Discrete Event Dyn Syst (2007) 17:531–583
DOI 10.1007/s10626-007-0027-y

Active Acquisition of Information for Diagnosis and Supervisory Control of Discrete Event Systems

David Thorsley · Demosthenis Teneketzis

Received: 17 October 2006 / Accepted: 7 August 2007 /
Published online: 11 September 2007

Abstract This paper considers the problems of fault diagnosis and supervisory control in discrete event systems through the context of a new observation paradigm. For events that are considered observable, a cost is incurred each time a sensor is activated in an attempt to make an event observation. In such a situation the best strategy is to perform an "active acquisition" of information, i.e. to choose which sensors need to be activated based on the information state generated from the previous readings of the system. Depending on the sample path executed by the system, different sensors may be turned on or off at different stages of the process. We consider the active acquisition of information problem for both logical and stochastic discrete event systems. We consider three classes of increasing complexity: firstly, for acyclic systems where events are synchronized to clock ticks; secondly, for acyclic untimed systems; and lastly, for general cyclic automata. For each of these cases we define a notion of information state for the problem, determine conditions for the existence of an optimal policy, and construct a dynamic program to find an optimal policy where one exists. For large systems, a limited lookahead algorithm for computational savings is proposed.

Keywords Failure diagnosis · Supervisory control · Measurement scheduling · Sensor optimization

D. Thorsley (✉)
Department of Electrical Engineering, University of Washington,
Paul Allen Center - Room AE100R, Campus Box 352500,
Seattle, WA 98195, USA
e-mail: thorsley@ee.washington.edu

D. Teneketzis
Department of EECS, University of Michigan,
1301 Beal Avenue, Ann Arbor, MI 48109, USA
e-mail: teneketzis@eecs.umich.edu

1 Introduction

Many types of systems, including communication networks (Pencolé 2000; Rozé and Cordier 1998), manufacturing processes (Holloway and Chand 1994), and queueing systems, can be modeled using discrete event systems (DES). An important problem in complex systems modelled by DES is the problem of detecting and isolating failure events.

One approach to the problem of failure detection in DES involves verification of the property of diagnosability (for an overview of this approach, see Lafortune et al. 2001). Roughly speaking, a DES is diagnosable if any failure that occurs can be diagnosed after a finite delay. In recent years, there has been interest in studying diagnosability of stochastic DES as well (Lunze and Schröder 2001; Thorsley and Teneketzis 2005).

A problem related to the verification of the diagnosability property is the sensor selection problem for DES (Debouk et al. 2002; Jiang et al. 2003; Yoo and Lafortune 2002a). In the sensor selection problem, the objective is to find the minimal sets of sensors under which diagnosability is preserved when these sensors are activated for the duration of the discrete-event process.

In some situations, finding a solution to a sensor selection problem may not result in a solution that is optimal in a practical sense. For example, in communication networks, the act of sensing an event at a remote location involves using system bandwidth to send the data to a network co-ordinator. If the sensor is wireless, the act of transmitting data involves using some of the small amount of energy available to the sensor. In these situations, we do not purchase a sensor at the start of the process and let it run for the duration; instead we incur a small cost each time the sensor is used.

If our objective is to minimize the total cost incurred by the active use of sensors, then, roughly speaking, our objective is to use the sensors as infrequently as possible, that is, to determine when it is necessary to actively acquire information along each possible system behavior. This is a different objective than that of the standard sensor selection problem, where the goal is to use as few sensors as possible, but to activate them for the duration of the process.

This paper investigates the use of active acquisition of information in the context of DES. Our objective is to minimize the cost of observing a finite-state machine when a cost is paid each time a sensor is activated, while preserving a diagnosability property similar to that of Sampath et al. (1995).

The distinguishing characteristic between the verification problems, the sensor selection problems, and the active acquisition problem proposed in this paper is the information structure. In verification problems such as Sampath et al. (1995), the information available to the observer/diagnoser is specified by a fixed projection or observation mask. In sensor selection problems (Debouk et al. 2002; Ding et al. 2002; Holloway and Chand 1994; Jiang et al. 2001, 2003; Khanna 1973; Kumar and Varaiya 1986; Kushner 1964, 1971; Lafortune et al. 2001; Lunze and Schröder 2001; Meier III et al. 1967; Pencolé 2000; Pollard 2002; Rago et al. 1996; Rozé and Cordier 1998; Sampath et al. 1995; Teneketzis 1996; Teneketzis and Andersland 2000; Thorsley and Teneketzis 2005; Witsenhausen 1971, 1975; Yoo and Garcia 2003; Yoo and Lafortune 2002a), the objective is to select the fixed observation mask that minimizes the cost associated with purchasing sensors that are then activated for the duration of the discrete-event process. In the active acquisition of information problem, the observer actively decides which sensors are to be used based on the information that it has already available. A cost is incurred each time a sensor is activated in an attempt to sense the event associated with that sensor. If a sensor is never activated, the system does not incur a cost from that sensor, even if it is available for the observer to use.

Variations on the active acquisition of information approach have been applied to many classes of systems other than DES. For example, problems involving sensors that can be activated or deactivated based on the system behavior have been considered for many different classes of systems, including centralized and decentralized linear stochastic systems (e.g., Athans 1972; Kushner 1964; Meier III et al. 1967; Khanna 1973; Andersland and Teneketzis 1996), communication networks, (Rago et al. 1996; Appadwedula et al. 2002) and operations research Ding et al. (2002). In this paper we consider a version of the problem where the decision as to what sensors are activated is made by a centralized diagnoser; a schematic of this diagnoser is shown in Fig. 1. In the architecture this paper considers, the diagnoser contains an observer that reads in data from a DES. It then sends the information it has obtained to a policy maker that instantaneously feeds back to the observer the set of events it should next attempt to observe.

Furthermore, although the primary focus of this paper is on the use of active acquisition of information for fault diagnosis, the active acquisition method can also be applied to the case of the supervisory control problem. In this problem, not only does the policy maker choose a set of sensors for the observer to activate; it also enables or disables certain events in the DES itself based on the information it has received from the observer in order to ensure that the controlled system achieves a given specification. Despite the differences between the supervisory control and diagnosis problem, we show how they can approached in a similar manner using the active acquisition of information method.

In the paper we consider the active acquisition problem for three classes of automata. The development of the information structure is simplest in the case where the automaton is acyclic and events are synchronized to ticks of a clock. The second

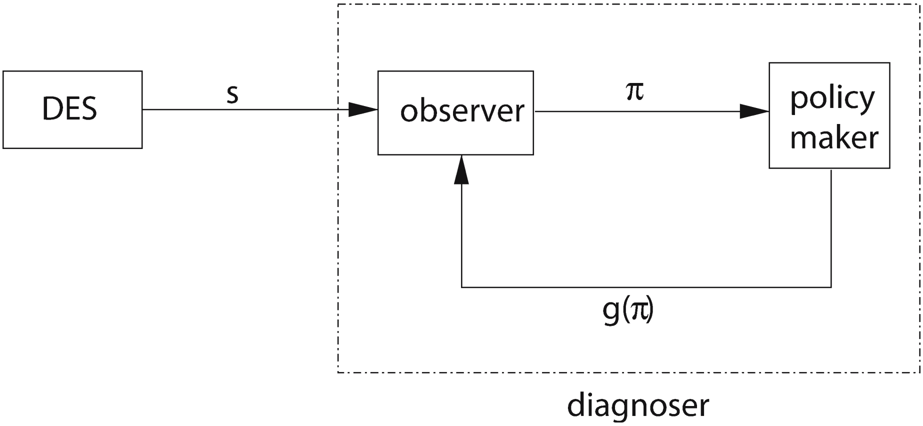


Fig. 1 Block diagram of the active acquisition system for diagnosis of DES

class we consider is the case where the automata are acyclic, but events are no longer synchronized; we place a mild assumption required that the time between successive events in the system's evolution is not only finite, but also bounded. The final class we consider is general, cyclic, automata.

We divide this paper into three sections. In each section we describe the DES model under consideration and define the active acquisition of information problem in its particular context. We then define appropriate spaces of information states for the particular class of automata. For the two acyclic cases, we describe a method for finding an optimal observation policy; in the cyclic case, we determine conditions for the existence of an optimal policy. In the acyclic, synchronous case, we describe a limited lookahead algorithm for computational savings. In the acyclic, asynchronous, case, we describe how to find optimal policies for both diagnosis and supervisory control problems. Throughout the paper, we discuss both stochastic and logical DES models and illustrate the results with examples.

The division of the paper is done so as to start with the simplest formulation of the active acquisition of information for diagnosis problem, introduce the key solution ideas within the context of that problem, and then show how these solution ideas evolve as one considers more complicated versions of the problem.

2 Acyclic timed automata

2.1 Modeling formalism

In the section we consider the simplest case of the active acquisition of information problem for diagnosis for DES for a restricted class of automata. A (logical) automaton is defined as $G = (X, \Sigma, \delta, x_0)$, where

- Σ is a finite set of events
- X is a finite state space
- $\delta : X \times \Sigma \to X$ is the partial transition function
- $x_0 \in X$ is the initial state

A logical automaton G generates a language $\mathcal{L}(G)$. To simplify the development of the problem, we make the following assumptions about the automaton:

(A1) The automaton G is acyclic. Therefore, there exists a constant T that bounds all the traces in the language generated by G. Traces that terminate before reaching the bound T can be extended by adding the appropriate number of ϵ transitions, where ϵ denotes the empty trace.

