

Stabilization, Tracking, and Hierarchical Modeling of Discrete-Event Dynamic Systems¹

C.M. Özveren
A.S. Willsky

Laboratory for Information and Decision Systems, MIT, Cambridge, MA 02139 USA

Abstract. This paper is concerned with the development of a regulator theory for discrete-event dynamic systems (DEDS), which we model using finite state automata with partially controllable and observable events. We describe a notion of stability that focuses on the ability of the system to recover from errors. This leads to a theory of feedback stabilization. In addition, we develop a theory of observability and observer design for DEDS in which only certain key events are observed. Our results on stability and observers lead to a theory of stabilization via dynamic output feedback. We also briefly describe results on tracking capabilities of DEDS, i.e. on the ability of a DEDS to follow prescribed event trajectories. This involves the introduction of the notion of *resiliency*, i.e. of the ability of the DEDS to resume correct tracking after an error. Finally, a crucial issue in the analysis of DEDS is computational complexity. We describe some aspects of our work on this and in particular describe a theory of hierarchical modeling in which sequences of events at a low level are mapped into a single event (a "task") at a higher level.

Keywords. Discrete systems; control theory; automata theory; hierarchical systems; mathematical system theory; stability; supervisory control; automation; observers; intelligent machines.

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, by Wonham, Ramadge, et al. (see, for example, Ramadge and Wonham (1987a,b)), and this work has prompted a considerable response by other researchers in the field. 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 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. The development of such a theory requires several ingredients which we have developed and which are described here. In the next section we provide some background and in particular describe a concept of stability for DEDS and the related notion of stabilizability. We also discuss the problem of observability and observer design, problems made somewhat more complex by the intermittent nature of the observations (caused by the fact that only some of the events are observed). Section 3 then describes results on output stabilizability, synthesizing results from the preceding sections. The observation structure again complicates the problem; for example observability and stabilizability do not imply output stabilizability. Finally we conclude in Section 4 with discussion of additional issues and results. In particular we discuss the problem of tracking specified event sequences (an essential element in the development of a true regulator theory) and of resiliency, i.e.,

the ability of DEDS to recover from errors. In addition a critical issue in DEDS theory is that of computational complexity, and we also describe some of our work in this area. Perhaps the most significant is the development of the notion of *tasks* which leads to higher-level, aggregated models of DEDS that captures task-level dynamics and allows us naturally to consider higher-level control problems without worrying about fine-scale DEDS evolution.

PRELIMINARIES

System Model

The class of systems we consider are nondeterministic finite-state automata with intermittent event observations defined over $G = (X, \Sigma, \Gamma, U)$, 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 is the set of admissible control inputs consisting of a specified collection of subsets of Σ , corresponding to the choices of sets of controllable events that can be enabled. The dynamics defined on G are:

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

$$\sigma[k+1] \in (d(x[k]) \cap u[k]) \cup e(x[k]) \quad (2)$$

Here, $x[k] \in X$ is the state after the k th event, $\sigma[k] \in \Sigma$ is the k th event, and $u[k] \in U$ is the control input after the k th 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), $e: X \rightarrow 2^\Sigma$ is a set valued function that specifies the set of events that cannot be disabled at each state, and the function $f: X \times \Sigma \rightarrow X$ is also set-valued, so that the state following a particular event is not necessarily known with certainty. Without loss of generality, we assume that $e(x) \subset d(x)$ for all x . The set $d(x)$ represents an "upper bound" on the set of events that can occur at state x , whereas the set $e(x)$, 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 $d(x) \cap e(x)$. While it is possible to consider a more general setting, for simplicity we assume the slightly more restrictive framework of Ramadge and Wonham (1987b) in which there is an event subset $\Phi \subset \Sigma$ such that we have complete control over events in Φ and no control over

¹Research 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. The research of the authors was partially done during their stay at Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA), Rennes, France, and the second author was also supported by IRISA during this time.

events in $\bar{\Phi}$, the complement of Φ . In this case, we can take $U = 2^*$ and

$$e(x) = d(x[k]) \cap \bar{\Phi} \quad (3)$$

Furthermore, we assume that $\Phi \subset \Gamma$.

