, (0, 2) is an indistinguishable pair since an infinite string of α's is a possible output sequence from either state. Since we have seen that this system is observable, we now see that the absence of indistinguishable pairs is *not* required for observability.[2]

The following lemma establishes a recursion for indistinguishable pairs.

Lemma 2.14: (x, y) is an indistinguishable pair iff there exists $s \in L_1(A, x)$, and $t \in L_1(A, y)$ such that $h(s) = h(t)$, and there exists an indistinguishable pair $(z, w) \in f(x, s) \times f(y, t)$.

Proof:

($\rightarrow$) Assume contrary, then, for all $(z, w) \in f(x, s) \times f(y, t)$ all output sequences differ in a finite number of transitions. Therefore, (x, y) cannot be indistinguishable and we establish a contradiction.

($\leftarrow$) Straightforward. □

A subset I_P of $Y \times Y$ is called an indistinguishable pair set if every element (x, y) of I_P is an indistinguishable pair. Obviously, indistinguishable pair sets are closed under arbitrary unions and intersections. Thanks to the preceding lemma, we have the following for the computation of the maximal indistinguishable pair set.

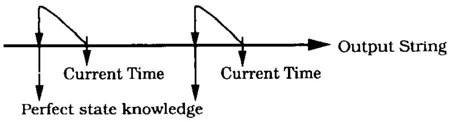


Fig. 5. Observability with a delay. The state, a finite number of transitions into the past, is known perfectly at intermittent (but not necessarily fixed) points in time.

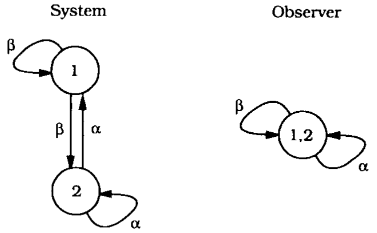


Fig. 6. Example for WD observability.

Proposition 2.15: The following algorithm computes the maximal set of indistinguishable pairs, I_M, and it has complexity $O(q^4)$.

Algorithm: Let $I_0 = Y \times Y$, and iterate

$$I_{k+1} = \{(x, y) \in I_k | f_{O_P}((x, y), \gamma) \cap I_k \neq \emptyset \quad \text{for some } \gamma\}.$$

Terminate when $I_{k+1} = I_k$. Then $I_M = I_k$.

Proof: The correctness of the algorithm is easily verified by using the definition of the automaton O_P and Lemma 2.14. To obtain a bound on computational complexity, note that I_0 has q^2 elements, and that the sequence of sets I_k is strictly decreasing up to some step at which the algorithm terminates. Thus, this algorithm terminates in at most q^2 steps. Since also at most q^2 states are visited at each step, the complexity of this algorithm is $O(q^4)$. □

E. Observability with a Delay

For observability with a delay, we require that we have perfect knowledge of the state some finite number of transitions into the past (as opposed to the current state) at intermittent (but not necessarily fixed) points in time (see Fig. 5).[3] For example, in Fig. 6, where all events are assumed to be observable, we have a system which is not observable. When α or β occurs, we do not have perfect knowledge of the current state, but if α (respectively, β) occurs, we know that the previous state is state 2 (respectively, state 1). Our formulation of this weak notion of observability is based on Definition 2.2, in which the prefix p of s characterized the point at which the *current* state is known perfectly. In the following definition, we use a prefix p_1 of s and a prefix p_2 of p_1, where $h(p_1)$ characterizes the information required to have perfect knowledge of the state at the *time in the past* just after the occurrence of p_2. For example, in Fig. 6, for a string $s = \alpha\beta\alpha\alpha$, $p_1 = s$ and $p_2 = \alpha\beta\alpha$. Perfect knowledge of the state is ensured by the third item below which (similar to Definition 2.2) states that for all strings t_1 which produce the same output as p_1, the state after t_2 is the same as the state after p_2 where t_2 is the prefix of t_1 that produces the same output as p_2.

Definition 2.16: A is *observable with a delay* (WD observable) if $\forall x \in X$, $s \in L(A, x)$ such that $|s| \geq nq^2$, there exist

[2] In general, if there are indistinguishable states, we will not always be able to determine which of these states we were in at some point in the *past*, but this does *not* rule out the possibility that we may occasionally know the *current* state.

[3] This is a concept which is of use in studying other aspects of DEDS such as invertibility [11]. In addition, delay in the knowledge of the state may not be of concern in the hierarchical study of DEDS where we represent strings of lower level events by a single event at the higher level [10].

prefixes $p_1 \in L_f(A, x)$ of s and $p_2 \in L_f(A, x)$ of p_1 such that

- $|x/p_2| \leq nq^2$,
- $f(x, p_2)$ is single valued,
- $\forall y \in X$ and $t_1 \in L_f(A, y): h(t_1) = h(p_1) \Rightarrow f(y, t_2) = f(x, p_2)$ where t_2 is the prefix of t_1 such that $h(t_2) = h(p_2)$. □

A test for WD observability can be constructed based on the following. If at any time the observer estimate $\hat{x}$ is such that all pairs in $\hat{x}$ are distinguishable, then by using future outputs we can distinguish between the states in $\hat{x}$ in a finite number of transitions. For example, in Fig. 6, since (1, 2) is *not* an indistinguishable pair, in a finite number of transitions, just one transition in this case, we can distinguish between 1 and 2. In general, a necessary and sufficient condition for WD observability is that the observer is stable with respect to the states that only include distinguishable pairs.