(A2) Events are synchronized to ticks of a clock, i.e., there is a constant amount of time between the occurrence of two successive events.

Assumption (A1) ensures that the worst case observation cost of the system remains finite and forces the existence of a finite horizon T. Assumption (A2) simplifies the development of the concepts of information state and σ-field that will be used to solve the active acquisition problem.

As there is a constant amount of time between events, we define for all $t \leq T$,

$$L_t = \{s : s \in \mathcal{L}(G) \wedge \|s\| = t\}. \tag{1}$$

L_t is simply the language that can be realized by the automaton at time t. In particular, L_T denotes the set of all strings realizable by the automaton when our observation of its behavior is completed.

In the active acquisition problem, an event is called observable if there is an available sensor that can detect its occurrence (although at any moment we may choose not to use that particular sensor) and it is called unobservable if there is no such available sensor. Formally, the event set is partitioned as $\Sigma = \Sigma_{\rm o} \dot{\cup} \Sigma_{\rm uo}$, where $\Sigma_{\rm o}$ is the set of observable events and $\Sigma_{\rm uo}$ is the set of unobservable events.

In an observation or diagnosis problem with a fixed set of activated sensors, the information available to the observer is defined using the projection operation (Cassandras and Lafortune 1999). Under assumption (A2), we define the projection for a timed system as $P : \Sigma^* \rightarrow (\Sigma_{\rm o} \cup \epsilon_{\rm o})^*$

$$P(\epsilon) = \epsilon \tag{2}$$

$$P(\sigma) = \begin{cases} \sigma & \text{if } \sigma \in \Sigma_{\rm o} \\ \epsilon_{\rm o} & \text{otherwise} \end{cases} \tag{3}$$

$$P(s\sigma) = P(s)P(\sigma). \tag{4}$$

The symbol $\epsilon_{\rm o}$ is considered to be observable; it indicates that no event in the alphabet Σ was observed at a particular time.

Similarly, the inverse projection is defined as

$$P_L^{-1}(s') = \left\{ s \in L : P(s) = s' \right\}. \tag{5}$$

In general, the inverse projection operation does not yield a single trace, but instead a set of traces in $\mathcal{L}(G)$.

There is a cost $\nu : \Sigma_{\rm o} \rightarrow [0, \infty)$ associated with activating each sensor in order to identify an occurrence of an observable event. If $\nu(\sigma) = 0$, then σ is said to be *freely observable*; the set of all freely observable events is denoted by $\Sigma_{\rm fo}$. Otherwise, σ is said to be *costly*; the set of all costly observable events is denoted by $\Sigma_{\rm co}$. The cost of an observation action $u \in 2^{\Sigma_{\rm co}}$ is simply the sum of the costs of each event observable under that action:

$$c(u) = \sum_{\sigma \in u} \nu(\sigma). \tag{6}$$

We use the symbol ν to denote the cost of a single event and the symbol c to denote the cost of an observation action to prevent confusion in later sections of this paper, where an action by the policy maker consists of making both control and observation decisions.

The set of failure events to be diagnosed is $\Sigma_{\rm f} \subseteq \Sigma$. We assume that $\Sigma_{\rm fo} \cap \Sigma_{\rm f} = \emptyset$, as it is a trivial problem to diagnose a failure that can be freely observed. The set of failure events is partitioned into a set of failure types $\Sigma_{\rm f} = \Sigma_{\rm f_1} \cup \cdots \cup \Sigma_{\rm f_m}$. If a failure event $f \in \Sigma_{\rm f_i}$ occurs, this is equivalent to the phrase "a failure of type F_i occurs."

Our objective is to find an optimal observation policy that diagnoses $\mathcal{L}(G)$ in the sense defined in the next subsection.

2.2 Problem formulation

The active acquisition problem is a problem of optimization. We wish to find an observation policy that minimizes the observation cost while allowing for the detection of any failures by the time the process terminates.

To formulate the active acquisition of information problem we need to introduce the following concepts. Let

$$\chi_t : L_t \rightarrow 2^{L_T} \tag{7}$$

be defined as for $0 \leq t \leq T$:

$$\chi_t(s') = \left\{ s \in L_T : P(s') \text{ is a prefix of } P(s) \right\}. \tag{8}$$

A string s' is a prefix of itself; it follows that $\chi_T(s') = \{s \in L_T : P(s) = P(s')\}$.

The functions χ_t are used in the following definition.

Definition 1 An observation policy $g := (g_0, \ldots, g_{T-1})$ is a sequence of functions $g_t : L_T \rightarrow 2^{\Sigma_{co}}$ such that for all $s' \in L_t$ and $s, \hat{s} \in \chi_t(s')$, $g_t(s) = g_t(\hat{(s)})$.

We next define the family of "maximal σ-fields" for the model of Section 2.1. This family of σ-fields plays a key role in the solution of the active acquisition problem. Let R_t be the range of χ_t, $t = 0 \ldots T$. For each t, R_t is a subset of 2^{L_T} that is also a partition of L_T. Furthermore, R_{t+1} is a finer partition of L_T than R_t for $t{=}0 \ldots T{-}1$.

Definition 2 The maximal σ-field $\mathcal{F}_t$ at time t, $t = 0 \ldots T$, is

$$\mathcal{F}_t = \sigma(\pi_t : \pi_t \in R_t), \tag{9}$$

where $\sigma(\pi_t : \pi_t \in R_t)$ denotes the σ-field generated by the elements of the partition R_t.

Since for each t, the partition R_{t+1} is finer than R_t, it follows that $\mathcal{F}_t \subseteq \mathcal{F}_{t+1}$ for each t, therefore $\{\mathcal{F}_t, t = 0 \ldots T\}$ is a *filtration* (Pollard 2002). By an argument similar to the above we can define the filtration $\{\mathcal{G}_t^g, t = 0 \ldots T\}$ corresponding to the observation policy g. Let P^g denote the projection operator corresponding to g. For any string $s \in L_T$, $P^g(s)$ selects only the events that are observed by g along s. For $t = 0 \ldots T$, let $\chi_t^g : L_t \rightarrow 2^{L_T}$ be defined by:

$$\chi_t^g(s') = \left\{ s \in L_T : P^g(s') \text{ is a prefix of } P^g(s) \right\}. \tag{10}$$

Let R_t^g be the range of χ_t^g, $t = 0 \ldots T$. For each t, R_t^g is a subset of 2^{L_T} that is also a partition of L_T. Furthermore, for each t, R_{t+1}^g is a finer partition of L_T than R_t^g.

Definition 3 The filtration $\{\mathcal{G}_t^g, t = 0 \ldots T\}$ corresponding to g is

$$\sigma\left(\pi_t : \pi_t \in R_t^g\right), t = 0 \ldots T. \tag{11}$$

Since the automaton G is acyclic by assumption (A1), we simply desire that there exists an observation policy so that when the process terminates at time T, we can be certain as to whether or not a failure has occurred. To formalize this, we need the following definitions.

Definition 4 A set of strings $S \in 2^{\mathcal{L}(G)}$ is *certain* if, for all failure types F_i, either every string $s \in S$ contains an event in Σ_{F_i} or no $s \in S$ contains any event in Σ_{F_i}.

Definition 5 A language $\mathcal{L}(G)$ is *diagnosed* by an observation policy g if, for all $s \in L_T$, $\chi_T^g(s)$ is certain with respect to all types of failures.

Definition 6 Let H denote the set of policies that diagnose $\mathcal{L}(G)$. The language $\mathcal{L}(G)$ is *diagnosable* if H is non-empty, i.e., if there exists a policy that diagnoses $\mathcal{L}(G)$.

The problem under consideration is to find an observation policy that diagnoses $\mathcal{L}(G)$ at minimal worst-case cost. Define the performance criterion:

$$J(g) = \max_{s \in L_T} \left\{ \sum_{t=1}^{T-1} c_t^g(s) + K_T^g(s) \right\}, \tag{12}$$

where $c_t^g(s)$ denotes the cost of implementing policy g at time t along the trajectory s, and $K_T^g(s)$ denotes the final penalty incurred after implementing policy g along s. $K_T^g(s)$ is defined as

$$K_T^g(s) = \begin{cases} 0 & \text{if } \chi_T^g(s) \text{ is certain} \\ \infty & \text{otherwise.} \end{cases} \tag{13}$$

The performance criterion is thus the maximum total cost of policy g for $t = 0 \ldots T$.

The active acquisition of information problem for diagnosis of acyclic timed systems is defined as follows.

Problem A Find a policy $g^* \in H$ such that

$$J(g^*) = \inf(J(g) | g \in H). \tag{14}$$

In the remainder of this section we present a systematic methodology for solving Problem A.

2.3 Information states for active acquisition

The difficulty in the active acquisition problem is the derivation of a systematic method of determining how information regarding the system behavior evolves as different events are observed at different stages of the system's evolution. To develop this method, we use a maximal σ-field approach. This approach was initially proposed in Witsenhausen (1971, 1975) in the context of general informationally decentralized systems and was further used in Andersland and Teneketzis (1992, 1994), Teneketzis (1996), Teneketzis and Andersland (2000).

