Output Stabilizability of Discrete-Event Dynamic Systems

Cüneyt M. Özveren, *Member, IEEE,* and Alan S. Willsky, *Fellow, IEEE*

Abstract—**In this paper, we investigate the problem of designing stabilizing feedback compensators for discrete-event dynamic systems (DEDS) modeled as finite-state automata in which some transition events are controllable and some events are observed. The problem of output stabilization is defined as the construction of a compensator such that all state trajectories in the closed-loop system go through a given set E infinitely often. We also define a stronger notion of output stabilizability which requires that the state not only pass through E infinitely often but that the set of instants when the state is in E and we *know* it is in E is also infinite. Necessary and sufficient conditions are presented for both notions. We also introduce and characterize a notion of resiliency that corresponds to the system being able to recover from observation errors. Finally, an important issue in all problems involving DEDS is computational complexity. We provide some general bounds for our algorithms and discuss several conditions under which far smaller bounds can be achieved.**

I. Introduction

DISCRETE-EVENT Dynamic Systems (DEDS) are dynamic systems, for which the evolution of the state is triggered by the instantaneous occurrence of discrete events. Such behavior can be found in many complex, man-made systems at some level of abstraction, such as flexible manufacturing systems and communication systems. Although DEDS have been studied extensively by computer scientists, the notion of control of a DEDS has been introduced only recently [9], [10], [12]. This work has prompted a considerable response from other researchers in the field in exploring alternate formulations and questions that build on the foundations of both computer science and control. Our work here and in [5] and [6] is very much in that spirit with, perhaps, closer ties to more standard control concepts.

Our work differs in several important ways from other approaches found in the literature. Much of this work, for example in [1], [2], [8], [10], [12], concerns itself with what can be thought of as linguistic questions in which the objective is to control the system so that the resulting event sequence lies in a desired set or *language* of strings of events. In contrast, we focus here and in [6] on controlling the *state* of the DEDS so that it returns regularly to a specified set of states E. Obviously, there are relationships between these settings just as there is between standard control problems dealing with input–output behavior and those dealing with internal state dynamics. Indeed in another part of our work [4] we make this connection much more explicit by constructing "target" state sets from which particular "tasks"—modeled as sets of desired event strings—can be initiated. However, the more important difference between our state-based approach and the linguistic methods pursued in the literature is our emphasis on *stability* and *error recovery*. For example, in a complex manufacturing system or a command and control system, desired behavior involves the coordinated action of many component systems. Such systems are, however, subject to failures, errors, and other anomalous events, e.g., a transmission line failure in an interconnected power plant, an error in routing or coordination in an assembly process, or an incorrect interpretation of an alarm signal in a nuclear power plant. Obviously, in such a case, we would like to guarantee that the system is fail-safe, i.e., that it can recover from such events and avoid the catastrophic propagation of undesirable events or errors following an initiating anamoly that, for example, are characteristic of large-scale blackouts, computer network crashes, or the behavior of poorly-designed convolutional decoders (where the term "catastrophic error propagation" was coined). In our context, we think of E as the set of states from which normal (and desirable) behavior can commence, e.g., states in a manufacturing system corresponding to the system being set up for the production of a particular set of products. The normal operation of the system will certainly take us out of E, as will the occurrence of anomalies, but what we wish to guarantee is that eventually we will return to E. Thus, as in standard feedback control contexts, we want to make sure that the transient response of our DEDS has desirable properties, in that we eventually return to effective operation following an anomaly.

The second important difference between the framework we consider here and those found in the literature is the nature of the observation model used. In contrast to the approaches found in [1]–[3], [7], [11] in which observations include partial state and/or event information at each point in time, we consider a model in which we receive information intermittently about the evolution of the DEDS when certain "key" events occur. This type of event-driven observation seems to be natural in many complex systems such as manufacturing or command and control systems in which only certain events (such as the completion of a task, the break-

Manuscript received July 28, 1989; revised September 15, 1990 and January 14, 1991. Paper recommended by Past Associate Editor, X.-R. Cao. 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 DAAL03-860K0171.

C. M. Özveren is with the Telecommunications and Networking Department, Digital Equipment Corporation, Littleton, MA 01460.

A. S. Willsky is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139.

IEEE Log Number 9100971.

down of a machine, the detection of a target) trigger the transmission of messages to a supervisor. One of the crucial issues that such a measurement model captures is that in a complex discrete-event system it is often the *timing* of information and events that is important. That is, for effective control of such systems—such as in preventing an initiating event sequence from causing a large-scale blackout in an interconnected power system—it is critical that sufficient information is available at a point at which there are effective control actions that can be taken. As shown in [5], one of the important aspects of our measurement model is that the level of our knowledge of the state fluctuates, so that, for example, the state may be known perfectly at intermittent points in time. As we will see, the design of output compensators must then explicitly deal with the timing of information and control action. For example, the availability of sufficient information to reconstruct the state of intermittent points together with the stabilizability of the system by full state feedback do *not* guarantee that the system can be stabilized by output feedback, as one must indeed make sure that the information is available when it is needed. The use of a state-based framework allows us to express this crucial idea in an explicit and simple way.

In the next section, we introduce the mathematical framework considered in this paper and briefly review those aspects of [5] and [6] that we use here. The core of this paper is Section III in which we formulate two notions of output stabilization and present algorithms for constructing compensators for both. Since the observers we construct are DEDS which keep track of *all* possible trajectories consistent with the observed events, we will see that we can recast the output stabilization problem as the stabilization of the *observer* by *state* feedback. Also, since our observations are sporadic, we may or may not know exactly *when* the system has recovered and returned to E, and it is this distinction that leads to the two notions of stabilization that we investigate. Finally, in Section IV, we present an informal discussion of several related topics. In particular, as made clear in [11], computational complexity is often a critical problem in DEDS control problems, and we describe how this issue enters in the output stabilization problem and discuss conditions under which efficient solutions can be obtained. In addition, we also extend the designs of Section III to obtain compensators that are resilient in the face not only of system anomalies but also observation errors.

II. Background and Preliminaries

A. System Model

The class of systems we consider are defined over the following quadruple:

$$G = (X, \Sigma, \Gamma, U) \tag{2.1}$$

where X is the finite set of states with $n = |X|$, Σ is the finite set of possible events, $\Gamma \subset \Sigma$ is the set of observable events, and $U = 2^{\Phi}$, where $\Phi \subset \Sigma$ is the set of controllable events. The dynamics defined on G that we consider are of the following form:

$$x[k+1] \in f(x[k], \sigma[k+1]) \tag{2.2}$$

$$\sigma[k+1] \in (d(x[k]) \cap u[k]) \cup (d(x[k]) \cap \bar{\Phi}). \tag{2.3}$$

Here, $x[k] \in X$ is the state after the kth event, $\sigma[k] \in \Sigma$ is the $(k+1)$st event, and $u[k] \in U$ is the control input after the kth event and $\bar{\Phi}$ denotes the complement of Φ. The function $d: X \to 2^{\Sigma}$ 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 $f: X \times \Sigma \to X$ is also set-valued, so that the state following a particular event is not necessarily known with certainty. The set $d(x)$ represents an "upper bound" on the set of events that can occur at state x, whereas the set $d(x) \cap \bar{\Phi}$, is a lower bound. The effect of our control action is adjusting the set of possible events between these bounds, by disabling some of the controllable events, i.e., elements of the set $d(x) \cap \Phi$. Note that with a slight increase in notational complexity, we can consider the slightly more general model in which the controllability of some events may vary from state to state (see [6]), and indeed all of the results in this paper can be extended to this case. Furthermore, we assume in this paper that $\Phi \subset \Gamma$. While it is again possible to extend our results to the case when this is not true, this assumption seems to be a natural one as it is consistent with the usual control formulation in which the control signals generated by a compensator are in fact observable by the compensator.