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

$$h(\sigma) = \begin{cases} \sigma & \text{if } \sigma \in \Gamma \\ \epsilon & \text{otherwise} \end{cases} \quad (4)$$

Then, our output equation is

$$\gamma[k+1] = h(\sigma[k+1]) \quad (5)$$

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, $h(\sigma_1 \dots \sigma_n) = h(\sigma_1) \dots h(\sigma_n)$.

An important notion is that of liveness: A is alive if it cannot reach a point at which no event is possible. 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.

Stability and Stabilizability

Definition 1 Let E be a specified subset of X . A state $x \in X$ is E -pre-stable 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 -pre-stable. The DEDS is E -stable (respectively, E -pre-stable) if every $x \in X$ is E -stable (respectively, E -pre-stable). \square

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, i.e., $i \leq n$. A cycle, is a finite sequence of states x_1, x_2, \dots, x_k , with $x_k = x_1$, so that there exists an event sequence s that permits the system to follow this sequence of states.

Definition 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 . \square

An upper bound on both the radius and the E -radius, for any E , of an E -stable system is n . Also, as shown in (Özveren, et al. 1989), a necessary and sufficient condition for E -stability of A is the absence of cycles that do not pass through E . We refer the reader to (Özveren, et al. 1989) 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 Definition 1 requires 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 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 (Özveren, et al. 1989) for stability can be used without change to test for weak stability.

A state feedback law is a map $K : X \rightarrow U$ and the resulting closed-loop system is $A_K = (G, f, d_K, h)$ where

$$d_K(x) = (d(x) \cap K(x)) \cup (d(x) \cap \bar{\Phi}) \quad (6)$$

Definition 3 A state $x \in X$ is E -pre-stabilizable (respectively, E -stabilizable) if there exists a state feedback K such that x is E -pre-stable (respectively, E -stable) in A_K . The DEDS is E -stabilizable if every $x \in X$ is E -stabilizable. \square

If A is E -stabilizable, then, there exists a state feedback K such that every $x \in X$ is E -stable in A_K . We refer the reader to (Özveren, et al. 1989) 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.

Observability and Observers

A system is observable if the current state is known perfectly at intermittent but not necessarily fixed intervals of time. Obviously, a necessary condition for observability is that it is not possible for our DEDS to generate arbitrarily long sequences of unobservable events, i.e., events in $\bar{\Gamma}$, the complement of Γ . This is not difficult to check and will be assumed.

We now introduce some notation. Let $R(A, x)$ denote the set of states reachable from x . Also, let Y denote the set of states such that either y has no transitions defined to it or there exists an observable transition from some state to y . 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 . 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 $\bar{L}(A) = \bigcup_{x \in X} L(A, x)$ 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 (Özveren and Willsky, 1989) we present a straightforward design of an observer which is a DEDS 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 of the observer is a subset Z of 2^Y , and the events and observable events are both Γ . If the present observer estimate is $\hat{x}[k] \in Z$ and the next observed event 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 $\gamma[k+1]$. This description can be translated directly into the dynamics of a DEDS, O . From the definition of observability, it is immediate that A is observable iff O stable with respect to its singleton states. We can also show that if A is observable then all trajectories from an observer state pass through a singleton state in at most q^2 transitions. Since also there can be at most q singleton states, the radius of the observer is at most q^3 . This will play an important role in determining the maximum number of transitions it takes a trajectory from a state, in an output stabilizable system, to pass through E .

OUTPUT COMPENSATORS AND OUTPUT STABILIZATION

An output compensator C is a dynamic system that transforms a string of observed output events into the control u to be applied until the occurrences of the next output event. Thus $C : \Gamma^* \rightarrow U$, although we actually need only define it for feasible output strings, i.e., for those in $h(I(A))$. When C is applied to a DEDS, A , the resulting closed loop system is denoted by A_c .

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 $C(s)$:

Definition 4 Given $Q \subset X$, $F \subset \Phi$, F is Q -compatible if for all $x \in R(A|F, Q)$, $(d(x) \cap F) \cup (d(x) \cap \bar{\Phi}) \neq \emptyset$. A compensator C is A -compatible if for all $s \in h(\bar{L}(A))$, $C(s)$ is $\hat{x}(s)$ -compatible. \square

As we will see, we can often restrict ourselves to compensators of the form $C(s) = K(\hat{x}(s))$ where $\hat{x}(s)$ is the state of O after $s \in \Gamma^*$ has been observed and $K : Z \rightarrow U$ is a memoryless function. In this case we call C O -compatible and K the observer feedback map.