For each $t, t = 0 \ldots T$, we select $\mathcal{F}_t$, defined by Definition 2, to be the space of *information states* for Problem A. First we show that $\pi_t \in \mathcal{F}_t$ satisfies the requirements of an information state as defined in Kumar and Varaiya (1986). For any action $u \in 2^{\Sigma_{co}}$ at time t, the information state $\pi_t \in \mathcal{F}_t$ is updated as

$$\pi_{t+1} = \hat{\delta}_u(\pi, \sigma) = \begin{cases} \{s \in \pi : s_{t+1} = \sigma\} & \text{if } \sigma \in \Sigma_{u,\text{obs}} \\ \{s \in \pi : s_{t+1} \in \Sigma_{u,\text{unobs}}\} & \text{otherwise.} \end{cases} \tag{15}$$

where s_{t+1} denotes the $(t+1)$st event in the string $s \in L_T$, $\Sigma_{u,\text{obs}}$ denotes the set of events that are observable under the action u, and $\Sigma_{u,\text{unobs}}$ denotes the set that is not observable under u. Furthermore, by its definition, π_t is a function of the data available up to time t. Consequently, $\pi_t \in \mathcal{F}_t$ satisfies the requirements of Definition 6.4.2 in Kumar and Varaiya (1986). In Section 2.4, we will prove that $\pi_t \in \mathcal{F}_t, t = 0 \ldots T$ is appropriate for performance evaluation, that is, it can be used to determine an optimal observation policy. Therefore, $\mathcal{F}_t$ is suitable to be the space of information states for Problem A at time $t, t = 0 \ldots T$. Before proving that π_t is suitable for finding an optimal observation policy we establish a relationship between the family of maximal σ-fields $\mathcal{F}_t, t = 0 \ldots T$ and the filtration $\{\mathcal{G}_t^g, t = 0 \ldots T\}$ corresponding to any policy g.

Theorem 1 *Consider any fixed observation policy g. For any $t, t = 0 \ldots T$ and any $s' \in L_t$, $\chi_t^g(s') \in \mathcal{F}_t$.*

Proof For any fixed policy g, any $t, t = 0 \ldots T$, and any $s' \in L_t$, we have

$$\chi_t^g(s') = \left\{s \in L_T : P^g(s') \text{ is a prefix of } P^g(s)\right\} \tag{16}$$

$$= \left\{s \in L_T : \exists s'' \in P^{g^{-1}}[P^g(s')] \text{ such that } s'' \text{ is a prefix of } s\right\} \tag{17}$$

$$= \bigcup_{s'' \in P^{g^{-1}}[P^g(s')]} \left\{s \in L_T : s'' \text{ is a prefix of } s\right\}. \tag{18}$$

Suppose $s^1 \in \chi_t^g(s')$. Then there exists $s'' \in P^{g^{-1}}[P^g(s')]$ such that s'' is a prefix of s^1. Now suppose $s^2 \in L_T$ is a string such that $P(s^2) = P(s^1)$; since $P(s'')$ is a prefix of $P(s^1)$, $P(s'')$ is also a prefix of $P(s^2)$. It follows that $P^g(s'')$ is a prefix of $P^g(s^2)$, because $P(s'')$ is a prefix of $P(s^2)$ and the projection P^g associated with the policy g has less refined information than the full projection P.

Therefore $\exists s''' \in P^{g^{-1}}[P^g(s'')]$ such that s''' is a prefix of s^2. Since $s'' \in P^{g^{-1}}[P^g(s')]$, it follow that $P^{g^{-1}}[P^g(s'')] \in P^{g^{-1}}[P^g(s')]$. Thus $\exists s''' \in P^{g^{-1}}[P^g(s')]$. Therefore, from Eq. 17, $s^2 \in \chi_t^g(s')$. It follows that for any $s^1, s^2 \in L_T$, if $s^1 \in \chi_t^g(s')$ and $P(s^2) = P(s^1)$, then $s^2 \in \chi_t^g(s')$. Then,

$$\chi_t^g(s') \supseteq \bigcup_{s'' \in P^{g^{-1}}[P^g(s')]} \left\{s \in L_T : P(s'') \text{ is a prefix of } P(s)\right\}. \tag{19}$$

To show set equality between the right and left-hand sides of the above statement, consider $s^3 \in L_T$ such that there does not exist $s'' \in P^{g^{-1}}[P^g(s')]$ such that $P(s'')$ is a prefix of $P(s^3)$. Then there does not exist an s'' such that s'' is a prefix of s^3. From Eq. 18, it follows that $s^3 \notin X_t^g(s')$. Thus the set inclusion above can be replaced by an inequality.

Since the right-hand side of Eq. 19 is a countable union of elements of R_t, it is an element of the σ-field $\mathcal{F}_t$. Therefore, $\chi_t^g(s') \in \mathcal{F}_t$. □

From Theorem 1, it follows that for any observation policy g and any t, $t = 0 \dots T$, $\mathcal{G}_t^g \subseteq \mathcal{F}_t$. Because the filtration defined by any observation policy g is no greater than the filtration $\{\mathcal{F}_t, t = 0 \dots T\}$, $\mathcal{F}_t, t = 0 \dots T$ has been defined as the family of maximal σ -fields.

The maximal σ-fields defined in Eqs. 8–9 are independent of the observation policy chosen by the policy maker. For every problem we formulate in this paper we define a family of σ-fields $\{\mathcal{F}_t\}$, $t = 0, 1, 2, \dots$ that have the following properties: (1) They are independent of the observation policy; (2) The filtrations $\mathcal{G}_t^g$, $t = 0, 1, 2, \dots$ resulting from any observation policy g are sub-σ-fields of the set of maximal σ-fields, i.e., for all t, $\mathcal{G}_t^g \subseteq \mathcal{F}_t$. $\mathcal{F}_t$, $t = 0 \dots T$, is the smallest family of σ-fields that satisfy this property. Such a choice of maximal σ-fields reduces the off-line computation required for the solution of the dynamic program that determines optimal observation policies.

Having developed a method to describe the information state and a sequence of maximal σ-fields in which the information state must reside, we now address the question of how to determine the existence of an optimal observation policy and develop a method to find such a policy when it exists.

2.4 Finding an optimal observation policy

In this subsection we first present a criterion for diagnosability that can be used to determine if an optimal observation policy exists. We then present a method of determining a policy which minimizes a worst case observation cost, subject to the constraint that all failures in the system are diagnosed.

2.4.1 Existence of an optimal policy

In order for a solution to Problem A to exist, the set of admissible observation policies H must be non-empty, i.e., the language $\mathcal{L}(G)$ must be diagnosable. Therefore the condition for existence of a solution to Problem A is simply the condition for diagnosability.

Theorem 2 *$\mathcal{L}(G)$ is diagnosable if and only if all elements of the partition R_T of L_T that generates $\mathcal{F}_T$ are certain.*

Proof (Sufficiency) Suppose each element of the partition R_T of $\mathcal{L}(G)$ that generates $\mathcal{F}_T$ is certain. Let $g_{\max}$ denote the policy where $g_t(\pi_t) = \Sigma_{\text{co}}$ for all π_t and all $t = 0, 1, \dots, T-1$, i.e., the policy where all costly sensors are always activated. Along any string in L_T, the only strings consistent with the observations made under $g_{\max}$ have identical projections onto Σ_{o}; therefore, the information state reached along any string $s \in L_T$ is an element of the partition R_T of L_T that generates $\mathcal{F}_T$ Since that information state is F-certain, $g_{\max}$ diagnoses $\mathcal{L}(G)$.

(Necessity) We prove necessity by proving the contrapositive statement. Suppose that there exists an element of the partition R_T of $\mathcal{L}(G)$ that generates $\mathcal{F}_T$ that is uncertain. Then there exist two traces $s_1, s_2 \in L_T$ such that $P(s_1) = P(s_2)$, where P is the projection of Σ onto Σ_{o} and $s_1 + s_2$ is uncertain.

Select any observation policy g and consider the information state reached by implementing g along s_1. That information state contains both s_1 and s_2; therefore it is uncertain. Since g was arbitrarily chosen, it follows that there is no policy that diagnoses $\mathcal{L}(G)$. □

Having demonstrated a criterion for testing the diagnosability of a language, we now present a dynamic programming technique to find an optimal observation policy when this criterion is satisfied.

2.4.2 Active acquisition dynamic program

The active acquisition dynamic program for Problem A is

$$V_T(\pi) = \begin{cases} 0 & \text{if } \pi \in \mathcal{F}_T \text{ is certain} \\ \infty & \text{otherwise,} \end{cases} \tag{20}$$

$$V_t(\pi) = \min_{u \in 2^{\Sigma_{co}}} \left\{ c_u + \max_{\sigma \in \Sigma} V_{t+1}\big(\hat{\delta}_u(\pi, \sigma)\big) \right\} \text{ for } \pi \in \mathcal{F}_t, t = 0 \ldots T-1, \tag{21}$$

where $\hat{\delta}_u$ is defined in Eq. 15.

We demonstrate the information state defined is suitable for finding an optimal observation policy. with the following theorem.

Theorem 3 *The solution of the dynamic program, defined by Eqs. 20–21, is a solution to Problem A. That is, the solution to Eqs. 20–21 determines an optimal observation policy $g^* := (g_1^*, g_2^*, \ldots, g_{T_1}^*)$ and the corresponding optimal cost $J(g^*) = V_0(L_T)$. The optimal cost is the minimum worst case observation cost that diagnoses $\mathcal{L}(G)$.*

Proof We follow the philosophy of Chapter 6 of Kumar and Varaiya (1986).