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 $h: \Sigma \to \Gamma \cup \{\epsilon\}$, 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. Finally, we mark the controllable events by ":u". Thus, in this example, $X = \{0, 1, 2, 3, 4\}$, $\Sigma = \{\alpha, \beta, \delta, \mu\}$, $\Gamma = \{\alpha, \beta\}$, and $\Phi = \{\alpha\}$. Note that in some cases (especially when considering DEDS representing observers and compensators), *all* events are observable. In such cases we will write our automaton as a triple $A = (G, f, d)$ with h understood to be the identity.

There are several basic notions that we will need in our investigation. The first is the notion of *liveness*. Intuitively, a system is alive if it cannot reach a point at which no event is

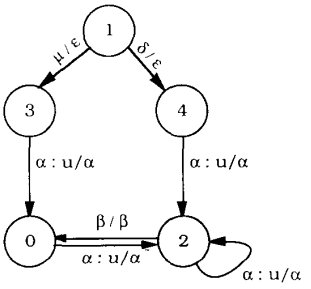


Fig. 1. A Simple Example.

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 $\Gamma_1 \cap S = \Gamma_2 \cap S$ and $\Phi_1 \cap S = \Phi_2 \cap S$ (i.e., any shared event observable (controllable) in one system is also observable (controllable) 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

$$G_{12} = \left(X_1 \times X_2, \Sigma_1 \cup \Sigma_2, \Gamma_1 \cup \Gamma_2, 2^{\Phi_1 \cup \Phi_2}\right) \tag{2.6}$$

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

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

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

Here, we have extended each f_i to all of $\Sigma_1 \cup \Sigma_2$ in the trivial way, namely, $f_i(x_1, \sigma) = x_1$ if $\sigma \notin \Sigma_i$. Also, the combined effect of the individual inputs $u_1[k] \subset U_1$ and $u_2[k] \subset U_2$ in the dynamics of the composite is given by

$$u_{12}[k] = \left(u_1[k] \cap \bar{S}\right) \cup \left(u_2[k] \cap \bar{S}\right) \cup \left(u_1[k] \cap u_2[k]\right). \tag{2.10}$$

B. Stability and Stabilizability

In [6], we define a notion of stability which requires that the trajectories go through a given set E infinitely often.

Definition 2.1: Let E be a specified subset of X. A state $x \in X$ is *E-prestable* if there exists some integer i such that every trajectory starting from x passes through E in at most i transitions. The state $x \in X$ is *E-stable* if A is alive and every state reachable from x is E-prestable. The DEDS is *E-stable* (respectively, *E-prestable*) if every $x \in X$ is E-stable (respectively, E-prestable). □

As discussed in Section I, the Set E should be thought of as the set of states from which the desired system behavior can commence, and thus we wish to return to it after the occurrence of an anomaly or the completion of some operation.

Definition 2.2: The *radius* of A is the length of the longest cycle-free trajectory between any two states of A. The *E-radius* of an E-stable system A is the maximum number of transitions it takes any trajectory to enter E. □

Note that an upper bound on both the radius and the E-radius, for any E, of an E-stable system is n. We refer the reader to [6] for a more complete discussion of this subject and for an $O(n^2)$ test for E-stability of a DEDS. Finally, we note that in [6] and Definition 2.1, we require liveness in order for a system to be stable so that trajectories can be continued indefinitely. While we will continue to require liveness in this paper as we consider compensator design, there are occasions on which it is useful to consider a notion of *weak stability*, in which all the conditions of Definition 2.1 are met except that A may not be alive. Thus, for a weakly E-stable system, all trajectories pass through E and can only die in E. We note without proof that the algorithm developed in [6] for stability can be used without change to test for weak stability.

In [6], we study stabilization by state feedback. Here, a state feedback law is a map $K: X \to U$ and the resulting closed-loop system is $A_K = (G, f, d_K, h)$ where

$$d_K(x) = \left(d(x) \cap K(x)\right) \cup \left(d(x) \cap \bar{\Phi}\right). \tag{2.11}$$

Definition 2.3: A state $x \in X$ is *E-prestabilizable* (respectively, *E-stabilizable*) if there exists a state feedback K such that x is E-prestable (respectively, E-stable) in A_K. The DEDS is *E-stabilizable* if there exists a state feedback K such that A_K is E-stable. □

We refer the reader to [6] for a more complete discussion of this subject and for an $O(n^3)$ test for E-stabilizability of a DEDS, which also provides a construction for a stabilizing feedback.

C. Observability and Observers

In [5], we consider the problem of constructing estimates of the current state of a DEDS based on knowledge only of the observable event sequence. In order for this to be meaningful, we obviously would like to preclude the possibility that our DEDS can generate arbitrarily long sequences of unobservable events, i.e., events in $\bar{\Gamma}$. A necessary and sufficient condition for checking this is that if we remove the observable events, the resulting automaton $A \mid \bar{\Gamma} = (G, f, d \cap \bar{\Gamma}, h)$ must be weakly D_O-stable, where D_O is the set that only have observable transitions defined, i.e., $D_O = \{x \in X \mid d(x) \cap \bar{\Gamma} = \emptyset\}$. This is not difficult to check and will be assumed.

We will make use of some notation introduced in [5], [6]. Specifically, $R(A, x)$ denotes the set of states reachable from x, and we let Y denote the set of states that either have observable transitions defined to them or that are purely initial in that there are no transitions to them from any state. Let $q = |Y|$. Also, 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. Also, let $L_f(A, x)$ be the set of strings in $L(A, x)$ that have an observable event as the last event, and let $\bar{L}(A) =$

$\cup_{x \in X} L(A, x)$ be the set of all event trajectories that can be generated by A. Finally, 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.

In [5], we present 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. Mathematically, if we let a function $\hat{x}: h(\overline{L}(A)) \to 2^Y$ denote the estimate of the current state given the observed output string $t \in h(\overline{L}(A))$, then

$$\hat{x}(t) = \{x \in Y \mid \exists y \in X \text{ and } s \in L_f(A, y) \text{ such that } h(s) = t \text{ and } x \in f(y, s)\}. \quad (2.12)$$

The observer, for which the state space is a subset Z of 2^Y, and the events and observable events are both Γ, is a DEDS which realizes this function. Suppose that the present observer estimate is $\hat{x}[k] \in Z$ and that the next observed event is $\gamma[k+1]$. The observer must then account for the possible occurrence of one or more observable events prior to $\gamma[k+1]$ and then the occurrence of $\gamma[k+1]$

$$\hat{x}[k+1] = w(\hat{x}[k], \gamma[k+1]) \triangleq \cup_{x \in R(A \mid \overline{\Gamma}, \hat{x}[k])} f(x, \gamma[k+1]) \quad (2.13)$$

$$\gamma[k+1] \in v(\hat{x}[k]) \triangleq h(\cup_{x \in R(A \mid \overline{\Gamma}, \hat{x}[k])} (d(x) \cap u[k]) \cup (d(x) \cap \overline{\phi})). \quad (2.14)$$

The set Z is then the reach of $\{Y\}$ using these dynamics, i.e., we start the observer in the state corresponding to a complete lack of state knowledge and let it evolve.[1] Our observer then is the DEDS $O = (F, w, v)$ (with identity output), where $F = (Z, \Gamma, \Gamma, U)$. The observer for the example in Fig. 1 is illustrated in Fig. 2.