Proposition 2.17: A is WD observable iff O is E_w-stable where

$$E_W = \{\hat{x} \in Z \mid \text{ there exists no } x, y \in \hat{x}, x \neq y \text{ such that } (x, y) \in I_M\}.$$

Proof:

($\rightarrow$) Assume contrary, then there exists a cycle $\hat{x}_1 \cdots \hat{x}_k \hat{x}_1$ in O such that $\hat{x}_i \supset \{x_i, y_i\}$ where $x_i \neq y_i$ and (x_i, y_i) is an indistinguishable pair for all i. Let w be a string such that $x_1 \in f(x_1, w)$ and the event sequence $h(w)$ drives O precisely through the cycle $\hat{x}_1, \cdots, \hat{x}_k, \hat{x}_1$. Referring to Definition 2.16, let $x = x_1$, $s = w^l$ for some large enough l such that $|s| \geq nq^2$. Also pick $y = y_1$. For any prefix $p_1 \in L_f(A, x)$ of s, there exists some $t_1 \in L_f(A, y)$ such that $h(t_1) = h(p_1)$. On the other hand, for all prefixes p_2 of p_1 and corresponding prefix t_2 of t_1 such that $h(t_2) = h(p_2)$, we have that $x_i \in f(x, p_2)$ and $y_i \in f(y, t_2)$ for some i. Since $x_i \neq y_i$ for all i, $f(x, p_2) \neq f(y, t_2)$, and we establish a contradiction with the third item in Definition 2.16, and A cannot be WD observable. Therefore, O must be E_W-stable.

($\leftarrow$) Straightforward. □

As we did with observability, we use the automaton O_P to construct a polynomial test for WD observability. It is necessary and sufficient to check stability of O_P with respect to the distinguishable pairs.

Proposition 2.18: A is WD observable iff O_P is E_{DP}-stable where $E_{DP} = \{(x, y) \notin I_M\}$.

Proof: Straightforward by assuming the contrary in each direction. □

Fig. 6 is a very simple example that illustrates this result.

III. Observer Implementation and Complexity

Recall that the next state of the observer is expressed as a function of the current state and the next event as follows (2.15):

$$\hat{x}[k+1] = \cup_{x \in R(A|\bar{\Gamma}, \hat{x}[k])} f(x, \gamma[k+1]) \tag{3.1}$$

which can also be expressed as

$$\hat{x}[k+1] = \cup_{x \in \hat{x}[k]} \hat{r}(x, \gamma[k+1]) \tag{3.2}$$

where

$$\hat{r}(x, \gamma) = f(R(A|\bar{\Gamma}, x), \gamma). \tag{3.3}$$

Clearly, $\hat{r}$ can be computed beforehand for all $x \in Y$ and $\gamma \in \Gamma$. This computation has $O(|\Gamma| q^2)$ complexity, and the result occupies $O(|\Gamma| q^2)$ memory. Thus, computation of the next state of the observer simply becomes taking the union of $\hat{r}(x, \gamma[k+1])$ for all $x \in \hat{x}$, which has $O(q^2)$ complexity. Since, also, observability can be tested in polynomial time, computational complexity associated with the observability problem by itself is polynomial.

While testing observability and the implementation of the observer do not require the complete enumeration of the observer state space, this enumeration is needed for other design and analysis problems. This is the case, for example, in the study of stabilization by output feedback which we will address in a subsequent paper. Thus, it is of interest to characterize the cardinality of the observer. Unfortunately, even if A is observable (or, for the same matter, a-observable), the observer may have an exponential number of states. As an example, consider the class of systems, shown in Fig. 7, which is a slightly modified version of Fig. 1 in [19].

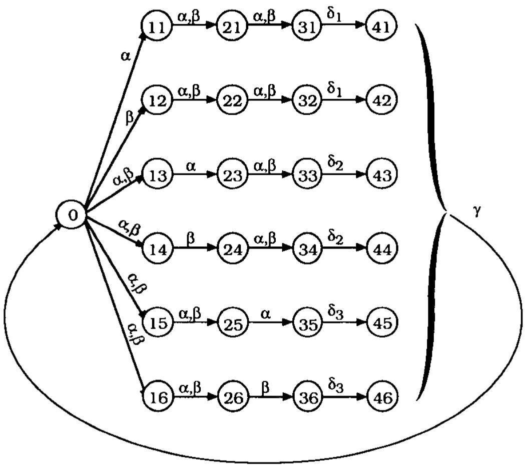


Fig. 7. Example of exponential observer state space.