To prove the theorem, we verify the following two statements:

(1) Consider any admissible observation policy $g \in H$, and let $\pi_t(s^{g,t}) \in \mathcal{F}_t$ denote the information state resulting when g is implemented and $s^{g,t}$ is observed up to time t. Then for all $t = 0 \ldots T$,

$$V_t\left(\pi_t\left(s^{g,t}\right)\right) \leq J_t^g\left(s^{g,t}\right) := \max_{s \in L_T/\pi_t(s^{g,t})} \left\{ \sum_{\ell=t}^{T-1} c_\ell^g(s) + K_T^g(s) \mid s^{g,t} \right\}, \tag{22}$$

where $L_T/\pi_t(s^{g,t})$ is the postlanguage of the information state $\pi_t(s^{g,t})$.

(2) Let g^* be an observation policy such that for all t and for all $\pi \in \mathcal{F}_t$, $g_t^*(\pi)$ achieves the minimum in Eq. 21. Then g^* is an optimal observation policy and

$$V_t\left(\pi_t\left(s^{g,t}\right)\right) = J_t^{g^*}\left(s^{g,t}\right) \tag{23}$$

for all $s^{g,t}$.

The proof of (1) proceeds by induction. For $t = T$, let

$$J_T^g\left(s^{g,t}\right) = \left\{K_T^g(S) \mid s^{g,t}\right\}, \tag{24}$$

where S is the set of all $s \in L_T$ that produce an observable trace equal to that of $s^{g,t}$ under the observation policy g. Therefore $S = \pi_t\left(s^{g,t}\right)$ and

$$J_T^g\left(s^{g,T}\right) = \begin{cases} 0 & \text{if } \pi_t\left(s^{g,T}\right) \text{ is certain} \\ \infty & \text{otherwise.} \end{cases} \tag{25}$$

From Eqs. 20 and 25 we conclude that:

$$V_T\left(\pi_t\left(s^{g,t}\right)\right) = J_T^g\left(s^{g,T}\right) \tag{26}$$

for all $s^{g,T} \in L_T$. This establishes the basis for the induction.

Suppose now that Eq. 22 holds for the index $t+1$. Then, by the induction hypothesis and Eq. 21,

$$\begin{aligned} J_t^g\left(s^{g,T}\right) = &\left\{ c_{g\left(\pi\left(s^{g,T}\right),t\right)} + \max_{\sigma\in\Sigma} \left\{ \max_{s\in L_T/\hat{\delta}_{g_t\left(\pi\left(s^{g,T}\right)\right)}\left(\pi\left(s^{g,t},\sigma\right)\right)} \right.\right. \\ &\left.\left. \left\{ \sum_{\ell=t+1}^{T-1} c_\ell^g(s) + K_T^g(s) \mid s^{g,t}\sigma \right\} \mid s^{g,t} \right\}\right\} \end{aligned} \tag{27}$$

$$\geq c_{g\left(\pi\left(s^{g,T}\right),t\right)} + \max_{\sigma\in\Sigma} V_{t+1}\left(\hat{\delta}_{g_t\left(\pi\left(s^{g,T}\right)\right)}\left(\pi\left(s^{g,t},\sigma\right)\right)\right) \tag{28}$$

$$\geq \min_{u\in 2_{\text{co}}^{\Sigma}} \left\{ c_u + max_{\sigma\in\Sigma} V_{t+1}\left(\hat{\delta}_{g_t\left(\pi\left(s^{g,t},t\right)\right)}\left(\pi\left(s^{g,t},\sigma\right)\right)\right)\right\} \tag{29}$$

$$= V_t\left(\pi\left(s^{g,t}\right)\right). \tag{30}$$

The first inequality holds due to the induction hypothesis; the second inequality holds because $g \in H$ is fixed and not necessary optimal. Therefore Eq. 22 holds for t and this completes the induction step and proof of statement (1).

To prove (2), we first prove Eq. 23 by induction. From Eq. 26 we know Eq. 23 holds for $\ell = T$. Suppose it holds for $\ell = t+1$. Then, in the derivation of Eq. 30 inequalities become equalities. The first inequality becomes equality by the induction hypothesis; the second inequality becomes inequality because, for every $s^{g^*,t}$, $g_t^*\left(\pi\left(s^{g^*,t}\right)\right)$ achieves the minimum. Thus Eq. 23 holds for all $s^{g^*,t}$, for all t. It remains to show that g^* is optimal. For $t = 0$, Eq. 23 gives

$$J_0^{g^*} = V_0(L_T), \tag{31}$$

where L_T is the known initial information state of the automaton. For any other $g \in H$, setting $t = 0$ we obtain

$$J_0^g \geq V_0(L_T). \tag{32}$$

Therefore g^* is an optimal observation policy, and $V_0(L_T)$ is the corresponding optimal cost. □

2.5 Example

We illustrate the results of the previous subsection by applying the active acquisition algorithm to the finite-state machine in Fig. 2. In this example, $\Sigma_{\text{uo}} = \{f, u\}$, $\Sigma_{\text{co}} =$

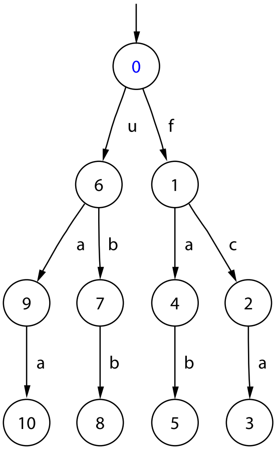


Fig. 2 An automaton used to illustrate the active acquisition method. $\Sigma_{\text{uo}} = \{f, u\}$, $\Sigma_{\text{co}} = \{a, b, c\}$, $\Sigma_{\text{f}} = \{f\}$, and $T = 3$

$\{a, b, c\}$, and $\Sigma_{\text{f}} = \{f\}$. The costs associated with each event are $\nu_a = 4$, $\nu_b = 1$, and $\nu_c = 2$.

The longest trace in the language of this automaton contains three events. The final σ-field is thus $\mathcal{F}_3$, defined as:

$$\mathcal{F}_3 = \sigma(ubb,\ fca, uaa,\ fab). \tag{33}$$

The elements of $\mathcal{F}_3$ are listed in the first column of Table 1. For each $\pi \in \mathcal{F}_3$, we assign a cost based on whether or not the information state is certain; these costs are shown in the second column of Table 1.

The σ-field $\mathcal{F}_2$ is a proper subset of $\mathcal{F}_3$, given by the following:

$$\mathcal{F}_2 = \sigma(ubb,\ fca, uaa + \ fab). \tag{34}$$

Table 1 Information states and their associated costs for the automaton in Fig. 2

π	$V_3(\pi)$	$V_2(\pi)$	$V_1(\pi) = V_0(\pi)$
uaa	0	—	—
ubb	0	0	—
fab	0	—	—
fca	0	0	—
$uaa + ubb$	0	—	—
$fab + fca$	0	—	—
$uaa + fab$	∞	1	—
$uaa + fca$	∞	—	—
$ubb + fab$	∞	—	—
$ubb + fca$	∞	1	—
L_3/uaa	∞	—	—
L_3/ubb	∞	∞	—
L_3/fab	∞	—	—
L_3/fca	∞	∞	—
L_3	∞	∞	4

Table 2 Calculation of an optimal observation action for the information state $uaa + fab$ at time $t = 2$

u	$c(u)$	$\hat{\delta}_{u,2}(\pi, a)$	$\hat{\delta}_{u,2}(\pi, b)$	$\max V_3\big(\hat{\delta}_{u,2}(\pi, \sigma)\big)$
$\emptyset$	0	$uaa + fab$	$uaa + fab$	∞
$\{a\}$	4	uaa	fab	0
$\{b\}$	1	uaa	fab	0
$\{a, b\}$	5	uaa	fab	0

The strings uaa and fab have an identical projection up to time $t = 2$ and thus are part of the same element of the partition of L_3 that generates $\mathcal{F}_2$.

For each $\pi \in \mathcal{F}_2$, the cost $V_2(\pi)$ is calculated using the dynamic programming equation:

$$V_2(\pi) = \min_{u \in 2^{\Sigma_{co}}} \left\{ c(u) + \max_{\sigma \in \Sigma} V_3\big(\hat{\delta}_u(\pi, \sigma)\big) \right\}. \tag{35}$$

The determination of an optimal observation action for the information state $uaa + fab$ at time $t = 2$ is shown in Table 2. Since c cannot be the next event from this information state, four observation actions must be evaluated at $uaa + fab$: $\emptyset$, $\{a\}$, $\{b\}$, and $\{a, b\}$.

Table 2 indicates that an optimal observation action for this information state is $\{b\}$; therefore the cost of the state V_2 is $\nu_b = 1$. The values of V_2 for all $\pi \in \mathcal{F}_2$ are shown in Table 1.

All strings in L_3 have the same projection up to $t = 1$ and thus $\mathcal{F}_1 = \sigma(L_3) = \{L_3, \emptyset\}$.

$$V_1(L_3) = \min_{u \in 2^{\Sigma_{co}}} \left\{ c(u) + \max_{\sigma \in \Sigma} V_1\left(\hat{\delta}_u\left(L_3, \sigma\right)\right) \right\}. \tag{36}$$

The value of $V_1(L_3)$ computed by this equation is 4, corresponding to the observation action $\{b, c\}$.

At $t = 0$, since both events are unobservable, the dynamic programming equation indicates that $V_0(L_3) = V_1(L_3)$. Therefore the minimum worst case observation cost is $V_0(L_3) = 4$.

Table 3 shows an optimal policy $g^* = \left(g^*_0, g^*_1, g^*_2\right)$ for all information states that are reachable under g^*. Note that, in order to determine which information states were reachable, it was necessary to determine an optimal observation policy for all information states.