In [5], we investigate a notion of observability corresponding to the requirement that at intermittent points we know the state of A exactly, which is equivalent to O being stable with respect to its singleton states. We also show that if A is observable then all trajectories from an observer state pass through a singleton state in at most q^2 observable transitions. Thus, since there are at most q singleton states, the longest cycle-free path in O must have length less than $q^2(q+1)$ so that the radius of O is at most $O(q^3)$. This will play an important role in bounding both the complexity of our algorithms and the maximum number of transitions it takes a trajectory from a state, in an output stabilizable system, to pass through E.

An important aspect of our work is our treatment of resiliency or error recovery. Specifically, suppose that the observed sequence of transitions includes errors corresponding to inserted events, missed events, or mistaken events. We term an observer *resilient* if after a finite burst of such measurement errors, the observer resumes correct behavior in a finite number of transitions, i.e., the current observer estimate includes the current state of the system. In [5], we construct a resilient observer by extending w and v as follows:

$$w_R(\hat{x}, \gamma) = \begin{cases} w(\hat{x}, \gamma) & \text{if } \gamma \in v(\hat{x}) \\ \{Y\} & \text{otherwise} \end{cases} \quad (2.15)$$

$$v_R(\hat{x}) = \Gamma. \quad (2.16)$$

That is, if an observed event γ is inconsistent with the present estimate $\hat{x}$ of the system state, we reset the observer to the initial state $\{Y\}$. The resulting system $O_R = (F, w_R, v_R)$ is a resilient observer if A is observable [5].

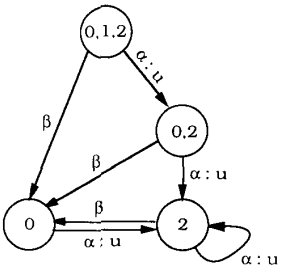


Fig. 2. Observer for the system in Fig. 1.

D. Compensators

We define a compensator as a map $C: \Gamma^* \to U$. Then, the closed-loop system A_c is the same as A but with

$$\sigma[k+1] \in d_C(x[k], s[k]) \triangleq (d(x[k]) \cap C(h(s[k]))) \cup (d(x) \cap \overline{\Phi}) \quad (2.17)$$

where $s[k] = \sigma[0] \cdots \sigma[k]$ with $\sigma[0] = \epsilon$: for output stabilizability, we only need to define compensators for strings $h(\overline{L}(A))$. However, when we talk about resiliency in Section V, we need to worry about defining C for arbitrary strings in Γ^*.

One constraint we wish to place on our compensators is that they preserve liveness. Thus, suppose that we have observed the output string s, so that our observer is in $\hat{x}(s)$ and our control input is $C(s)$. Then, we must make sure that any x reachable from any element of $\hat{x}(s)$ by *unobservable events only* is alive under the control input $C(s)$. That is, for all $x \in R(A \mid \overline{\Gamma}, \hat{x}(s))$, $d_C(x, s)$ should *not* be empty. This leads to the following definition.

Definition 2.4: Given $Q \subset X$, $F \subset \Phi$, F is *Q-compatible* if for all $x \in R(A \mid \overline{\Gamma}, Q)$, $(d(x) \cap F) \cup (d(x) \cap \overline{\Phi}) \neq \emptyset$. A compensator C is *A-compatible* if for all $s \in h(\overline{L}(A))$, $C(s)$ is $\hat{x}(s)$-compatible. □

Finally, note that the class of compensators that we have defined is quite large. As we will see in the next section, we can actually restrict attention to a computationally more useful subclass of compensators each of which can be realized as the cascade of a finite-state automaton followed by a memoryless function of the state of this automaton. For example, one class of compensators that we will encounter is

[1] To get Z, we consider the richest possible behavior by enabling all controllable events, i.e., we let $u[k] = \Phi$.

the class of *O-compatible* compensators, i.e., those that can be realized as the cascade of the observer O for A and a memoryless *observer feedback* function K so that $C(s) = K(w(\{Y\}, s))$. □

III. Two Notions of Output Stabilizability

The obvious notion of output E-stabilizability is the existence of a compensator C so that the closed-loop system A_C is E-stable. Because of the intermittent nature of our observations, it is possible that such a stabilizing compensator may exist, so that we are sure that the state goes through E infinitely often, but so that we never know *when* the state is in E. For this reason, we define a stronger notion of output stabilizability that not only requires that the state pass through E infinitely often but that we regularly know *when* the state has moved into E. We begin with this later notion which is easier to analyze. Also, in linear system theory, observability is not required for a system to be output stabilizable, and the basic results we present in this section do not assume observability. On occasion, however, we will see that the assumption of observability leads to useful complexity bounds.

A. Strong Output Stabilizability

Definition 3.1: A is *strongly output stabilizable* if there exists a compensator C and an integer i such that A_C is alive and for all $p \in \bar{L}(A_C)$ such that $|p| \geq i$, there exists a prefix t of p such that $|p/t| \leq i$ and $\hat{x}(h(t)) \subset E$, i.e., so that we *know* that the state is in E. We term such a compensator a *strongly output stabilizing* compensator.

The next result shows that we can restrict attention to observer feedback.

Proposition 3.2: A is strongly output stabilizable iff there exists a state feedback $K: Z \to U$ for the observer such that X_I in $A \| O_K$ is E_{OC}-stable, where $X_I = \{(x, \{Y\}) \mid x \in X\}$ is the set of possible initial states in $A \| O_K$ and where $E_{OC} = \{(x, \hat{x}) \in Y \times Z \mid \hat{x} \subset E\}$ is the set of composite states for which the system is in E and we know it.

Proof: We need only prove that given any strongly output stabilizing compensator C_1, we can construct another which is O-compatible. To begin, let l_i be the set of length i elements of $h(\hat{L}(A))$ and let $Z_1 = \{\{Y\}\}$ be the singleton set containing the initial state $\{Y\}$ of O. Also, let $K(\{Y\}) = C_1(\epsilon)$. Next, let $S_{11}, \cdots, S_{1k_1}$ be a collection of disjoint subsets of l_1 such that 1) $\cup_i S_{1i} = l_1$; 2) for all $\sigma \in S_{1i}$, $v(\{Y\}, \sigma) = \hat{x}_i$ for some $\hat{x}_i \in Z$; and 3) for any S_{1i}, S_{1j}, $i \neq j$, $\hat{x}_i \neq \hat{x}_j$. Let us term such a collection of subsets an l_1-*collection*. For each $\hat{x}_i \notin Z_1$, pick some $\alpha_1 \in S_{1i}$ and let $K(\hat{x}_i) = C_1(\alpha_i)$. Construct a compensator C_2 such that for all output strings of the form σs, for some $\sigma \in S_{1i}$, $C_2(\sigma s) = C_1(\alpha_i s)$. Clearly, C_2 is a strongly output stabilizing compensator for A. Also, let $Z_2 = Z_1 \cup \cup_i \hat{x}_i$ which denotes the set of observer states for which we have defined K so far.