We index this class by an integer i. In the system corresponding to $i = 3$, illustrated in Fig. 7, all events are observable. The set of events for this class consists of α, β, γ, and δ_1 through δ_i. There are $2i(i + 1) + 1$ states, and one of them is state 0, whereas the rest is indexed by pairs of integers (j, l) for j ranging from 1 to $i + 1$ and l ranging from 1 to $2i$. It is not difficult to check that this system is observable and that 0 is an always-observable state. One can also show that the number of states in the observer is $O(2^i)$. To see why, suppose that the system is in state 0. If α (respectively, β) occurs, then the next state is in the set $\{11, 13, \cdots, 16\}$ (respectively, $\{12, \cdots, 16\}$). With the next event, the ambiguity in the current state is reduced to four states, then three states, etc. Furthermore, due to the particular way the transitions α and β are defined, the estimates corresponding to *each* sequence consisting of events α and β are different. It is this fact that leads to the exponential growth in the observer state space.

While the observer state space is exponential for the preceding example, there are many cases in which the cardinality of the state space is much smaller. Thus, it is of interest to characterize structure and characteristics of DEDS that may lead to significantly smaller observer state spaces. In the remainder of this section, we develop a bound on the size of the observer state space which, for certain DEDS, yields a much smaller number than 2^n. First of all, we restrict ourselves to put a bound on Z_R, the *persistent* part of the observer state space Z. For any problem such as stabilization, focusing on long-term behavior such as stability, it is only Z_R that is of concern (for example, in output feedback design we can simply let the system evolve without active control during the start-up period—until O enters Z_R—and at that point we can begin to apply feedback).

We begin our analysis by noting that two states x and y are elements of the *same* persistent observer state iff the pair (x, y) is indistinguishable in A^{-1}. For example, in Fig. 7, states 32 and 35 are indistinguishable if we reverse all the transitions in this automaton (since these two states then share the string, for example, $\alpha\alpha\beta(\gamma\delta_1\beta\beta\alpha)^*$). Therefore, the observer estimate after observing $(\alpha\beta\beta\delta_1\gamma)^*\beta\alpha\alpha$ is the set $\{32, 33, 35\}$ which includes the states 32 and 35. We use I_M^{-1} to denote the maximal set of indistinguishable pairs in A^{-1}, and this set will play a central role in the computation of our bound.

Let Y_R denote the persistent part of Y in our original automaton A (i.e., these are elements of Y that may be visited after arbitrarily long sequences of events). For any subset $S \subset Y_R$, we let $\eta(S)$ denote the number of persistent observer states which include different subsets of S:

$$\eta(S)=|\{Q \subset S \mid S \cap \hat{x}=Q \quad \text{for some } \hat{x} \in Z_R\}|. \tag{3.4}$$

Then, clearly, $|Z_R| = \eta(Y_R)$. To compute a bound, we first find a collection of disjoint subsets of Y_R such that each persistent observer state is a subset of exactly one element of this collection. First of all, we term a collection $\mathcal{B} = \{B_1, \cdots, B_k\}$ of disjoint subsets B_i of Y_R a *Y_R-partition* if $\cup_i B_i = Y_R$. A Y_R-partition $\mathcal{B}$ is termed a *Y_R-distinguishability-partition* if each pair indistinguishable in the inverse automaton is in some element of this partition, i.e., for all $(x, y) \in I_M^{-1}$, $\{x, y\} \subset B_i \in \mathcal{B}$. Since all pairs in an observer state are indistinguishable in the inverse automaton, they all must be in the same element of $\mathcal{B}$. For calculating a tight bound, we need to have the elements of $\mathcal{B}$ as small as possible. Thus, a Y_R-distinguishability-partition $\mathcal{B}$ is termed *fine* if for each $B_i \in \mathcal{B}$, the only B_i-distinguishability-partition is B_i itself. Clearly, there is only one Y_R-distinguishability-partition that is also fine, and we denote this partition by $\mathcal{B}^{\mathfrak{F}}$. Note that $\mathcal{B}^{\mathfrak{F}}$ is the quotient of Y_R by the transitive closure of indistinguishability in the inverse automaton, and there are well-known polynomial algorithms for computing $\mathcal{B}^{\mathfrak{F}}$ (see, for example, [16]). For Fig. 7, $\mathcal{B}^{\mathfrak{F}}$ consists of the sets $\{0\}$, $\{11, \cdots, 16\}$, $\{21, \cdots, 26\}$, $\{31, \cdots, 36\}$, $\{41\}, \cdots, \{46\}$. We then have the following result.

Proposition 3.1: For all $\hat{x} \in Z_R$, $\hat{x} \subset B_i \in \mathcal{B}^{\mathfrak{F}}$ for some i.

Proof: Straightforward. □

The following result immediately follows from the above proposition.