Table 3 An optimal observation policy for diagnosing the automaton in Fig. 2

π	g^*_0	g^*_1	g^*_2
ubb	—	—	$\emptyset$
fca	—	—	$\emptyset$
$uaa + fab$	—	—	$\{b\}$
L_3	$\emptyset$	$\{b, c\}$	—

Only reachable information states are shown

2.6 Limited lookahead algorithm

Determining an optimal observation policy using the method described in the previous subsection can become computationally formidable for large T. In this subsection, we propose a limited lookahead algorithm that approximates an optimal observation policy.

Roughly speaking, in the limited lookahead algorithm a sequence of active acquisition programs are run for a time horizon $T' < T$. Information states at T' are assigned infinite cost only if it is not possible to diagnose $\mathcal{L}(G)$ from all possible future observations. This notion of information state diagnosability (as opposed to diagnosability of a language, which was defined in Definition 6) is formalized in the following definition.

Definition 7 An information state $\pi \in \mathcal{F}_t$ is *diagnosable at time t* if the cost $V_t(\pi)$ determined by the active acquisition dynamic program is finite.

Definition 7 indicates that from a diagnosable information state, the cost-to-go required to diagnose $\mathcal{L}(G)$ is finite. The following statement is equivalent to Definition 7, and relates the concepts of information state diagnosability and language diagnosability.

Theorem 4 *Express an information state as $\pi = s^1t^1 + s^2t^2 + \cdots + s^nt^n$, where $\|s^i\| = t$ for $i = 1 \ldots n$. The information state π is diagnosable at time t if and only if the language $L_\pi := \hat{P}(s^1)t^1 + \hat{P}(s^2)t^2 + \cdots + \hat{P}(s^n)t^n$ is diagnosable at finite cost, where $\hat{P}$ is the projection of Σ onto Σ_{uo}.*

Proof (Sufficiency) Suppose that $V_t(\pi) < \infty$. Then there exists a policy $g = (g_t, g_{t+1}, \ldots, g_{T-1})$ such that the information state π_T reached by implementing g along any t_i is certain. The final information state π_T consists of those $s_jt_j \in \pi$ that are consistent with the observations made along t_i under policy g.

To diagnose L_π, implement the policy $g' = (\emptyset, \emptyset, \ldots, \emptyset, g_t, g_{t+1}, \ldots, g_{T-1})$. Since the first t events along any string in L_π are unobservable, along any string $\hat{P}(s^i)t^i$, the final information state π'_T consists of those $\hat{P}(s^j)t^j$ that are consistent with the observations made along t_i. Since the policy g' is identical to g for times greater than t, $\hat{P}(s^j)t^j \in \pi'_T$ if $s^jt^j \in \pi$. Since $\hat{P}(s^j)t^j$ and s^jt^j contain the same failure events, π'_T is certain if π_T is certain. Therefore, the policy g' diagnoses L_π.

(Necessity) We prove necessity by proving the contrapositive statement. Suppose that $V_t(\pi) = \infty$. Then for any $g = (g_t, g_{t+1}, \ldots, g_{T-1})$, there exists a t^i such that the information state π_T reached by implementing g along t^i is uncertain.

Select any policy $g' = (g_0, g_1, \ldots, g_{t-1}, g_t, \ldots, g_{T-1})$ and consider the final information state reached by implementing g' along $\hat{P}(s^i)t^i$. Again, since the first t events along any string in L_π are unobservable, the final information state π'_T consists of those $\hat{P}(s^j)t^j$ that are consistent with the observations made along t^i. Since $\hat{P}(s^j)t^j$ and s^jt^j contain the same failures, π'_T is uncertain if π_T is uncertain. Since for any g we can choose a t^i such that π_T is uncertain, for any g' we can choose a $\hat{P}(s^i)t^i$ such that π'_T is uncertain. Therefore L_π is not diagnosable. □

To start the limited lookahead algorithm, we choose a horizon $T' < T$ and consider the σ-field $\mathcal{F}_{T'}$. For each information state $\pi \in \mathcal{F}_{T'}$, a cost is assigned as follows:

$$V_{T'}(\pi) = \begin{cases} 0 & \text{if } \pi \text{ is diagnosable at } T' \\ \infty & \text{otherwise.} \end{cases} \tag{37}$$

The cost $V_{T'}(\pi)$ is assigned to each element in $\mathcal{F}_{T'}$ by constructing the language L_π described in Theorem 4, and then using the result of Theorem 2 to determine whether or not L_π is diagnosable.

The dynamic programming equation solved is identical to that in the previous subsection:

$$V_{t-1}(\pi) = \min_{u \in 2^{\Sigma_{co}}} \left\{ c_u + \max_{\sigma \in \Sigma} V_t\big(\hat{\delta}_u(\pi, \sigma)\big) \right\} \text{for } \pi \in \mathcal{F}_{t-1}, t = 1, 2, \ldots, T' - 1. \tag{38}$$

Once the dynamic program is solved, $V_0(L_{T'})$ and an observation action $g_0^*(L_{T'})$ for $t = 0$ are determined. The observer then implements $g_0^*(L_{T'})$ and calculates the information state at $t = 1$ based on $g_0^*(L_{T'})$ and its observation.

For $0 < t \leq T - T'$, the observer generates a sub-σ-field $\mathcal{G}_{T'+t} \subseteq \mathcal{F}_{T'+t}$ by considering only those elements in $\mathcal{F}_{T'+t}$ that are reachable from π_t, the information state at time t resulting from the implementation of policy $g_0^*, g_1^*, \ldots, g_{t-1}^*$ along the system trajectory up to time $t - 1$. This sub-σ-field $\mathcal{G}_{T'+t}$ is defined as:

$$\mathcal{G}_{T'+t} = \{A \in \mathcal{F}_{T'+t} : A \cap \pi_t = A\}. \tag{39}$$

Costs are assigned to each element of $\mathcal{G}_{T'+t}$ as:

$$V_{T'+t}(\pi) = \begin{cases} 0 & \text{if } \pi \text{ is diagnosable at } T' + t, \pi \in \mathcal{G}_{T'+t} \\ \infty & \text{otherwise,} \end{cases} \tag{40}$$

and then the dynamic program in Eq. 38 is used to calculated an observation action for π_t. The observer then implements that action, calculates a new information state π_{t+1}, and iterates the algorithm to find an observation action for that information state.

The algorithm finishes when $t = T - T'$ and the observer looks ahead to the final time horizon of the system. The observer implements the policy specified by the solution of the dynamic program (20–21) where the horizon is $T - T'$ and the initial information state is $\pi_{T-T'}$.

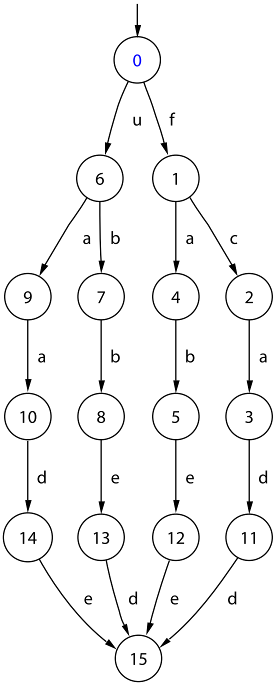
As an example, consider the automaton in Fig. 3, and suppose $\Sigma_{co} = \{a, b, c, d, e\}$, $\nu_a < \nu_b < \nu_c < \nu_d < \nu_e$ and $T' = 2$.

At $t = 0$, the observer considers the σ-field $\mathcal{F}_2 = \sigma(fc, ub, fa + ua)$. Since every element of $\mathcal{F}_2$ is diagnosable at $t = 2$, solving Eq. 38 results in the observation action $\emptyset$ at $t = 0$.

At $t = 1$, the information state generated by the observation action at $t = 0$ is necessarily $\pi_1 = L_T$. Consider $\mathcal{G}_3 = \{A \in \mathcal{F}_3 : A \cap \pi_t = A\} = \mathcal{F}_3$. Every element of $\mathcal{G}_3$ is diagnosable at $t = 3$, and, as a result of Eq. 38, the observation action is $\emptyset$.

At $t = 2$, the information state is $\pi_2 = L_T$, and the observer considers $\mathcal{G}_4 = \{A \in \mathcal{F}_4 : A \cap \pi_t = A\} = \mathcal{F}_4$. The information states $fcadd + ubbed$ and $fabee + uaade$ are not diagnosable at $t = 4$ and thus have infinite cost. Using Eq. 38, we find that an optimal observation action at $t = 2$ is to observe $\{a\}$.

Fig. 3 An automaton used to illustrate the limited lookahead method



Thus there are two possible information states at $t = 3$: if a is observed when $t = 2$, $\pi_3 = fcadd + uaade$; otherwise, $\pi_3 = fabee + ubbed$.

In the case where a is observed, the observer generates the σ-field $\mathcal{G}_{5,a}$ using Eq. 39:

$$\mathcal{G}_{5,a} = \{\emptyset,\ fcadd, uaade,\ fcadd + uaade\}, \tag{41}$$

and assigns a cost to each element of $\mathcal{G}_{5,a}$ according to Eq. 40; since no further observations can be made after $t = 5$, an information state in $\mathcal{G}_{5,a}$ is diagnosable only if it is certain. By Eq. 38, we find that the observation actions are to observe no events when $t = 3$ and then to observe $\{d\}$ when $t = 4$. A similar calculation for the case where ϵ_o is observed at $t = 3$ finds that the same sequence of actions is used there as well.