We repeat this construction for l_2, l_3, etc. After step $j-1$, C_j is a strongly output stabilizing compensator for A, and we will have defined K for observer states Z_j that can be reached by $\{Y\}$ with output strings of length at most $j-1$. At step j, let $S_{j1}, \cdots, S_{jk_j}$ be the l_j-collection. For each $\bar{x}_i$ such that $v(\{Y\}, S_{ji}) = \hat{x}_i$ and $\hat{x}_i \notin Z_j$, pick some $a_i \in S_{ji}$ and let $K(\hat{x}_i) = C_j(a_i)$. Construct a compensator C_{j+1} such that for all output strings of the form ts, for some $t \in S_{ji}$, $C_{j+1}(ts) = C_j(r_i s)$. Clearly, C_{j+1} is a strongly output stabilizing compensator for A. Also, let $Z_{j+1} = Z_j \cup \cup_i \hat{x}_i$.

Proceed in this fashion until, at some step j, $Z_j = Z$, which implies that we have defined a feedback for all observer states. The reach of X_I in $A \| O_K$ is alive since by construction $K(\hat{x})$ is $\hat{x}$-compatible. Since also C_j is a strongly output stabilizing compensator for A, the compensator C defined by $C(s) = K(v(\{Y\}, s))$ is a strongly output stabilizing compensator for A. Therefore, X_I in $A \| O_K$ is E_{OC}-stable. □

Corollary 3.3: A is a strongly output stabilizable iff there exists a state feedback $K: Z \to U$ for the observer such that O_K is stable with respect to $E_O = \{\hat{x} \in Z \mid \hat{x} \subset E\}$ and for all $\hat{x} \in Z$, $K(\hat{x})$ is $\hat{x}$-compatible. Furthermore, if A is also observable, then the trajectories in the reach of X_I in $A \| O_K$ go through E_{OC} in at most $O(nq^3)$ transitions.

Proof: The first statement follows from the fact that O by itself captures the behavior of A. The second follows from the fact that E_O-radius of O_K (and hence the E_{OC}-radius of $A \| O_K$) is certainly no longer than the uncontrolled radius of O and from the fact that by assumption there are at most n unobservable transitions between observable events. □

As an example, consider the system in Fig. 3, where $E = \{1, 2\}$ and where all events are observable. Note that in this case, we need to check the stabilizability of the observer with respect to $E_O = \{2\}$. We achieve stability if α is disabled at the observer state $\{0, 2\}$.

Corollary 3.3 essentially tells us that we can test strong output stabilizability by testing the observer for stabilizability by (observer) state feedback, while preserving liveness of $A \| O_K$. In [6] we develop an algorithm that tests for and constructs stabilizing state feedback laws for DEDS. The following extends this, with the key difference being the check for compatibility.

Proposition 3.4: The following algorithm is a test for strong output stabilizability.

Algorithm: Let $Z_0 = E_O$ and iterate

$$P_{k+1} = \{\hat{x} \in Z \cap \bar{Z}_k \mid \{\gamma \in v(\hat{x}) \mid w(\hat{x}, \gamma) \in Z_k\} \text{ contains } v(\hat{x}) \cap \bar{\Phi} \text{ and is } \hat{x}\text{-compatible}\}$$

$$K(\hat{x}) = \{\gamma \in v(\hat{x}) \mid w(\hat{x}, \gamma) \in Z_k\} \cap \Phi \text{ for } \hat{x} \in P_{k+1}$$

$$Z_{k+1} = Z_k \cup P_{k+1}.$$

Terminate when $Z_{k+1} = Z_k \triangleq Z^*$. A is strongly output stabilizable iff $Z = Z^*$. The corresponding feedback is K as computed above. □

Here, Z_k consists of all observer states that can be driven to E_O in k or fewer steps by the application of $\hat{x}$-compatible feedback. The complexity of this algorithm is determined by the number of states in Z that must be examined at each step times the number of steps. The former is bounded by $|Z|$, while the latter is bounded by the radius of O. Thus, for an observable system we see that the complexity of this algorithm is $O(q^3 |Z|)$. Also, if a system is strongly output

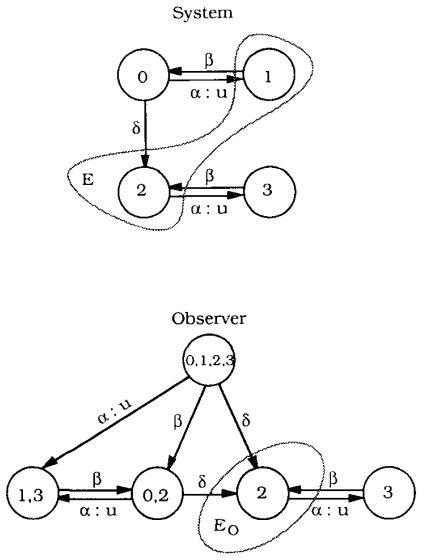


Fig. 3. Example for strong output stabilizability (all the events are observable).

stabilizable there is in general a range of stabilizing compensators ranging from maximally restrictive (in which we cannot disable any additional events while preserving liveness) to minimally restrictive (in which enabling any other events would violate stability). By modifying the aforementioned algorithm so as to compute at each stage those new observer states in $Z \cap \overline{Z}_k$ which can be driven to Z_k with the fewest enabled inputs (that still preserve liveness), we can obtain a maximally restrictive feedback. As discussed in [6], from this, one can also construct a minimally restrictive feedback by enabling events until stability is violated.

B. Output Stabilizability

In this section, we study the following somewhat weaker notion.

Definition 3.5: A is *output prestabilizable* (respectively, *output stabilizable*) with respect to E if there exists a compensator C such that for *any* initial state $x \in X$, the resulting X-trajectory in A_C is guaranteed to pass through E (respectively, through E infinitely open). □

A trivial observation is that in the case of full state feedback any feedback law $K(x)$ which makes A_K E-prestable while preserving liveness obviously makes A_K E-stable. In the case of output feedback, it is *not* true that an arbitrary compensator C that prestabilizes A_C while preserving liveness also stabilizes A_C. The reason for this is that C is a *dynamic* compensator, and there is no guarantee that after having initially driven the system through E that the dynamics of C will generate controls that guarantee return visits to E. If, however, we have a point in time by which we *know* that A_C has visited E, then we can simply restart the compensator to guarantee a return visit, i.e., by modifying the prestabilizing compensator, we can construct a stabilizing one.

To see that this is possible, suppose that C is an output prestabilizing compensator that preserves liveness. Then, for each $x \in X$, there exists an integer i such that the trajectories from x in A_C go through E in at most i transitions. Thanks to our assumption that A *cannot* generate arbitrarily long sequences of unobservable events, for each $x \in X$, there exists an integer j such that the trajectories from x in A_C go through E in at most j *observable* transitions. Let j^* be the maximum over all j. Then, we know that the trajectories in A_C go through E in at most j^* observable transitions independently of the initial state. This allows us to construct a stabilizing compensator C' by restarting C every j^* transitions. Specifically, given $s \in h(\overline{L}(A_C))$, let s^* denote the suffix of s for which $|s^*| = |s| \bmod j^*$, and let $C'(s) = C(s^*)$. Clearly, $A_{C'}$ is alive. Also, $A_{C'}$ is E-stable since it is guaranteed to go through E at least once every j^* observable transitions.

There are two important points in the preceding discussion. The first is that since we know that output prestabilizability plus liveness is equivalent to output stabilizability, we can focus on the former, simpler notion, i.e., we need only worry about driving the system to E and then specifying the restarting mechanism to guarantee return visits to E. The second is that the existence of j^* implies that output stabilization, when possible, can be accomplished via a compensator with finite memory. As we now show, in contrast to the strong output stabilization case in which the observer provided the required memory for stabilization, we will need a bit more memory here. In particular, our construction of a prestabilizing compensator involves 1) constructing a modified observer *which also keeps track of the states the system can be in if the trajectory has not yet passed through* E, 2) formulating the problem of prestabilizing A by output feedback as a problem of stabilzing this observer by state feedback, and 3) constructing a prestabilizing compensator by using this observer and the state feedback constructed in 2).