Corollary 3.2: Given $S \subset Y_R$, and $\mathcal{B}^{\mathfrak{F}} = \{B_i\}$, $\eta(S) = \Sigma_i \eta(B_i \cap S)$. Therefore,

$$|Z_R| = \Sigma_i \eta(B_i). \qquad \square$$

Corollary 3.3: We have the following first bound on the cardinality of the persistent part of the observer state space:

$$|Z_R| \leq \Sigma_i (2^{|B_i|}-1). \qquad \square$$

The "minus 1" in this equation corresponds to the fact that we can omit the empty set.

While this bound is exponential, it may be much tighter than $2^{|Y|} - 1$ if the partition $\mathcal{B}^{\mathfrak{F}}$ is quite fine. Furthermore, if B_i is large, in many cases $\eta(B_i)$ will be much smaller than $2^{|B_i|} - 1$. Now, we proceed with showing that by exploiting the structure of the system, we may compute a possibly tighter bound for Z_R, and we use Corollary 3.2 for this. For any $S \subset Y_R$, let $\phi(S, \alpha)$ be the set of states that can reach a state in S with a string that has α as its last and *only* observable event, i.e.,

$$\phi(S, \alpha)=R(A^{-1}|\bar{\Gamma}, f^{-1}(S, \alpha)). \tag{3.5}$$

Thus, given α, there are $\eta(\phi(S, \alpha))$ observer states that may make a transition, with α, to an observer state which is a subset of S. Thus, if we add these for all such events α, we get an upper bound for $\eta(S)$:

$$\eta(S) \leq \sum_{\alpha \in \Gamma} \eta(\phi(S, \alpha)). \tag{3.6}$$

But, by using Corollary 3.2, we can decompose $\phi(S, \alpha)$ using the partition $\mathcal{B}^{\mathfrak{F}}$ and compute η for each part. We thus have the following result, where we assume that $S \subset B_i \in \mathcal{B}^{\mathfrak{F}}$, since otherwise we can decompose S itself using the following partition.

Proposition 3.4: Given $S \subset B_i \in \mathcal{B}^{\mathfrak{F}}$,

$$\eta(S) \leq \min\left(2^{|S|}-1, \sum_{\alpha \in \Gamma} \sum_i \eta(B_i \cap \phi(S, \alpha))\right).$$

Proof: Straightforward. □

We can apply this to Y_R and thus get the following.

Corollary 3.5: Given $\mathcal{B}^{\mathfrak{F}}$,

$$|Z_R| = \eta(Y_R) = \Sigma_i \eta(B_i) \leq \Sigma_i \min(2^{|B_i|}-1,$$
$$\Sigma_{\alpha \in \Gamma} \Sigma_j \eta(B_j \cap \phi(B_i, \alpha))). \qquad \square$$

Now, a recursive application of Proposition 3.4 will give us a bound that gets progressively tighter with each application. If at any time $2^{|S|} - 1$ is a better bound for some set S, then clearly there is no reason to apply the proposition further after that step. However, this algorithm may in general require an exponential amount of computation if iterated to the fullest. For example, this is the case for the example in Fig. 7. On the other hand, the algorithm may be terminated at any step by using the bound $2^{|S|} - 1$. Alternatively, the following approximation can be used to compute a bound using less computation.

We now replace the summation over Γ in Proposition 3.4 by an approximation as follows. Given $S, Q \subset Y$, let $\rho(S, Q)$ denote the number of observable events that take states in $R(A|\bar{\Gamma}, Q)$ to states in S:

$$\rho(S, Q)=|\{\alpha \in d(R(A|\bar{\Gamma}, Q))$$
$$\cap\, \Gamma \mid f(R(A|\bar{\Gamma}, Q), \alpha) \cap S \neq \emptyset\}|. \tag{3.7}$$

First of all, note that

$$\sum_{\alpha \in \Gamma} \eta(B_i \cap \phi(S, \alpha)) \leq \rho(S, B_i \cap \phi(S, \Gamma))$$
$$\cdot \max_{\alpha \in \Gamma} \eta(B_i \cap \phi(S, \alpha)). \tag{3.8}$$

Since computing the maximization requires computing $\eta(B_i \cap \phi(S, \alpha))$ for each α, we replace it with $\eta(B_i \cap \phi(S, \Gamma))$ instead. Then,

$$\sum_{\alpha \in \Gamma} \eta(B_i \cap \phi(S, \alpha)) \leq \rho(S, B_i \cap \phi(S, \Gamma))\eta(B_i \cap \phi(S, \Gamma)). \tag{3.9}$$

We thus have the following result.

Proposition 3.6: Given $S \subset B_i \in \mathcal{B}^{\mathfrak{F}}$,

$$\eta(S) \leq \min\left(2^{|S|}-1, \sum_i \rho(S, \tau_i(S))\eta(\tau_i(S))\right)$$

where

$$\tau_i(s)=B_i \cap \phi(S, \Gamma).$$

Proof: Straightforward. □

We can apply this result to Y_R and we get the following.