At each stage of the limited lookahead algorithm, we optimize the worst case T'-step observation cost. The result of this policy is a "procrastinating" diagnoser that makes just enough observations within the lookahead window to ensure that there is some policy that will allow the failure to be diagnosed after the window has passed.

Had the observer used the algorithm of Section 2.4, it would have determined that the worst-case observation cost is $2\nu_a$, which is less than $\nu_a + \nu_d$.

2.7 Active acquisition of information for stochastic automata

The active acquisition of information problem can be solved for stochastic automata in an analogous manner. The model is identical to that of Section 2.1, except that the partial transition function δ is extended to a state transition probability function p.

Consider a stochastic automation G_s, formally defined as:

$$G_s = (\Sigma, X, p, x_0), \tag{42}$$

where

- Σ is a finite set of events
- X is a finite state space
- $p : X \times \Sigma \times X \rightarrow [0, 1]$ defines the state transition probability function
- $x_0 \in X$ is the initial state

As in the logical case, the event set is partitioned into the sets Σ_{uo}, Σ_{fo}, and Σ_{co}, and again we assume the automaton satisfies (A1) and (A2). The state transition probability function $p(x_1, e, x_2)$, defined for all events and pairs of states, denotes the probability that, in state x_1, the event e will occur and cause a transition to state x_2. For ease of notation, we also assume that $p(x_1, e, x_2) > 0$ for at most one $x_2 \in X$, and thus define the transition function δ as $\delta(x_1, e) = x_2$ if $p(x_1, e, x_2) > 0$.

The probability that an event e follows a trace s is therefore given by:

$$\Pr(e \mid s) = p(\delta(x_o, s), e). \tag{43}$$

Consider an arbitrary but fixed observation policy g and define the expected cost corresponding to g by

$$J(g) = E^g \left\{ \sum_{t=0}^{T-1} c_t^g(s) + K_T^g(s) \right\}, \tag{44}$$

where $c_t^g(s)$ denotes the cost of implementing policy g at time t along trajectory s and $K_T^g(s)$ is the terminal cost incurred by g at time T along the string s. The cost $K_T^g(s)$ is defined as follows. For any $s \in L_T$ let

$$\hat{\chi}_T^g(s) = \left\{ s \in \chi_T^g(s) : \Pr\left(s \mid y^{g,T}(s)\right) > 0 \right\}, \tag{45}$$

where $y^{g,T}(s)$ denotes the sequence of observation incurred along the string s when g is implemented. Then

$$K_T^g(s) = \begin{cases} 0 \text{ if } \hat{\chi}_T^g(s) \text{ is certain,} \\ \infty \text{ otherwise.} \end{cases} \tag{46}$$

Using $\hat{\chi}_T^g(s)$ for $s \in L_T$, we define when a language is diagnosed by an observation policy.

Definition 8 A language $\mathcal{L}(G)$ is diagnosed by an observation policy g if, for all $s \in L_T$, $\hat{\chi}_T^g(s)$ is certain with respect to all failure types.

The above definition is the stochastic analogue to Definition 5 for logical, cyclic, timed automata. Definition 6 holds for both logical and stochastic acyclic, timed automata.

The active acquisition of information problem for diagnosis of acyclic, timed stochastic automata is defined as:

Problem SA Find a policy $g^* \in H$ such that

$$J(g^*) = \inf(J(g) | g \in H). \tag{47}$$

A space of information states appropriate for Problem SA is the space of PMFs on L_T. An information state at time t is the conditional PMF on L_T given the sequence of observations y^t up to t and the sequence of control actions u^{t-1} up to time $t-1$. The information state at $t = 0$ is the a priori PMF on L_T. The probability of each string is updated according to the following equation

$$\Pr(s_1 e s_2 \mid \pi, y_{t+1}, u_t) = \begin{cases} \frac{\Pr(e|s_1)\Pr(s_1|\pi)\Pr(s_2|s_1 e)}{\sum_{s' \in L_t} \Pr(e|s')\Pr(s'|\pi)\Pr(s_2|s'e)} & \text{if } y_{t+1} = e \\ \frac{\Pr(e|s_1)\Pr(s_1|\pi)\Pr(s_2|s_1 e)}{\sum_{s' \in L_t} \sum_{e' \in \Sigma_{u_t,unobs}} \Pr(e'|s')\Pr(s'|\pi)\Pr(s_2|s'e')} & \text{if } e \in \Sigma_{u_t,unobs} \text{ and } y_{t+1} = \epsilon_o \\ 0 & \text{otherwise.} \end{cases} \tag{48}$$

If the event e is observed at time t, the probability of all traces in L_T that do not contain e at time t must be zero; the probabilities of the remaining traces are computed by normalization. If ϵ_o is observed, the probability of all traces where an event observable under our observation action at time t occurs is zero, and the probabilities of the remaining traces are again computed by normalization.

There is a strong relationship between the stochastic and logical information state transition functions. As events are observed, traces that are not consistent with the observations are eliminated from the logical information state; in the stochastic case, the probability of these traces is set to zero. Thus, at time t, the conditional PMF is always supported on some element of the σ-field $\mathcal{F}_t$.

Furthermore, given the observations up to t and the observation actions up to $t-1$, the conditional PMF on the traces of L_T is uniquely defined and is independent of the observation policy g (this result is, as expected, consistent with that of Lemma 6.5.10 of Kumar and Varaiya (1986)).

Let P_{L_t} be the space of all PMFs supported on the elements of the σ-field $\mathcal{F}_t$, $t = 0 \dots T$. For any such PMF π, denote by $S(\pi)$ its support. Then the dynamic program that solves Problem SA is

$$V_T(\pi) = \begin{cases} \infty & \text{if } S(\pi) \text{ is certain} \\ 0 & \text{otherwise.} \end{cases} \tag{49}$$

for $\pi \in P_{L_T}$, and

$$V_{t-1}(\pi) = \min_{u \in 2^{\Sigma_{co}}} \left\{ c_u + \sum_{\sigma \in \Sigma} V_t\big(\hat{\delta}_{u,t}(\pi, \sigma)\big) \Pr(\sigma \mid \pi(t-1)) \right\} \text{ for } \pi \in \mathcal{F}_{t-1}, t = 0, 1, \dots, T, \tag{50}$$

where $\hat{\delta}_{u,t}(\pi, \sigma)$ is defined by Eq. 48. Any solution of the above dynamic program provides an optimal observation policy for Problem SA. The proofs of the optimality of this dynamic program is similar to the proof of Theorem 3 and is therefore omitted.

A remark for the remainder of the paper: The results of Section 2 show that in searching for an optimal observation policy $g := (g_0, g_1, \ldots, g_{T-1})$, it is sufficient to restrict attention to functions of the form

$$g_t : \mathcal{F}_t \to 2^{\Sigma_{co}}, t = 0 \ldots T-1, \tag{51}$$

where $\mathcal{F}_t$ is the maximal σ-field (defined in Section 2.2) at time t. Based on these results we will restrict attention to policies of the above form for all the variations of Problem A and SA formulated in this paper. The proper maximal σ-fields will be identified for each problem. The presentation in the remainder of the paper will assume that the maximal σ-fields for each problem are the spaces of information states for that problem.

3 Acyclic untimed automata

3.1 Introduction

In this section we relax the assumption (A2) from the previous section and consider an acyclic automaton that evolves in the standard manner, i.e., where events can occur spontaneously at any instant in time. For the diagnosis problem we follow a procedure similar to that of the previous section, differing only in the construction of the filtration of maximal σ-fields.

We choose to take a more general approach in this section and show how the maximal σ-field approach can be applied to both the diagnosis problem and supervisory control problem. The proper general construction of maximal σ-fields in the presence of control is presented, and the diagnosis and supervisory control problems are both presented in this framework. Solutions to both problems are given in both stochastic and logical cases.

3.2 Problem formulation

As in the previous section, we consider an automaton $G = (X, \Sigma, \delta, x_0)$ and maintain assumption (A1). We no longer require assumption (A2) synchronizing events to ticks of a clock; in its place we consider the following:

(A3) The amount of time elapsed between two successive events is bounded by some positive constant.

This assumption allows us to conclude that the system behavior has reached a final state because if no event is observed for a sufficiently long time, then the controlled system must have reached a state where no further events are both feasible and enabled.

The event set Σ is partitioned into controllable events Σ_c and uncontrollable events Σ_{uc}. For each controllable event we assign a disabling cost $\kappa : \Sigma_c \to [0, \infty)$. Since we may not wish to disable an event even though we can do so freely, we make no distinction between freely controllable and costly controllable events.

At each information state, the action we wish to take consists of two parts: deciding which set of costly observable events to observe and deciding which set of controllable events to disable. The cost of such an action $u = (u_{\text{ctrl}}, u_{\text{obs}})$ is defined as:

$$c(u) = \sum_{\sigma \in u_o} \nu(\sigma) + \sum_{\sigma \in u_c} \kappa(\sigma), \tag{52}$$

where $u_{\text{ctrl}} \in 2^{\Sigma_c}$ and $u_{\text{obs}} \in 2^{\Sigma_{co}}$.

To set up the supervisory control problem, we define a specification $K \subseteq \mathcal{L}(G)$ that represents the desired controlled behavior of the system. Assumption (A1) forces the specification K to be acyclic and we denote the maximum length of the set of strings in K by T. For the situation we define a control-observation policy g by $g : \mathcal{F}_T \to 2^{\Sigma_{co}} \times 2^{\Sigma_c}$, where $\mathcal{F}_T$ is the maximal σ-field at stage T with respect to the specification K. The family of maximal σ-fields with respect to K will be defined in Section 3.3. The σ-field $\mathcal{F}_T$ is the space of information states for the problem studied in this section. Our objective is to find a control-observation policy g that achieves the specification K (that is, it does not allow strings that extend beyond K, not does it prevent any string in K from being reached by the system) and minimizes a worst-case cost defined by Eq. 53 below. To formulate precisely this problem we need the following definitions.