To provide the motivation behind our approach, consider the system in Fig. 3. For output stabilizability, we do not really need to disable α (as we had to for strong output stabilizability). Consider the loop in the observer that consists of the states $\{1, 3\}$ and $\{0, 2\}$. If the system is in state 1 (respectively, state 2), it is already in E. If the system is in state 3 (respectively, state 0), it makes a transition into E after the next event. Therefore, A is stable and thus is trivially output stabilizable (without disabling any event). This example illustrates the key idea in our analysis of output stabilizability: we must keep track of those state trajectories that have not yet passed through E; if that set becomes empty at some point, we will know that the system has passed through E, although we may not know the point in time at which it did.

The following construction allows us to perform this function: Delete all events in A that originate from the states in E and construct the corresponding observer. Let A_E denote this system and let $O_E = (F_E, w_E, v_E)$ denote its observer. For example, Fig. 4 illustrates such an automaton and observer for the system in Fig. 3. The observer O_E captures all the behavior of A until its trajectories enter E. When we

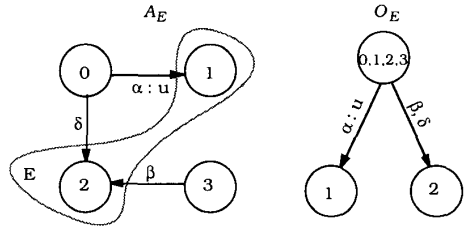


Fig. 4. Example for A_E and O_E (all the events are observable).

look at the states of O_E, we see that there are some "trapping" states, each of which is a subset of E and thus has no events defined. Let us consider an event trajectory s in A and the corresponding trajectory $h(s)$ in O_E that starts from the initial state $\{Y\}$. If the trajectory ever evolves to a "trapping" state in O_E, then we know that it has passed through E in A. Other states of O_E may have some elements in E and some elements that are *not* in E. Let $\hat{x}$ be such a state of O_E, then for a trajectory that evolves to $\hat{x}$, the system can be in one of the states in $\hat{x} \cap \bar{E}$ only if that trajectory has *not* passed through E yet. Even though O_E keeps track of trajectories that have *not* passed through E yet, it does *not* keep track of enough information to design a prestabilizing compensator, since, in order to preserve liveness, we also need to know *all* the states that the system can be in so that we can check if our control input keeps the system alive: the automaton

$$Q = (F_Q, w_Q, v_Q) = O_E \| O \tag{3.1}$$

together with the initial state (Y, Y) keeps track of *all* the information we need for designing an output stabilizing compensator. Note that

$$w_Q((y_1, y_2), \sigma) = (w_E(y_1, \sigma), w(y_2, \sigma)) \tag{3.2}$$

and $v_Q((y_1, y_2)) = v_E(y_1)$. The state space of Q, is $W = R(Q, (Y, Y))$. Fig. 5 illustrates the automaton Q for the system in Fig. 3. Note that the number of states of Q is the same as that of O_E. For each state of Q, the second component denotes the set of states that the system can be in, whereas the first component denotes the set of states that the system can be in *if* the trajectory has not gone through E yet.

The following lemma shows that the problem of output prestabilization can be formulated as a problem of prestabilization of Q. The key is to find a state feedback K for Q, which we can then adapt to a corresponding compensator for A, and which forces all trajectories in Q_K to have finite length. This in turn will force corresponding trajectories in A to go through E in a finite number of transitions. In doing this, however, we need to make sure that the compensator for A keeps A alive.

Lemma 3.6: A is output prestabilizable with respect to E while preserving liveness iff there exists a feedback $K: W \to U$ such that for all

$$(y_1, y_2) \in R(Q_K, (Y, Y))$$

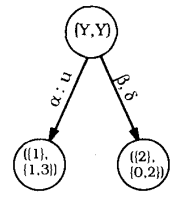


Fig. 5. Example of the automaton Q (all the events are observable).

$K((y_1, y_2))$ is y_2 = compatible, and Q_K is prestable with respect to its dead states, i.e., with respect to the states y such that $v_{Q_K}(y) \neq \emptyset$.

Proof:

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

($\leftarrow$) We claim that the compensator defined by

$$C(s) = K(w_{Q_K}((Y, Y), s))$$

for $s \in L(Q_K, (Y, Y))$ and $C(s) = \Phi$ for all other s, prestabilizes A and we prove this as follows: thanks to the compatibility condition, A_C is alive. Also

$$h(\bar{L}(A_C)) \subset L(Q_K, (Y, Y))\Gamma^*.$$

Given $s \in \bar{L}(A_C)$, if $s \in L(Q_K, (Y, Y))$ then the trajectory may not have passed through E yet. If $s \notin L(Q_K, (Y, Y))$, suppose that $s = p\sigma$ for some $p \in L(Q_K, (Y, Y))$ and $\sigma \in \Gamma$. Since σ is *not* defined at $w_{Q_K}((Y, Y), p)$, σ could have occurred only if the trajectory has already passed through E. Since also all strings in $L(Q_K, (Y, Y))$ are finite and C preserves liveness, A_C is E-prestable. □

In order to construct a compensator as proposed by the above lemma, let us first characterize the states in Q that we can "kill" while preserving liveness in A. In particular, let E_Q be the set of states $y = (y_1, y_2) \in W$ so that we can find a y_2 compatible set of events $F \subset \Phi$ which, if used as a control input at y, disables all events defined from y, i.e.,

$$E_Q = \{y = (y_1, y_2) \in W \mid \exists F \subset \Phi \text{ such that } v_{QF}(y) = \emptyset \text{ and } F \text{ is } y_2\text{-compatible}\} \tag{3.3}$$

where $v_{QF}(y) = (v_Q(y) \cap F) \cup (v_Q(y) \cap \bar{\Phi})$. For example, consider the system in Fig. 6, where Fig. 6(a) illustrates A, (b) illustrates A_E, (c) illustrates the observer O for A, and (d) illustrates the observer O_E for A_E. The automaton Q for this example is illustrated in Fig. 7(a). Note that we can disable β at both of the states (2, 123) and (2, 2) so that no transitions are enabled in Q at these states, but the states 1, 2, and 3 remain alive in A. Thus, $E_Q = \{(2, 123), (2, 2)\}$. Thus, what we have shown is the following.

Proposition 3.7: A is output prestabilizable while preserving liveness iff there exists a state feedback K_0 such that Q_{K_0} is E_Q-prestable and for all $(y_1, y_2) \in W$, $K((y_1, y_2))$ is y_2-compatible in A. Furthermore, the compensator defined by

$$C(s) = K(w_{Q_{K_0}}((Y, Y), s))$$

for $s \in L(Q_K, (Y, Y))$ and $C(s) = \Phi$ for all other s, prestabilizes A, where

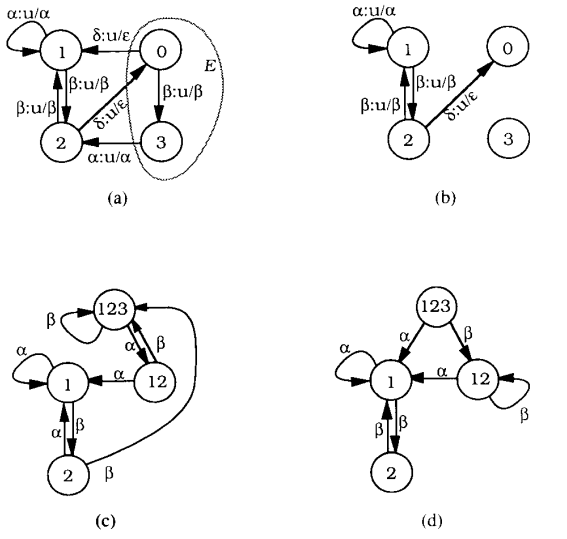


Fig. 6. Output stabilizability example. (a) The system A. (b) A_E. (c) The observer O for A. (d) the observer O_E for A_E.