Corollary 3.7: Given $\mathcal{B}^{\mathfrak{F}}$,

$$|Z_R| = \eta(Y_R) \leq \Sigma_i \min(2^{|B_i|}-1, \Sigma_j \rho(B_i, \tau_j(B_i))\eta(\tau_j(B_i))). \qquad \square$$

As before, Proposition 3.6 can be applied recursively. Alternately, one can terminate this algorithm at any step by using the bound $2^{|S|} - 1$. It is not known in general if the full iteration of the algorithm requires a polynomial or exponential number of steps. However, as the following example shows, it requires a *linear* number of steps for the system of Fig. 7, and in fact yields $|Z_R|$ exactly.

Example 3.8: For the system in Fig. 7, $\mathcal{B}^{\mathfrak{F}}$ consists of $B_1 = \{0\}$, $B_2 = \{11, \cdots, 16\}$, $B_3 = \{21, \cdots, 26\}$, $B_4 = \{31, \cdots, 36\}$, $B_5 = \{41\}$, $B_6 = \{42\}$, $B_7 = \{43\}$, $B_8 = \{44\}$, $B_9 = \{45\}$, and $B_{10} = \{46\}$. Let us use η_i as a shorthand for $\eta(B_i)$. Then, clearly, $\eta_1 = \eta_5 = \cdots = \eta_{10} = 1$. On the other hand, since $\tau_1(B_2) = \{0\}$ and $\rho(B_2, \tau_1(B_2)) = 2$, $\eta_2 \leq 2\eta_1 = 2$. Similarly,

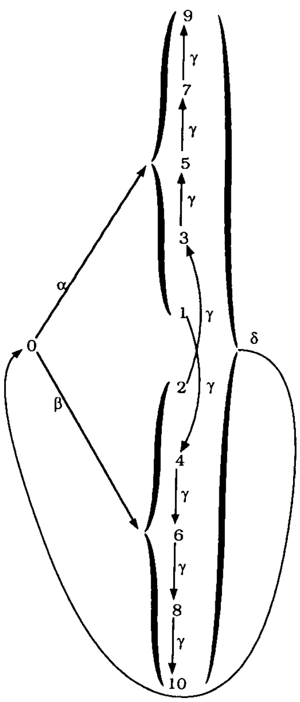


Fig. 8. Example for linear observer state space.

$\eta_3 \leq 2\eta_2 = 4$ and $\eta_4 \leq 2\eta_3 = 8$. Therefore, for this example,

$$|Z_R| \leq 1+2+4+8+1+1+1+1+1+1=21$$

and, in fact, this is the exact value of $|Z_R|$. □

We conclude this section by presenting the following class of systems for which the cardinality of the observer state space is linear in n, and our algorithm for computing a bound for $|Z_R|$ also yields $|Z_R|$ exactly.

Example 3.9: Consider the following class of systems, indexed by i (see Fig. 8 for $i = 4$). The set of events for this class consists of α, β, δ and γ, where all of them are observable. There are $2(i + 1) + 1$ states and one of them is state 0. The event α (respectively, β) defines a transition from 0 to the odd-numbered (respectively, even-numbered) states. The event δ defines transitions from all other states to state 0. The event δ defines a transition from state 1 to 4, from 2 to 3, and for all other states j with $j \geq 3$, γ defines a transition from j to $j + 2$. These systems are all observable (in fact a-observable), and Z_R is linear in i. For $i = 4$, $\mathcal{B}^{\mathcal{F}}$ consists of $B_1 = \{0\}$ and $B_2 = \{1, \cdots, 10\}$. Clearly, $\eta_1 = 1$. On the other hand, to calculate η_2, we need to know $\eta(\{1, \cdots, 8\})$, which we denote by η_3. Similarly, to calculate η_3, we need to know $\eta(\{1, \cdots, 6\})$, which we denote by η_4. Denoting $\eta(\{1, \cdots, 4\})$ by η_5, and $\eta(\{1, 2\})$ by η_6, and arguing as above, we see that we need to calculate η_6 first. Since $\eta_6 \leq \min(2^2, 2\eta_1) = 2$, $\eta_5 \leq \min(2^4, 2\eta_1 + \eta_6) = 4$. Similarly, $\eta_4 \leq 6$, etc., and thus $\eta_2 \leq 10$. Therefore, $|Z_R| \leq 1 + 10 = 11$ and, in fact, this is the exact value of Z_R. □

IV. Resilient Observers

In this section, we introduce the possibility of measurement error in our model, and address a problem of resilient observability. Specifically, suppose that the output string that we observe contains errors. Then a major question is how this measurement error affects the behavior of the observer. In particular, does it lead to catastrophic error propagation, or does the observer resume desired, correct behavior in a finite number of transitions. Let us consider three types of measurement errors.