Definition 9 A specification K is *realized* by a policy g if the following is true: if s is a string in $\mathcal{L}(G)$ such that s is reachable under g and there are no feasible events at s enabled under g, then $s \in K$.

Definition 10 Let H denote the set of all policies that realize K. K is *realizable* if H is non-empty.

Define the performance criterion:

$$J(g) = \left\{ \max_{s \in \mathcal{L}(G)} \sum_{t=1}^{T} c_t^g(s) + L_t^g(s) \right\}, \tag{53}$$

where $c_t^g(s)$ denotes the cost of implementing policy g at stage t along the trajectory s and

$$L_T^g(s) = \begin{cases} 0 & \text{if the information state reached by implementing} \\ & \ g \text{ along } s \text{ is a subset of } K \\ \infty & \text{otherwise.} \end{cases} \tag{54}$$

The performance criterion is thus the maximum total cost of policy g. The active acquisition of information for supervisory control with acyclic specifications is defined as follows.

Problem C Find a policy $g^* \in H$ such that

$$J(g^*) = \inf(J(g) | g \in H). \tag{55}$$

3.3 Construction of maximal σ-fields

The presence of control allows for a further refinement of information than in the case of the diagnosis problem, where the behavior of the system is merely observed and not affected by the supervisor. For example, consider two strings $s_1 = a\sigma_1 b$ and $s_2 = a\sigma_2 b$, where σ_1 and σ_2 are both unobservable and controllable. Since $P(s_1) = P(s_2) = ab$, these two strings are indistinguishable to a diagnoser that passively observes the system. However, a controller may choose to disable σ_1 and thus, when ab is observed, the controller can conclude that s_2 has occurred and s_1 has not. Thus to construct a sequence of maximal σ-fields for the control problem considered in this section, it is not sufficient to consider traces that are merely equivalent under projection; we must consider traces that are equivalent under control actions as well.

3.3.1 Control projection

Suppose that two strings contain an identical sequence of observable events. In order to have an admissible supervisor, we must choose the same control action after both of these strings occur. This control action may enable or disable any number of unobservable or observable controllable events, but we cannot take a new control action until a new observation is made.

Furthermore, suppose that between successive observable events, these two strings contain identical sets of unobservable controllable events. If we choose to disable one of these strings by disabling an unobservable event, we must also disable the second string as that string also contains any event that we can feasibly disable. Therefore, we must disable both these traces or we must disable neither.

We formalize this notion of traces that must be enabled or disabled jointly using the idea of the *control projection*. We extend the standard projection operation by introducing symbols to indicate which set of unobservable controllable events occurs between each pair of observable events in the projection.

We denote the set of symbols specifying the sets of unobservable events as C_A. Each symbol in this alphabet will be of the form 1_A, where A is a set of unobservable controllable events. For example, if our system has three unobservable controllable events $\{\alpha, \beta, \gamma\}$, the associated alphabet is $C_A = \{1_\emptyset, 1_{\{\alpha\}}, 1_{\{\beta\}}, 1_{\{\gamma\}}, 1_{\{\alpha,\beta\}}, 1_{\{\alpha,\gamma\}}, 1_{\{\beta,\gamma\}}, 1_{\{\alpha,\beta,\gamma\}}\}$.

The control projection is a string whose events alternate between the symbols indicating controllable unobservable events and observable events. Formally it is defined for events as:

$$CP(\sigma) = \begin{cases} \sigma & \text{if } \sigma \in \Sigma_o \\ 1_{\{\sigma\}} & \text{if } \sigma \in \Sigma_c \cap \Sigma_{uo} \\ \epsilon & \text{otherwise.} \end{cases} \tag{56}$$

where s_f denotes the final event in the string $CP(s)$. Each symbol in C_A indicates the unobservable events that can be disabled before the next observation is made.

When the control projection is extended to traces, we must ensure the alternation of symbols from C_A and symbols from Σ_o. In the case where two observable events may end up adjacent, we simply insert the symbol $1_\emptyset$ between them, as there as no unobservable events, either controllable or uncontrollable, between those events. When two symbols in C_A are adjacent, we must merge the two symbols using the function $\vee : C_A \times C_A \rightarrow C_A$:

$$1_A \vee 1_B = 1_{A\cup B}. \tag{57}$$

The merge function is extended to strings in $(C_A \Sigma_o) * C_A$ by maintaining all symbols in the string except for the last, which is merged with the symbol to be concatenated to the string.

$$(t1_A) \vee 1_B = t1_{A \cup B}. \tag{58}$$

Formally, the function is extended to traces as $CP : \Sigma^* \to \overline{(C_A \Sigma_o)*}$ such that:

$$CP(s\sigma) = \begin{cases} CP(s)1_{\emptyset}CP(\sigma) & \text{if } s_f \in \Sigma_o \text{ and } \sigma \in \Sigma_o \\ CP(s) \vee 1_{\{\sigma\}} & \text{if } s_f \in C_A \text{ and } \sigma \in \Sigma_c \cap \Sigma_{uo} \\ CP(s)CP(\sigma) & \text{otherwise.} \end{cases} \tag{59}$$

For each string of unobservable events, the control projection records the set of controllable events that occur along that string. If two strings contain the same sequence of observable events and the same sets of unobservable controllable events between pairs of observable events, it is not possible to choose a policy that distinguishes between these two traces.

3.3.2 Formulation of σ-fields

The maximal σ-fields are defined with respect to the specification K and not the language generated by the automaton. For $n = 0...T$, we define

$$X_n = \left\{ s \in CP_L^{-1}[CP(K)] : \max_{t \in P(K)/P(s)} \|t\| = n \right\}, \tag{60}$$

and we define the sequence of σ-fields as follows:

$$\mathcal{F}_t = \sigma\left(\cup_{n=0}^{t} X_n\right). \tag{61}$$

As t increases, each σ-field $\mathcal{F}_t$ is generated by a larger set of traces in $L(G)$. Therefore $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots \subseteq \mathcal{F}_T$.

The untimed case differs from the timed case in that strings in the information state are not extended to stage t. In the untimed case, it is necessary to know what the most recent observed event was to choose a new observation action, instead of consulting the clock.

In return for increasing the complexity of the filtration (as opposed to the timed case), we do not need to define distinct cost functions and information state transition functions for each stage t.

3.3.3 Example

Figure 4 shows an automaton marking the specification $K = \{ac_1b, ac_2b, c\}$ where $\Sigma_o = \{a, b, c\}$ and all events are controllable. We construct σ-fields using the specification only; since information states outside the specification are certainly undesirable, we will need not to calculate costs for such states.

To construct the σ-field, we partition $\overline{K}$ into three generating sets:

$$X_0 = \{ac_1b, ac_2b, c\} \tag{62}$$

$$X_1 = \{a\} \tag{63}$$

$$X_2 = \{\epsilon\}. \tag{64}$$

Since c_1 and c_2 are both controllable, the two elements of the set X_0 can be distinguished even though they have the same projection onto Σ_o; since we could choose to disable c_1 but not c_2 or vice versa, if the string ab is observed we may be able to know which one of the two strings in X_0 actually occurred. Because we can distinguish between these two strings as a result of our control actions, they have distinct control projections.

The σ-fields are generated from these sets as follows:

$$\mathcal{F}_0 = \sigma\left(X_0\right) \tag{65}$$

$$\mathcal{F}_1 = \sigma\left(X_0, X_1\right) \tag{66}$$

$$\mathcal{F}_2 = \sigma\left(X_0, X_1, X_2\right). \tag{67}$$

Since the generating sets of the σ-fields increase, the σ-fields are nested as $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \mathcal{F}_2$. The elements of these σ-fields are enumerated in Table 4.

3.4 Assigning costs to information states

In order to find an optimal control-observation policy, we first must determine which information states we wish to avoid at any (finite) cost. Since our objective is to achieve a given specification, we wish to avoid allowing the possibility that the system has executed a trace that lies outside the specification. Therefore we initialize our assignment of costs by assigning an infinite cost to these illegal information states π:

$$\hat{V}(\pi) = \infty \text{ if } \pi \not\subseteq \overline{K}. \tag{68}$$

In general, our specification K may not be prefix-closed; therefore we not only need to ensure that we do not allow the system behavior to exceed K—we also need to ensure that we do not select a control action that allows the system to reach a deadlock state before the specification is achieved. To disallow policies that could

Fig. 4 An automaton for illustration the use of active acquisition to find an optimal control specification