In this section, we present and analyze two notions of output stabilizability (where, for simplicity we restrict attention to ob-

servable systems). 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 :

Definition 5 A is strongly output stabilizable if there exists a compensator C and an integer i such that A_C is alive and for all $p \in L(A_C)$ such that $|p| \geq i$, there exists a prefix t of p such that $|p/t| \leq i$ and $\mathcal{R}(h(t)) \subset E$. We term such a compensator a strongly output stabilizing compensator. \square

What this definition states is that in addition to keeping the system alive, the compensator C also forces the observer to a state corresponding to a subset of E at intervals of at most i observable transitions. The next result shows that we can restrict attention to observer feedback:

Proposition 1 A is strongly output stabilizable if there exists a state feedback $K: Z \rightarrow U$ for the observer such that X_I in $A \parallel O_K$ is E_{OC} -stable, where $X_I = \{(x, \{Y\}) | x \in X\}$ is the set of possible initial states in $A \parallel O_K$ and where $E_{OC} = \{(x, \hat{x}) \in Y \times Z | \hat{x} \subset E\}$ is the set of composite states for which the system is in E and we know that the current state is in E .

Proof: See (Ozveren 1989). \square

An important point is that since O captures all of the behavior that can be generated by A and since we wish to know why we are in E , we can limit ourselves to the stabilization of O by state feedback, while paying attention to keeping A alive:

Proposition 2 A is strongly output stabilizable iff there exists a state feedback $K: Z \rightarrow U$ for the observer such that O_K is stable with respect to $E_O = \{\hat{x} \in Z | \hat{x} \subset E\}$ and for all $\hat{x} \in Z$, $K(\hat{x})$ is \hat{x} -compatible. Furthermore, if A is strongly output stabilizable then the trajectories in the reach of X_I in $A \parallel O_K$ go through E_{OC} in at most nq^3 transitions.

Proof: A straightforward consequence of Proposition 1 and the fact that the radius of O is at most q^3 . \square

Proposition 3 The following algorithm is a test for strong output stabilizability. It has complexity $O(q^3|Z|)$:

Algorithm Let $Z_0 = E_O$ and iterate:

$$\begin{aligned} P_{k+1} &= \{\hat{x} \in Z | \{\gamma \in v(\hat{x}) | w(\hat{x}, \gamma) \in P_k\} \text{ is } \hat{x}\text{-compatible}\} \\ K(\hat{x}) &= \{\gamma \in v(\hat{x}) | w(\hat{x}, \gamma) \in P_k\} \text{ for } \hat{x} \in P_{k+1} \\ Z_{k+1} &= Z_k \cup P_{k+1} \end{aligned}$$

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

Proof: The proof is straightforward and based on the proof of the algorithm for testing pre-stabilizability in (Ozveren, et al. 1989). Computational complexity follows from the fact that the observer has $|Z|$ states and the algorithm terminates in at most q^3 steps. \square

We now turn to the following somewhat weaker notion:

Definition 6 A is output stabilizable (respectively, output pre-stabilizable) with respect to E if there exists a compensator C such that A_C is E -stable (respectively, E -pre-stable). We term such a compensator an output stabilizing (respectively, output pre-stabilizing) compensator. \square

Note that this definition implicitly assumes that there exists an integer i such that the trajectories in A_C go through E in at most i transitions. Using this bound, we can show that output pre-stabilizability and liveness are necessary and sufficient for output stabilizability, as is the case for stabilizability and pre-stabilizability (see (Ozveren, et al. 1989)):

Proposition 4 A is output stabilizable iff A is output pre-stabilizable while preserving liveness (i.e., the closed loop system is pre-stable and alive). \square

Thanks to this result, we only need to design a pre-stabilizing compensator. Thus, all we are interested in accomplishing is in ensuring that the state passes through E : if a trajectory has already passed through E , it is of no further concern to us. That is, what we need to do is to keep track of the state in which the system can be if the trajectory has not yet passed through E . If we can force this set to be empty (while preserving liveness in A), we will be certain that the state has passed through E .