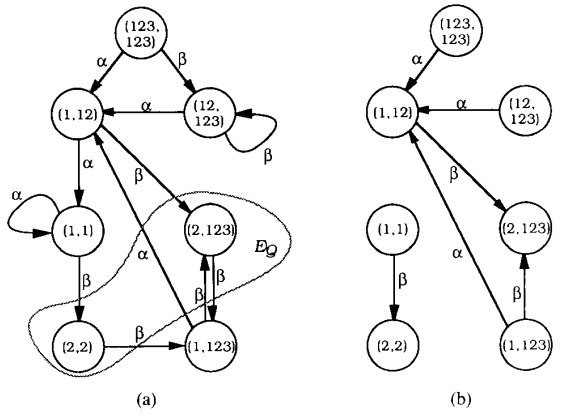


Fig. 7. Output prestabilization of Fig. 6 (recall that α and β are both controllable and observable). (a) Automaton Q. (b) Q_K as computed by algorithm 3.8.

$$K(y = (y_1, y_2)) = \begin{cases} F \subset \Phi \mid v_{QF}(y) = \emptyset \text{ and } F \text{ is } y_2\text{-compatible} \\ \quad \text{if } y \in E_Q \\ K_0(y) \\ \quad \text{otherwise.} \end{cases}$$ □

Appropriately modifying algorithm 3.4 for Q we also have.

Proposition 3.8: The following algorithm is a test for output prestabilizability while preserving liveness.

Algorithm: Let $Z_0 = E_Q$ and for $y = (y_1, y_2) \in E_Q$, let $K(y) = F \subset \Phi$ where F is such that $v_{QF}(y) = \emptyset$ and f is y_2-compatible. Iterate

$$P_{k+1} = \{y \in W \cap \hat{Z}_k \mid \{\gamma \in v_Q(y) \mid w_Q(y, \gamma) \in Z_k\} \text{ contains } v_Q(\hat{x}) \cap \bar{\Phi} \text{ and is } y_2\text{-compatible in } A\}$$

$$K(y) = \{\gamma \in v_Q(y) \mid w_Q(y, \gamma) \in Z_k\} \cap \Phi \text{ for } y \in P_{k+1}$$

$$Z_{k+1} = Z_k \cup P_{k+1}.$$

Terminate when $Z_{k+1} = Z_k = Z^*$. A is output prestabilizable iff $(Y, Y) \in Z^*$. The corresponding feedback is K as computed above. □

Note that if A is observable, the bound on the radius of O implies that A_C goes through E in at most $O(nq^3)$ transitions and that the aforementioned algorithm has complexity at most $O(q^3 \mid W \mid)$. Also, as is the case for strong output stabilizability, it is possible to construct maximally and minimally restrictive prestabilizing compensators.

Fig. 7(b) illustrates the closed-loop system Q_K after this algorithm is applied to Q in Fig. 7(a). In order to construct a compensator that prestabilizes the system in Fig. 6(a), we use the range of (123, 123) in Q_K as follows: initially (i.e., before any observable events are seen so that we are in (123, 123) of Q_K), we disable β. After α is observed (so that the state in Q_K is (1, 12)), α is disabled, while β is enabled, and finally, after β is observed (corresponding to a transition to the state (2, 123)), β is disabled while α is enabled. When α occurs again, we know that all the trajectories have passed through E, and thus we do not care about what the control input is after this point as long as it keeps the system alive.

As discussed previously, we can now construct a stabilizing compensator by restarting our prestabilizing compensator after we are certain that the state has passed through E. We now present an approach which allows us to detect, as soon as possible, that the trajectory has passed through E. Given an output prestabilizable A, suppose that C is the corresponding compensator and K is the corresponding Q-feedback for C. Recall that for Q_K, no event are defined at states $(y_1, y_2) \in E_Q$, and in general, given some $y = (y_1, y_2) \in R(Q_K, (Y, Y))$, not all events defined at y_2 are defined at y. Given an output trajectory of A_C, let us trace the corresponding trajectory in Q_K starting from the state (Y, Y). Suppose that we observe a transition which is *not* defined at the current state of Q_K. By the way we have constructed Q_K we know that the occurrence of such a transition implies that the trajectory has already passed through E. This is precisely the mechanism which we use to detect that the trajectory has passed through E. So, given $s \in h(\bar{L}(A_C) \cup L(Q_K, (Y, Y))$, let $y = w_{Q_K}((Y, Y), s)$ and suppose that the next observation is a transition $\sigma \notin v_{q_K}(y)$, and thus we know that the trajectory has passed through E. At this point, we wish to force the trajectory to pass through E again, but in doing so, we can use our knowledge of the set of states that the system can be in at the time we have detected that the trajectory has passed through E, i.e., $w(y_2, \sigma)$. What we would then like to do is to have q transition to the state $z = (w(y_2, \sigma), w(y_2, \sigma))$. However, as we have defined it so far, z may *not* be in W. What we must do in this case is to augment W with all such z's and any new subsequent states that might be visited starting from such a z and using an extension of the dynamics of Q. Specifically, the dynamics of q given in (3.2) can be defined for arbitrary subsets $y_1, y_2 \subset Y$, as can its restriction w_{Q_K} by feedback. We modify this definition as follows: if $w_{EK}(y_1, \sigma) = \emptyset$, then we set $w_{Q_K}((y_1, y_2), \sigma)$

to $(w(y_2, \sigma), w(y_2, \sigma))$. Let W^a be the union of the reaches of all states of the form (Y', Y') with $Y' \subset Y$ and define $Q^a = (F^a, w, \upsilon)$ where $F^a = (W^a, \Gamma, \Gamma)$. Note that $E_Q \subset W^a$ and $R(Q_K, (Y, Y)) \subset W^a$. If in fact any $z = (Y', Y')$ is prestabilizable with respect to $R(Q_K, (Y, Y))$ in Q^a, then we can force the trajectory to pass through E. It is straightforward to check (by assuming the contrary) that prestabilizability of Q is sufficient for being able to do this.