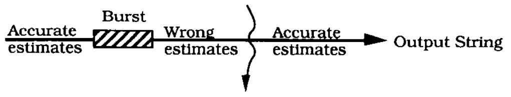


Fig. 9. Resilient observability. Following a burst of measurement error, observer estimates can only be wrong for a finite number of transitions.

- Although the system did not have any transitions, a transition has been mistakenly inserted.
- A transition has been mistaken for another.
- An observable transition has been totally missed in the output string.

An output corrupted with a burst of such measurement errors can be modeled by taking out a finite length string from the output string and replacing it with an arbitrary finite length string over Γ. Our goal here is to design resilient observers so that after a burst of measurement errors, the observer resumes correct behavior in a finite number of transitions, i.e., the actual state of the system in an element of the observer estimate. This is illustrated in Fig. 9.

Since we allow the burst to be any string in Γ, the corrupted output is *not* necessarily an output string that can be generated by a state in X, and thus the response of O, as we have specified it so far, is undefined for this erroneous string. Thus, we must extend the observer so that it is defined for all such strings.

Definition 4.1: An observer is a map $\boldsymbol{B}:\Gamma^* \to 2^Y$ so that for those strings that *can* occur in A, $\boldsymbol{B}$ yields the same behavior as O, i.e., for any $x \in X$ and $s \in L_f(A, x)$, we require that

$$B(h(s)) = \{y \in Y \mid \exists z \in Y,\ r \in L_f(A, z)$$
$$\text{such that } y \in f(z, r) \quad \text{and } h(r) = h(s)\}. \quad \square$$

There is one special observer that will deserve particular attention. Specifically, not all events γ may be defined at certain states of O. For any such state and event, we then define a transition, back to the "know nothing" state $\{Y\}$—i.e., the observer is simply reset if an inconsistent event occurs. We denote this observer by $O_R = (F, w_R, v_R)$, and mathematically, it is obtained from O as follows:

$$w_R(\hat{x}, \gamma) = \begin{cases} w(\hat{x}, \gamma) & \text{if } \gamma \in v(\hat{x}) \\ \{Y\} & \text{otherwise} \end{cases} \tag{4.1}$$

$$v_R(\hat{x}) = \Gamma. \tag{4.2}$$

As before, the initial state of O_R is the state $\{Y\}$. Note that O_R does define a map from Γ^* to 2^Y and, thus, by a mild abuse of terminology, we refer to the system or the map as an observer. Note also that O_R is not stable with respect to its singleton states, but $A \| O_R$ is stable with respect to the composite states at which the observer is at a singleton state and the system is also at that state.

Proposition 4.2: $A \| O_R$ is stable with respect to $\{(x, \{x\}) | x \in Y\}$.

Proof: Straightforward. □

In order to define what we mean by a resilient observer, we also need to define a notion to represent the discrepancy between two strings. There are many ways to define this, all of which depend on the reference point for comparing two strings. Since the actual point that the burst ends is important for our definition of resiliency, we compare two strings from their beginning, and we represent their discrepancy by how much they differ at the end. In particular, we say that the discrepancy between two strings s and t is of length (at most) i, denoted by

$$\xi(s, t) \leq i \tag{4.3}$$

if there exists a prefix, p, of both s *and* t such that $|s/p| \leq i$ and $|t/p| \leq i$. Now we can precisely define what we mean by a resilient observer $\boldsymbol{B}$.

Definition 4.3: $\boldsymbol{B}$ is a resilient observer if for all strings s that can be generated by A, i.e.,

- $\forall x \in X$,
- $\forall s \in L_f(A, x)$,

for all possible output strings t which can be generated by corrupting $h(s)$ with a finite length burst, i.e.,

- $\forall$ positive integers i,
- $\forall t \in \Gamma^*$ such that $\xi(t, h(s)) \leq i$,

and for all possible completions r of s with a suffix of length at least nq^2 (so that the observer has enough time to recover), i.e.,

- $\forall r \in L_f(A, x)$ such that $|r| \geq |s| + nq^2$ and s is a prefix of r,

the observer estimate, in response to the corrupted output $th(r/s)$, includes the current state of the system:

$$f(x, r) \subset \boldsymbol{B}(th(r/s)). \qquad \square$$

Note that in the case of a number of finite bursts that are spaced far enough apart, the estimates of a resilient observer are guaranteed to be correct starting from a finite number of transitions following each burst, up to the occurrence of the next burst. On the other hand, if the number of correct measurements between each burst is less than q^2, then we cannot guarantee any correct state estimates.

Existence of a resilient observer does not necessarily imply that the system is observable. That is, all we require is that resilient observers resume correct estimates in a finite number of transitions following a burst.

Proposition 4.4: A resilient observer $\boldsymbol{B}$, for A, exists iff $A \| O_R$ is E_1-stable, where

$$E_1 = \{(x, \hat{x}) | x \in \hat{x} \in Z\}.$$

Proof:

($\rightarrow$) Straightforward by assuming the contrary.