The following construction allows us to perform this function: Delete all events in A that originate from the states in E (since we are unconcerned with behavior after the system has entered 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. The observer O_E captures all the behavior of A until its trajectories enter E . When we look at the states of O_E , we see that there are some "trapping" states, each of which is a subset of E . 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 $\hat{x} \cap E$ 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 pre-stabilizing 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

$$Q = (F_Q, w_Q, v_Q) = O_E \parallel O \quad (7)$$

together with the initial state (Y, Y) keeps track of all the information we need for designing an output stabilizing compensator. Let $W = R(Q, (Y, Y))$ denote the state space of Q .

The following result captures the intuitive concept presented previously: we wish to drive the system so that it is impossible for the state to have avoided E , while making sure that our compensation keeps A alive:

Lemma 1 A is output pre-stabilizable with respect to E while preserving liveness iff there exists a feedback $K: W \rightarrow U$ such that for all

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

$K((y_1, y_2))$ is y_2 -compatible, and Q_K is pre-stable with respect to its dead states, i.e., with respect to the states y from which no transitions are defined. \square

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 $F(y) \subset \Phi$ which, if used as a control input at y , disables all events defined from y . That is, restriction to the set of events F allows transitions in O by itself (and thus preserves liveness in A), but kills O_E , so that it is impossible for the state in A to have avoided E . The following result shows that we can restrict attention to state feedback on Q , i.e., that we need only consider compensation consisting of the dynamics of Q followed by a memoryless feedback map:

Proposition 5 A is output pre-stabilizable while preserving liveness iff there exists a state feedback K_0 such that Q_{K_0} is E_Q -pre-stable and for all $(y_1, y_2) \in W$, $K((y_1, y_2))$ is y_2 -compatible in A . Furthermore, let $w_{Q_{K_0}}$ denote the state transition map for Q with the feedback K_0 . Then 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 , pre-stabilizes A , where

$$K(y = (y_1, y_2)) = \begin{cases} F(y) & \text{if } y \in E_Q \\ K_0(y) & \text{if } y \notin E_Q \end{cases}$$

where $F(y)$ was defined previously for $y \in E_Q$. Finally, the trajectories in A_C go through E in at most nq^3 transitions.

Proof: Straightforward using Lemma 1 and the fact that the radius of the observer is at most q^3 . \square

We now present an algorithm to test for output pre-stabilizability and to construct the corresponding feedback by appropriately modifying Algorithm 3 for Q :

Proposition 6 The following algorithm is a test for output pre-stabilizability while preserving liveness. It has complexity $O(q^3|W|)$:

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:

$$P_{k+1} = \{y \in W \mid \{\gamma \in v_Q(y) \mid w_Q(y, \gamma) \in P_k\} \text{ is } y_2\text{-compatible in } A\}$$

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

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

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

The preceding algorithm produces a pre-stabilizing output compensator. From Proposition 4 we know how to convert this to a stabilizing compensator: when we are certain that the trajectory has passed through E , we can force the trajectory to go through E again by starting the compensator over, i.e., by ignoring all the observations to date and using the pre-stabilizing compensator on the new observations. In the proof of Proposition 4, we computed an integer j^* so that all the trajectories are guaranteed to go through E in at most j^* transitions independently of the initial state of the system, so that we can "reset" the output pre-stabilizing compensator after every set of j^* transitions. Waiting j^* transitions is obviously a conservative design since in some cases we may know that A has passed through E at an earlier point. We refer the reader to (Özveren 1989) for the details of an approach that detects passage through E as quickly as possible.

FURTHER TOPICS

There are several other important notions that are needed in order to develop an effective regulatory theory for DEDS. One of these is the notion of tracking. Let $\Xi \subset \Sigma$ be the set of events we wish to track and, as in the definition of h , let $t: \Sigma^* \rightarrow \Xi^*$ be the projection of strings over Σ into Ξ^* . We also assume that the set of controllable events Φ is contained in Ξ , and we now consider compensators as maps $C: X \times \Sigma^* \rightarrow 2^\Phi$ which depend upon the current state and the event trajectory. A string $s \in \Xi^*$ is trackable from $x \in X$ if we can find a compensator C such that A_C is alive and such that the set of strings is the first set of events in Ξ produced by the system, i.e., $t(L(A_C, x)) \subset s\Xi^*$. A language L over Ξ is any subset of Ξ^* . A string $s \in L$ has an infinite extension in L if for any $i \geq |s|$ we can find $r \in L$ so that $|r| = i$ and s is a prefix of r . A language L is prefix closed if for any i and s with $|s| > i$, the prefix p of s with $|p| = i$ is also in L . L is complete if each string in L has an infinite extension and if L is prefix closed. We now have:

Definition 7 Given $x \in X$, a complete language L over the alphabet Ξ is trackable from x if each string in L is trackable from x . \square

Note that the class of trackable languages is closed under arbitrary unions and let $L_T(A, x)$ denote the supremal language trackable from x . In (Özveren 1989) we develop the machinery needed to compute $L_T(A, x)$ and the compensation needed to track any string in this language. The basic ideas are as follows: We first construct an automaton $A^t = (G^t, f^t, d^t, e^t)$ over the set of states Y^t that either have no transitions defined to them or have at least one transition from Ξ defined to them. This automaton keeps track of the state in A after the occurrence of events in Ξ (and thus its transitions must also capture any possible A -transitions resulting from events in Ξ). Note that if $|e^t(x)| = 1$, the event in $e^t(x)$ may always occur and thus is the only trackable event. Consequently, we must apply a first feedback $K^t(x)$ that disables all controllable events if there are any uncontrollable events defined at x . Further, if $|e^t(x)| = 1$ for $x \in Y^t$, then x is a state to be avoided, since

there are at least two possible events in Ξ that can occur, and we cannot force A to track any specific one of them. Thus our feedback must keep A^t out of

$$D_T = \{x \in Y^t \mid |e^t(x)| \geq 2\} \quad (8)$$

The construction of such a feedback requires the development of the concept of f -invariance and (f, u) -invariance, coupled here with the fact that we must make sure that the system remains alive. That is, let V be the maximal sustainable (f, u) -invariant subset of $\overline{D_T}$ in A_K^t , i.e., V is the largest set so that we can find a feedback such that the dynamics stay in V if they begin there and A remains alive. Also, let K_V be the associated minimally restrictive feedback (i.e., it disables as few events as possible), and let $K(x) = K_V(x) \cap K^t(x)$. We then have

Proposition 7 If A_K^t restricted to V is deterministic, then for all $x \in V$, $L_T(A, x) = L(A_K^t, x)$. Furthermore, for all $x \in Y^t \cap \overline{V}$, $L_T(A, x) = \emptyset$. \square

We refer the reader to (Özveren 1989) for the characterization of $L_T(A, x)$ for $x \in Y^t$ and for the variation on this construction if A_K^t is non-deterministic (as it may very well be). Also, in (Özveren 1989) we consider several related notions including: the concept of restrictability, i.e., of designing a compensator to keep the event trajectory in a specified language, which makes contact with the notions of controllable language of Ramadge and Wonham; and the concept of invertibility, i.e., of reconstructing the full event trajectory given observation of the output event sequence, which is, roughly, the dual of tracking.

Another extremely important concept throughout our work is that of resiliency or reliability, that is the ability of the system to recover from errors. In the context of observability this corresponds to the ability of the observer to recover from the occurrence of a burst of erroneous output measurements (corresponding to incorrectly interpreted outputs, missed events, or falsely detected events), where "recover" is defined to mean that the state x is actually in the set of states \hat{x} indicated by the observer. The observer as we have defined in Section 2 is only defined for output strings that could actually occur in the automaton A . When an event occurs, we may observe an infeasible output event sequence, in which case we reset the observer state to Y . It can be shown that the resulting observer O_R is resilient if A is observable.

In the context of output stabilizability we have two notions of resiliency, for strong output stabilizability and for output stabilizability. In the former, after an error burst we wish to drive the state x to E and the observer O_R to a state \hat{x} such that $x \in \hat{x} \subset E$. We must be a bit more conservative concerning keeping the system alive in this case since immediately after an error burst x and \hat{x} may have no relation to one another. The following result shows that, as we have seen before, we can analyze this problem by looking at the corresponding state feedback problem for the observer

Proposition 8 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_Q -stable and for all $\hat{x} \in Z$, $K(\hat{x})$ is X -compatible. \square

An algorithm for testing resilient, strong output stabilizability and constructing a feedback is identical to Algorithm 3 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 $O(q^3|Z|)$. Thus, if we can find K that satisfies Proposition 8, then $C(s) = K(w_{KR}(\{Y, \}, s))$ is a resiliently, strongly stabilizing compensator for A .