Proposition 3.9: If there exists a feedback K for Q such that Q_K is E_Q-prestable and $K(y)$ is y_2-compatible, then there exists a feedback K' such that for any $Y' \subset Y$, $z = (Y', Y')$ is prestable with respect to $R(Q_K, (Y, Y))$ in Q_K^a, and $K'(y)$ is y_2-compatible for each $y = (y_1, y_2) \in R(Q_{K'}^a, z)$. □

In order to construct an output stabilizing compensator, we use the aforementioned proposition recursively as follows: let K_0 be a feedback that prestabilizes Q and preserves liveness, as can be constructed using the algorithm in Proposition 3.8. Let Z_0 represent the initial state of Q_{K_0} and let W_0 represent the range of Z_0, i.e., the states we may be in when we know that the trajectory has already passed through E:

$$Z_0 = (Y, Y) \tag{3.3}$$

$$W_0 = R(Q_{K_0}, Z_0). \tag{3.5}$$

We then augment Z_0 to include the states to which we may "reset" our compensator, i.e.,

$$Z_1 = Z_0 \cup \{(\hat{x}, \hat{x}) \mid \hat{x} = w(y_2, \sigma) \text{ for some } y = (y_1, y_2) \in W_0 \text{ and } \sigma \in \hat{\upsilon}(y_2, K_0(y))\} \tag{3.6}$$

where $\hat{\upsilon}(y_2, K_0(y)) = (\upsilon(y_2) \cap K_0(y)) \cup (\upsilon(y_2) \cap \bar{\Phi})$. Next, we find a feedback K_1 that satisfies Proposition 3.9 for each $(Y', Y') \in Z_1$. Note that we can always choose K_1 so that it is an extension of K_0, i.e., $K_1(y) = K_0(y)$ for $y \in R(Q_{K_0}, Z_0)$. Then, we let $W_1 = R(Q_{K_1}, Z_1)$. Proceeding in this fashion, we construct W_2, W_3, etc., and the corresponding extensions of the feedback law, until $W_{k+1} = W_k = W'$ for some k (note that k must necessarily be finite). Let K' be the corresponding feedback, then

- $Q_{K'}$ is E_Q-prestable,
- $K'(y)$ is y_2-compatible for all $y \in W'$, and
- for all $y \in E_Q \cap W'$ and $\sigma \in \hat{\upsilon}(y_2, K'(y))$, $(w(y_2, \sigma), w(y_2, \sigma)) \in W'$.

Finally, we construct an automaton $Q' = (F', w', \upsilon')$ where $F' = (W', \Gamma, \Gamma)$ which includes the transitions to states in Z', i.e.,

$$w'(y, \sigma) = \begin{cases} w_Q(y, \sigma) & \text{if } \sigma \in \upsilon_{Q_{K'}}(y) \\ (w(y_2, \sigma), w(y_2, \sigma)) & \text{otherwise} \end{cases} \tag{3.7}$$

$$\upsilon'(y) = \hat{\upsilon}(y_2, K(y)). \tag{3.8}$$

Then, the compensator defined by

$$C(s) = K'(w'((Y, Y), s)) \tag{3.9}$$

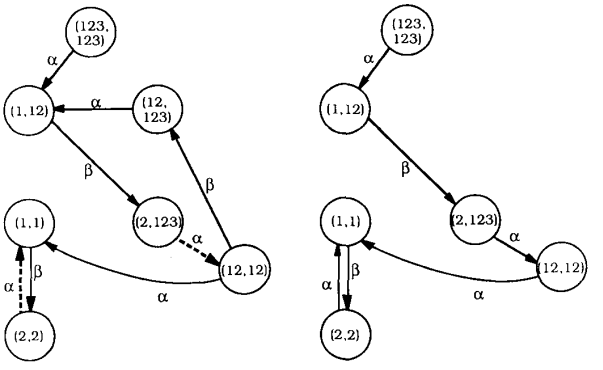


Fig. 8. Output stabilization of Fig. 6 (recall that both α and β are controllable). (a) Adding the new states (through the dashed arcs). (b) Q'.

for all $s \in L(Q', (Y, Y))$ stabilizes A. Thus the compensator consists of the automaton Q', started in (Y, Y) and the feedback $K': W' \to 2^{\Phi}$ so that the desired compensator is given by (3.9). For example, for the system in Fig. 6, we need to prestabilize the state (12, 12) (see Fig. 8(a)). The resulting automaton Q' that produces the desired compensator is shown in Fig. 8(b).

IV. Discussion

In this paper, we have introduced and studied concepts of output stabilization for discrete-event dynamic systems. Key features of our formulations, which distinguish it from many of the problems and approaches considered in the literature are the focus on stability, i.e., the ability of the system to recover from anomalies without catastrophic error propagation, and the event-driven observation model which raises the important question of the coordinated timing of information and control action. The work presented here and in [5] and [6] also provide the basis for our work on controlling DEDS so that particular tasks are completed, where the completion of a task is modeled by the occurrence of one of a specific set of event sequences. Such a problem obviously brings our work much closer to the linguistic framework of [1]–[3], [8]–[10], [12]. However, by using a state framework—and by constructing sets of states E_i corresponding to allowable starting states for completion of tasks i—we can not only design controllers that supervise the completion of tasks, but can in fact address the problem of achieving acceptable transient behavior by making the supervised system stable in the presence of anomalies and errors. Also in [4], we consider the design of systems that accept task sequence commands as inputs and produce the desired control inputs. In this context, it is essential that task completion be detectable and thus the notion of strong output stabilizability developed here plays an important role. Indeed, combining all of these pieces leads not only to a methodology for task-level control but also to a procedure for the hierarchical modeling of DEDS in which strings (corresponding to tasks) at the lower level are modeled as single events at the higher level.

Several additional points deserve some comment. First, as we have remarked, one of the key features of our intermittent

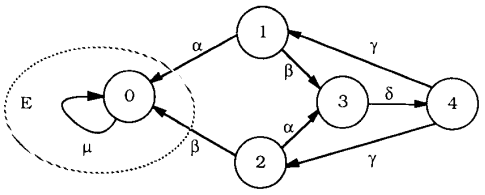


Fig. 9. Stabilizable, observable, but not output stabilizable system (all the events are controllable and observable).

observation model is that it highlights the importance of the coordinated timing of information and control action. Indeed the natural notion of observability for this measurement model, together with stabilizability do *not* imply output stabilizability in our framework. For example, consider the system in Fig. 9, where all the events are controllable and observable. This system is stabilizable by disabling β at state 1 and α at 2, and it is also observable. However, it is *not* output stabilizable, since we can never distinguish between states 1 and 2, and thus we cannot selectively disable α or β, i.e., timing of information and control action are incompatible. The same phenomenon can also be found if we use a slightly more general model including partial and possibly sporadic state information.

In contrast, as discussed in [4], this phenomenon cannot occur if one uses a strong notion of observability representing a slight generalization of the concept used, for example, by Ramadge in [7]. Specifically, if, following an initial transient, the system state is known perfectly after *every* observable transition, then this plus stabilizability imply output stabilizability. More precisely and generally, let K_u denote an observer feedback chosen so that (1) the observer remains alive and we also maximize the size of the set E_u of observer singleton states that is invariant under the dynamics of O_{K_u}; and (2) O is E_u-stable (so that with this feedback the state of the system is guaranteed to be known perfectly and to be in E_u following each observable event after an initial transient). If $E \cap E_u \neq \emptyset$ and if A is $E \cap E_u$-stabilizable, then the system is in fact output stabilizable. Indeed, if K_s is the state feedback that achieves $E \cap E_u$-stability, then the output compensator can be constructed by merging the separately constructed observer and state feedbacks:

$$\hat{K}(\hat{x}) = \begin{cases} K_u(x) \cap K_s(x) & \text{if } \hat{x} = \{x\} \in E_u \\ \Phi & \text{otherwise.} \end{cases} \quad (4.1)$$

We refer the reader to [6] for a general discussion of the computation of controlled-invariant sets and to [4] for details of the preceding development which also introduces a proof that the computational complexity of the testing of these conditions and the construction of the feedback (4.1) is guaranteed to be at most $O(q^4)$. This is a nontrivial point since, as made clear by Tsitsiklis [11], the construction of controllers based on partial information in general has exponential complexity. As we have seen, the complexity of the procedures we have presented in Section III are (under the assumption of observability) of polynomial order in q times the cardinality of the observer state space. As discussed in [5], the observer state space for an observable system can in fact be exponential in q. While for many systems the observer is in fact much smaller than this, It is important to investigate conditions under which polynomial complexity can be guaranteed. In [5] we provide a tighter bound on observer complexity that can often be useful, and the construction described in the preceding paragraph, for systems in which state observability can be maintained after *every* observable transition, represents another potentially useful special case. Similarly in [4], we describe another such case which involves the concept of *always observable states* [5]. Specifically, a state x is always observable if *whenever* the system is in x, the observer estimate $\{x\}$. If the observer is stable with respect to always observable states and if A is E-stabilizable when we only allow control action when we are in always observable states, then we can clearly design a stabilizing output compensator since we will know exactly when we are in such a state and, thanks to stability with respect to always observable states, this will happen regularly. We refer the reader to [5], [4] for details and for a proof testing these conditions and constructing the required compensator having $O(q^4)$ complexity.