($\leftarrow$) Obvious, since then O_R is a resilient observer. □

What this proposition implies is that we only need to look at O_R to check resiliency. The stability condition on O_R simply states that after a finite number of steps following an error, the composite $A \| O_R$ returns to a state so that the estimate provided by the state $\hat{x}$ of O does indeed include the true state, x, of A. In general, since the observer state space may be exponential in q, checking stability may be computationally difficult. However, if we have WD observability—which can be checked by a test of polynomial complexity—resiliency is guaranteed.

Lemma 4.5: If A is WD observable, then $A \| O_R$ is E_1-stable.

Proof: Straightforward by assuming the contrary, since if $A \| O_R$ is not E_1-stable, there exists a cycle $(x_1, \hat{x}_1), \cdots, (x_k, \hat{x}_k), (x_1, \hat{x}_1)$ in $Y \times Z$ such that $x_i \notin \hat{x}_i$ for all i. Thus, there exists a cycle $(x_1, y_1), \cdots, (x_k, y_k), (x_1, y_1)$ in $Y \times Y$ such that $y_i \in \hat{x}_i$ and (x_i, y_i) is an indistinguishable pair, for all i. By Proposition 2.18, A is *not* WD observable, and we establish a contradiction. Therefore, $A \| O_R$ is E_1-stable. □

When we have observability or WD observability, O_R actually has a much stronger property. We need the following definition.

Definition 4.6: A system is ***resiliently observable*** (respectively, resiliently WD observable) if the system is observable (respectively, WD observable) and a resilient observer exists. □

Consider the observer O_R and its composition, $A \| O_R$, with A. Let E_2 be the set of composite states where the observer makes the precise and correct estimate, i.e., $E_2 = \{(x, \{x\}) | x \in X\}$. Then, we have the following.

Proposition 4.7: A is resiliently observable iff $A \| O_R$ is E_2-stable.

Proof: Straightforward by using Lemma 4.5. □

Finally, the following result shows that we do not need any test for resilient observability, since observability itself is necessary and sufficient for resilient observability.

Proposition 4.8: A is resiliently observable (respectively, resiliently WD observable) and O_R is a resilient observer iff A is observable (respectively, WD observable).

Proof:

($\rightarrow$) Obvious.

($\leftarrow$) Straightforward using Lemma 4.5. □

V. Conclusions

In this paper, we have introduced notions of observability, and resiliency for discrete-event systems described by finite-state automata, and we have developed polynomial algorithms to test for observability, resiliency, and to construct resilient observers. We showed that a central element in these concepts is the notion of stability that we considered in a previous paper [12]. We have also shown that an observer may be implemented in polynomial time, but the cardinality of its state space may be exponential. Although this issue is not of practical importance for the problems discussed in this paper, it is of central importance for problems of stabilization by output feedback that will be addressed in a forthcoming paper.

As we have seen, if a system is observable, the canonic observer O_R is always resilient, i.e., catastrophic error propagation will never occur. In a subsequent paper, we address the problem of invertibility, i.e., of deducing the entire event string from the output string, and we also introduce the notion of error recovery or resiliency in that context. In that case, invertibility is *not* enough to guarantee the existence of a resilient inverter, and further conditions are required to ensure resiliency and the absence of catastrophic error propagation. These notions would seem to be of value in trying to characterize the coordinated behavior of interconnections of DEDS and the ability of the composite to recover from a loss of coordination.

References

[1] R. Cieslak, C. Desclaux, A. Fawaz, and P. Varaiya, "Supervisory control of discrete-event processes with partial observations," *IEEE Trans. Automat. Contr.*, Mar. 1988.

[2] G. A. Frank, D. L. Franke, and W. F. Ingogly, "An architecture design and assessment system," *VLSI Design*, Aug. 1985.

[3] W. B. Gevarter, "Expert systems: Limited but powerful," *IEEE Spectrum*, Aug. 1983.

[4] W. M. L. Holcombe, *Algebraic Automata Theory*. London, England: Cambridge University Press, 1982.

[5] F. Lin and W. M. Wonham, "Decentralized supervisory control of discrete event systems," Syst. Contr. Group Rep. 8612, Univ. Toronto, July 1986.

[6] ——, "On observability of discrete event systems," *Inform. Sci.*, vol. 44, no. 3, 1988.

[7] M. E. Merchant, "Production: A dynamic challenge," *IEEE Spectrum*, May 1983.

[8] J. S. Ostroff and W. M. Wonham, "A temporal logic approach to real time control," in *Proc. CDC*, Dec. 1985.

[9] C. M. Özveren, "Analysis and control of discrete event dynamic systems: A state space approach," Ph.D. dissertation, M.I.T., Cambridge, MA, Aug. 1989; and Lab. Inform. Decision Syst. Rep. LIDS-TH-1907.