Resilient output stabilizability is defined similarly: the existence of a compensator that drives x through E infinitely often in the presence of a finite error burst. In this case we must modify the automaton Q by resetting its state to (Y, Y) if an infeasible output event sequence is observed, and we must again make sure that our feedback is X -compatible:

Proposition 9 A is resiliently output stabilizable iff there exists a state feedback K such that Q_K is E_{QR} -pre-stable and for all $y \in W$, $K(y)$ is X -compatible in A . Here E_{QR} is set of states $y =$

$(y_1, y_2) \in W$ so that we can find an X -compatible set $F(y) \subset \Phi$ which, if used as control input at y , disables all events defined at y . Furthermore, the compensator defined by

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

for all $s \in \Gamma^*$ resiliently stabilizes A , where w_{KR} is the state transition map of Q_K , with reset to (Y, Y) when infeasible events occur. \square

The concept of resiliency is also of great importance in the context of restrictability and invertibility, where the possibility exists for *catastrophic error propagation*, i.e., the occurrence of a finite number of measurement errors leading to an infinite string of system errors. We refer the reader to (Özveren 1989) for the analysis of this phenomenon and conditions under which it cannot occur.

Finally, there is the issue of computational complexity. If one examines the complexity bounds we have given for various problems we see that the bounds are generally polynomial in the size of the state space of the system or the observer. However, as discussed in (Özveren and Willsky 1989), the state space of the observer can in fact be exponential in the size of the system state space. This is not a problem for observability since it is not necessary to perform an explicit state space enumeration to test for observability or to design the observer dynamics. However, such an enumeration is required for output stabilization. Thus it is of interest to determine realistic conditions under which the observer state space is of more moderate size. Such conditions are described in (Özveren and Willsky 1989), while in (Özveren 1989) we describe additional conditions under which output stabilizability tests are guaranteed to have polynomial complexity.

Another approach to reducing complexity involves the development of hierarchical models of DEDS in which the size of the state space, as one moves up the hierarchy, decreases rapidly. Such a structure is also quite natural in many applications in which complex procedures can be thought of as sequences of tasks which in turn can be thought of as sequences of events. In (Özveren 1989) we develop a multi-level control structure

for DEDS precisely along these lines. A task is defined as a set of finite event sequences, where the completion of a task corresponds to the occurrence of one of the sequences in this set. We develop a method for constructing a compensator which causes the system to perform a specified task (note the similarity to the notion of tracking). If we then consider a set of possible tasks and a compensator for each, we can construct a higher-level controller which can enable any one of the compensators and thus cause the lower level system to perform the corresponding task. This controller can be thought of as controlling an aggregated DEDS in which the states and events correspond to "performing Task i ", "completion of Task i ", etc. Note that an entire event sequence in the original automaton is translated into a single event – "completion of Task i " – in the higher level, so that this aggregation in effect involves looking at the system at a longer time scale.

CONCLUSIONS

In this paper we have described many of the elements needed for a regulator theory for DEDS. We have focused most of our attention on the problem of output stabilizability but we have also presented a brief description of such topics as tracking, resiliency, and multi-level aggregation and control of DEDS.

REFERENCES

- Özveren, C.M. (1989). Analysis and control of discrete event dynamic systems: A state space approach. *PhD Thesis*. Massachusetts Institute of Technology, Cambridge, Massachusetts.
- Özveren, C.M. and A.S. Willsky (1989). Observability of discrete event dynamic systems. Submitted to the *IEEE Transactions on Automatic control*.
- Özveren, C.M., A.S. Willsky and P.J. Antsaklis (1989). Stability and stabilizability of discrete event dynamic systems. Submitted to the *Journal of the ACM*.

Ramadge P.J. and W.M. Wonham (1987a). Modular feedback logic for discrete event systems. *SIAM J. of Cont. and Opt.*

Ramadge P.J. and W.M. Wonham (1987b). Supervisory control of a class of discrete event processes. *SIAM J. of Cont. and Opt.*