Finally, as in [5], we can address the problem of designing output compensators that are *resilient* in that they maintain system liveness and stability in the presence of a burst of observation errors, where such a burst can include missed detections of observable events, incorrect insertion of extraneous detections of such events, and erroneous identification of events. In such cases, our compensators must be based on the resilient observer of [5], discussed in Section II. The more major impact on our design of compensators is caused by the fact that errors might cause the system and the observer to be in arbitrary and unrelated states. Thus, in order to guarantee liveness of the closed-loop system, we must use X-compatible feedback. Specifically, it is not difficult to verify [4] the following variation of the result in Section III-A.

Proposition 4.1: An observable system A is resiliently strongly output stabilizable with respect to E iff there exists a state feedback K for the observer such that O_K is E_O-stable and for all $\hat{x} \in Z$, $K(\hat{x})$ is X-compatible. □

An algorithm for testing resilient strong output stabilizability and constructing a feedback is identical to Algorithm 3.4 except that when we search for a feedback, we search for one that is X-compatible, as opposed to $\hat{x}$-compatible, and the computational complexity is again $O(q^3 \mid Z \mid)$. Thus, if we can find K that satisfies Proposition 4.1, then $C(s) = K(w_{KR}(\{Y,\}, s))$ is a resiliently strongly stabilizing compensator for A, where w_{KR} denotes the closed-loop dynamics of O_R using feedback K.

Similarly, necessary and sufficient conditions for resilient output stabilizability parallel those of output stabilizability except that we need to use X-compatible feedback. In this case, we need to use a resilient version of the automaton Q defined in Section III-B. Specifically, for any feedback K defined on Q_K, we define $Q_{KR} = (G_{KR}, w_{KR}, \upsilon_{KR})$ so that

$$w_{KR}(y, \gamma) = \begin{cases} w_{Q_K}(y, \gamma) & \text{if } \gamma \in \upsilon_{Q_K}(y) \\ (Y, Y) & \text{otherwise} \end{cases} \quad (4.2)$$

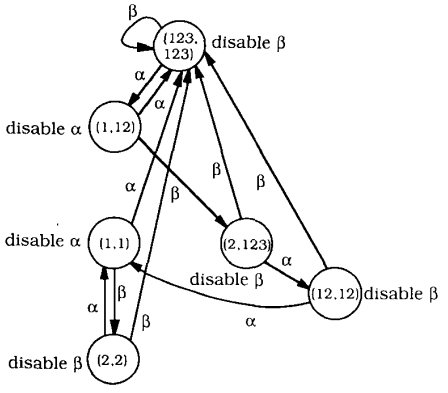


Fig. 10. Resilient output stabilizing compensator for Fig. 6.

$$v_{KR} = \Gamma. \tag{4.3}$$

We state the following companion of Proposition 3.7 where

$$E_{QR} = \{y = (y_1, y_2) \in W \mid \exists F \subset \Phi \text{ such that } v_{QF}(y) = \emptyset \text{ and } F \text{ is } X\text{-compatible}\}. \tag{4.4}$$

Proposition 4.2: An observable system A is resiliently output stabilizable iff there exists a state feedback K such that Q_K is E_Q-prestable and for all $y \in W$, $K(y)$ is X-compatible in A. Furthermore, the compensator defined by $C(s) = K(w_{KR}((Y, Y), s))$ for all $s \in \Gamma^*$ resiliently stabilizes A. □

An algorithm for testing resilient output stabilizability and constructing a feedback can be generated from Algorithm 3.8 in a straightforward fashion. In particular, we use E_{QR} in place of E_Q in Algorithm 3.8 and we check X-compatibility, instead of y_2-compatibility.

For example, the feedback we computed for Q in order to stabilize the system in Fig. 6 is also X-compatible (see Fig. 8(b)), since, in this case, disabling either, but only one of, α or β does not disable *all* the events in any state of the system. A resilient output stabilizing compensator for the system in Fig. 6 is illustrated in Fig. 10 for which the initial state is (123, 123).

References

[1] H. Cho and S. I. Marcus, "On the supremal languages of sublanguages that arise in supervisor synthesis problems with partial observations," *MCSS*, vol. 2, no. 2, pp. 47–69, 1989.

[2] R. Cieslak, C. Desclaux, A. Fawaz, and P. Varaiya, "Supervisory control of discrete-event processes with partial observations," *IEEE Trans. Automat. Contr.*, vol. 33, no. 3, pp. 249–260, Mar. 1988.

[3] F. Lin and W. M. Wonham, "On observabililty of discrete event systems," *Informat. Sci.*, vol. 44, pp. 173–198, 1988.

[4] C. M. Özveren, "Analysis and control of discrete event dynamic systems: A state space approach," Ph.D. thesis, M.I.T., Cambridge, MA, Aug. 1989; also in Laboratory for Information and Decision Systems, M.I.T., Cambridge, MA, Rep. LIDS-TH-1907, Aug. 1989.

[5] C. M. Özveren and A. S. Willsky, "Observability of discrete event dynamic systems," *IEEE Trans. Automat. Contr.*, vol. 35, no. 7, pp. 797–806, July 1990.

[6] C. M. Özveren, A. S. Willsky, and P. J. Antsaklis, "Stability and stabilizability of discrete event dynamic systems," Laboratory for Information and Decision Systems, M.I.T., Cambridge, MA, Rep. LIDS-P-1853, Feb. 1989; also in *J. ACM* (to be published).

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

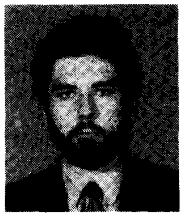
[8] ——, "Some tractable supervisory control problems for discrete event systems modeled by buchi automata," *IEEE Trans. Automat. Contr.*, vol. 36, pp. 10–19, Jan. 1989.

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

[10] —— "Supervisory control of a class of discrete event processes," *SIAM J. Contr. Optimiz.*, vol. 25, no. 1, Jan. 1987.

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

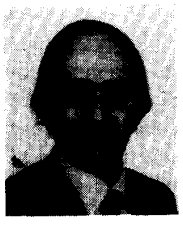
[12] A. F. Vax and W. M. Wonham, "On supervisor reduction in discrete event systems," *Int. J. Contr.*, 1986.



Cüneyt M. Özveren (S'82–M'84–S'84–M'89) 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 Principal 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, Cambridge, MA, 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 Institute 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 Nobel Prize by the ASCE and the 1980 Browder J. Thompson Memorial Prize Award by the IEEE for a paper excerpted from his monograph.