[10] C. M. Özveren and A. S. Willsky, "Aggregation and multi-level control in discrete event dynamic systems," Lab. Inform. Decision Syst. Rep. LIDS-P-1902, M.I.T., Cambridge, MA, Aug. 1989; also submitted to *Automatica*.

[11] ——, "Invertibility of discrete event dynamic systems," Lab. Inform. Decision Syst. Rep LIDS-P-1895, M.I.T., Cambridge, MA, July 1989; also submitted to *MCSS*.

[12] C. M. Özveren, A. S. Willsky, and P. J. Antsaklis, "Stability and stabilizability of discrete event dynamic systems," Lab. Inform. Decision Syst. Rep. LIDS-P-1853, M.I.T., Cambridge, MA, Feb. 1989; also submitted to *J. ACM*.

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

[14] P. J. Ramadge and W. M. Wonham, "Modular feedback logic for discrete event systems," *SIAM J. Contr. Opt.*, Sept. 1987.

[15] ——, "Supervisory control of a class of discrete event processes," *SIAM J. Contr. Opt.*, Jan. 1987.

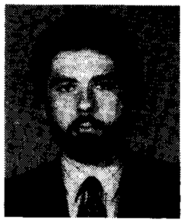
[16] M. N. S. Swamy and K. Thulasiraman, *Graphs, Networks, and Algorithms*. New York: Wiley, 1981.

[17] G. Tadmor and O. Z. Maimon, "Control of large discrete event systems: Constructive algorithms," LIDS Publ. LIDS-P-1627, M.I.T., Dec. 1986.

[18] L. Tobias and J. L. Scoggins, "Time-based air-traffic management using expert systems," *IEEE Contr. Syst. Mag.*, Apr. 1987.

[19] J. N. Tsitsiklis, "On the control of discrete event dynamical systems," *Math. C. S. S.,* 1989.

[20] A. F. Vaz and W. M. Wonham, "On supervisor reduction in discrete event systems," *Int. J. Contr.*, 1986.



Cüneyt M. Özveren (S'81–M'90) was born in Istanbul, Turkey, on July 20, 1962. He received the B.S. and M.S. degrees in electrical engineering and computer science, the Electrical Engineer degree, the M.S. degree from the Sloan School of Management, and the Ph.D. degree in electrical engineering, all from the Massachusetts Institute of Technology, Cambridge, in 1984, 1987, 1987, 1989, and 1989, respectively.

He is currently a Senior Software Engineer at Digital Equipment Corporation, working on the design and the implementation of a high-speed communications switch. From January to August 1988 he conducted research at the Institut de Recherche en Informatique et Systèmes Aléatoires, France, and from September to December 1989 he was a Postdoctoral Research Associate at the Laboratory for Information and Decision Systems at M.I.T. His interests are associated with the analysis and control of large scale dynamic systems including applications to communications systems, manufacturing systems, and economics.

Dr. Özveren is a member of Sigma Chi, Tau Beta Pi, and Eta Kappa Nu. In 1989 he was a finalist for the 28th IEEE Conference on Decision and Control Best Student Paper Award. He is also the 1989 recipient of the Pugh-Roberts Associates Prize in Computer Simulation Applied to Corporate Strategy.

Alan S. Willsky (S'70–M'73–SM'82–F'86) received the S.B. and Ph.D. degrees from the Massachusetts Institute of Technology in 1969 and 1973, respectively.

From 1969 through 1973 he held a Fannie and John Hertz Foundation Fellowship. He joined the M.I.T. Faculty in 1973, and his present position is Professor of Electrical Engineering. From 1974 to 1981 he served as Assistant Director of the M.I.T. Laboratory for Information and Decision Systems. He is also a founder and member of the board of directors of Alphatech, Inc. He has held visiting positions at Imperial College, London; L'Université de Paris-Sud; and the Institut de Recherche en Informatique et Systèmes Aléatoires in Rennes, France. He is Editor of the M.I.T. Press series on signal processing, optimization, and control. He has been an Associate Editor of several journals. He is the author of the research monograph *Digital Signal Processing and Control and Estimation Theory* and is co-author of the undergraduate text *Signals and Systems.* His present research interests are in problems involving multidimensional and multiresolution estimation and imaging, discrete-event systems, and the asymptotic analysis of control and estimation systems.

Dr. Willsky has been an Associate Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, and has served as a member of the Board of Governors and as Vice President for Technical Affairs of the IEEE Control Systems Society, and was Program Chairman for the 1981 Bilateral Seminar on Control Systems held in the People's Republic of China. In addition, he gave the opening plenary lecture at the 20th IEEE Conference on Decision and Control, and in 1988 was made a Distinguished Member of the IEEE Control Systems Society. He was Program Chairman for the 17th IEEE Conference on Decision and Control. In 1975 he received the Donald P. Eckman Award from the American Automatic Control Council. He was awarded the 1979 Alfred Noble Prize by the ASCE and the 1980 Browder J. Thompson Memorial Prize Award by the IEEE for a paper excerpted from his monograph.