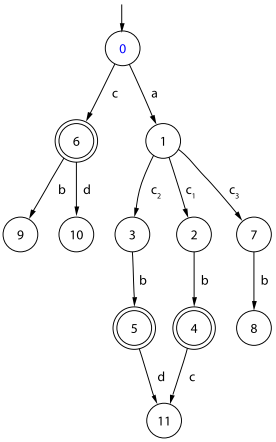


Table 4 Elements of the generated σ-fields $\mathcal{F}_n$

$\mathcal{F}_0$	$\mathcal{F}_1 - \mathcal{F}_0$	$\mathcal{F}_2 - \mathcal{F}_1$
$\emptyset$	a	ϵ
ac_1b	$a + ac_1b$	$\epsilon + ac_1b$
ac_2b	$a + ac_2b$	$\epsilon + ac_2b$
c	$a + c$	$\epsilon + c$
$ac_1b + c$	$a + ac_1b + c$	$\epsilon + ac_1b + c$
$ac_2b + c$	$a + ac_2b + c$	$\epsilon + ac_2b + c$
$ac_1b + ac_2b$	$a + ac_1b + ac_2b$	$\epsilon + ac_1b + ac_2b$
$ac_1b + ac_2b + c$	$a + ac_1b$ $+ ac_2b + c$	$\epsilon + ac_1b + ac_2b + c$
		$\epsilon + a$
		$\epsilon + a + ac_1b$
		$\epsilon + a + ac_2b$
		$\epsilon + a + c$
		$\epsilon + a + ac_1b + c$
		$\epsilon + a + ac_2b + c$
		$\epsilon + a + ac_1b + ac_2b$
		$\epsilon + a + ac_1b + ac_2b + c$

potentially deadlock, we introduce the *stopping cost function* $\hat{V}_s$, defined as follows:

$$\hat{V}_s(\pi) = \begin{cases} 0 & \text{if } \pi \subseteq K \\ \infty & \text{otherwise.} \end{cases} \tag{69}$$

Note that $\emptyset \subseteq K$ and thus $\hat{V}_s(\emptyset) = 0$. If an information state is in $\overline{K}$ but not K, its stopping cost will be infinite since we do not want the system to terminate in such a state; however, its cost function $V(\pi)$ could be finite as there may exist a policy that reaches π on its way to achieving K. Thus we cannot determine $V(\pi)$ for these information states in advance. V_s is defined for all possible sublanguages of $\overline{K}$, even those that do not appear in the sequence of σ-fields.

If certain events can be disabled, the information state transition depends on which events have been disabled as well as which events have been observed. Thus for any $u \in 2^{\Sigma_{co}} \times 2^{\Sigma_c}$, the information state transition function $\hat{\hat{\delta}}_u : \mathcal{F}_t \times \Sigma_{u,obs} \cup \epsilon \to \mathcal{F}_{t+1}$ at stage t, t=0…T-1 is defined as

$$\hat{\hat{\delta}}_u(\pi, \sigma) = \left\{ st\sigma : s \in \pi \wedge t \in (\Sigma_{u,\text{unobs}} \cap \Sigma_{u,\text{enabled}})^* \right\} \tag{70}$$

$$\hat{\hat{\delta}}_u(\pi, \epsilon) = \left\{ st : s \in \pi \wedge t \in (\Sigma_{u,\text{unobs}} \cap \Sigma_{u,\text{enabled}})^* \wedge \Gamma(\delta(x_0, st)) \subseteq \Sigma_{u,\text{disabled}} \right\}. \tag{71}$$

The structure of the information state transition function allows us to quickly conclude that many of the information states in $\mathcal{F}_1, \ldots, \mathcal{F}_T$ are unreachable as it is not possible for a reachable information state to contain two strings where one is a proper prefix of another. We will call an information state π where no string in π is a proper prefix of another string in π an *antichain*.

Theorem 5 *All reachable information states are antichains.*

Proof Suppose that an information state $\pi = s^1 + s^2 + \cdots + s^n$ is an antichain. Then the next observation will either be ϵ or an observable event σ. If ϵ is observed, the new information state must be an antichain because if there existed t^i, t^j in $\hat{\delta}_u(\pi, \epsilon)$ such that t^i were a prefix of t^j, then there would be an event enabled after t^i, which cannot be the case as a result of Eq. 71.

If an observable event σ is observed, the new information state will be of the form $\hat{\delta}_u(\pi, \sigma) = s^1u^{1,1}\sigma + s^1u^{1,2}\sigma + \cdots + s^1u^{1,k_1}\sigma + \cdots + s^nu^{n,1}\sigma + \cdots + s^nu^{n,k_n}\sigma$, where each $u^{i,j} \in \Sigma_{\mathrm{uo}}^*$ and where $u^{i,j} \neq u^{i,k}$ if $j \neq k$.

Since the continuations $u^{i,j}\sigma$ each contain the event σ exactly once, no continuation can be a proper prefix of an another because if that were the case σ would have to appear twice in a continuation. Furthermore, since by assumption there do not exist s^i, s^j such that s^i is a proper prefix of s^j, no continuations $u^{i,k_1}\sigma$, $u^{j,k_2}\sigma$ can exist such that $s^iu^{i,k_1}\sigma$ is a prefix of $s^ju^{j,k_2}\sigma$, as that would require either s^i to be a prefix of s^j or vice versa.

Since the initial information state $\pi_0 = \{\epsilon\}$ contains no prefixes, by induction, all reachable information states are antichains. □

As a result of this theorem, within the context of the example in Section 3.3.3, we can eliminate most information states in Table 4 and solve the dynamic programming equations only for those information states that are also antichains, provided that a solution indeed exists for the particular system under consideration. In the set $\mathcal{F}_2 - \mathcal{F}_0$, the only information states that are not determined to be unreachable by the above theorem are ϵ, a, and $a + c$.

3.4.1 Size of the space of information states

Since each information state corresponds to an antichain, we can determine the size of the space of information states by counting the number of antichains in the automaton. If $k = \|\Sigma_{\mathrm{o}} \cup \epsilon\| = \|\Sigma_{\mathrm{o}}\| + 1$, the number of information states can be no more than the number of antichains in a k-ary tree of depth T. That bound can be computed using the recursion

$$N(t) = (1 + N(t-1))^k, \tag{72}$$

with the initial condition $N(1) = 1$. The solution to this recursion is $O\big(2^{k^T}\big)$, doubly exponential with respect to T. In the timed case, the number of information states at the time horizon T is $O\big(2^{k^T}\big)$ also; therefore, the size of the reachable set of information states is of the same order even when the σ-fields are expanded in the untimed case.

3.5 Solution existence

As in the case of acyclic, timed automata, we require conditions under which there exists an optimal solution to Problem C. The required conditions relate to the controllability and observation of the desired specification K.

Theorem 6 *An optimal control policy exists if and only if the specification K is controllable and observable with respect to Σ_{o}, Σ_c, and $\mathcal{L}(G)$.*

Proof (Sufficiency) Suppose K is controllable and observable with respect to Σ_o, Σ_c, and $\mathcal{L}(G)$. Then K is achieved by the policy where g where

$$g(\pi) = \left(\Sigma_o, \Sigma_c / \left\{\sigma \in \Sigma_c : \exists s\sigma \in \overline{K}(s \in \pi)\right\}\right). \tag{73}$$

(Necessity) Now suppose K is uncontrollable. Then there exists $s \in \overline{K}$ and $\sigma \in \Sigma_{uc}$ such that $s\sigma \in \mathcal{L}(G)$ and $s\sigma \notin \overline{K}$. Consider any policy g. If g is to realize K, g must enable the string s; let $\pi_{g,s}$ denote the information state reached by implementing policy g along s. Since σ is feasible from this information state and takes the system outside K, it must be disabled; but since $\sigma \in \Sigma_{uc}$, it cannot be. Therefore there is no feasible action at the reachable information state $\pi_{g,s}$, and thus no policy g can realize K.

Now suppose K is unobservable. Then there exist $s_1, s_2 \in \overline{K}$ and $\sigma \in \Sigma_c$ such that $P(s_1) = P(s_2), s_1\sigma \notin \overline{K}$ and $s_2\sigma \in \overline{K}$. Consider any policy g. If g is to realize K, g must enable s_1; since $P(s_1) = P(s_2)$, the information state reach by implementing g along s_1 must also contain s_2. Since $s_1\sigma \notin \overline{K}$, enabling σ is not admissible since it would allow an information state not in $\overline{K}$; however, since $s_2\sigma \in \overline{K}$, disabling σ would not allow a string in K to be realized. Therefore there is no admissible action at this information state and thus no policy that realizes K. □

3.6 Dynamic programming equations

For each information state in $\mathcal{F}_0$, we need to disable all feasible events and thus no observations will be possible and no sensors need to be activated. The cost of each information state in $\mathcal{F}_0$ is given by:

$$V(\pi) = \sum_{\sigma \in \Gamma(\pi)} \kappa(\sigma), \tag{74}$$

where $\Gamma(\pi)$ denotes the active event set of π, that is, the set of events that are feasible following any string in π.

An optimal policy can then be calculated for information states π not in $\mathcal{F}_0$ by solving the following dynamic programming equation.

$$V(\pi) = \min_{u \in 2^{\Sigma_c \times \Sigma_{co}}} \left\{ c_u + \max_{\sigma \in \Sigma} \left[V\left(\hat{\delta}_u(\pi, \sigma)\right), V_s\left(\hat{\delta}_{g(\pi)}(\pi, \epsilon)\right) \right] \right\}, \pi \in \mathcal{F}_T. \tag{75}$$

The reverse filtration of σ-fields indicates the order in which we must determine the costs for various information states. The costs of information states in $\mathcal{F}_0$ depend only on the costs of information states that are pre-assigned. The costs of information states in $\mathcal{F}_1 - \mathcal{F}_0$ depend on the pre-assigned costs and the costs calculated for elements of $\mathcal{F}_0$. We therefore solve the dynamic programming equation first for elements in $\mathcal{F}_0$, then for elements in $\mathcal{F}_1 - \mathcal{F}_0$, and so on, and finally for the elements in $\mathcal{F}_T - \mathcal{F}_{T-1}$.

3.7 Example computation on an optimal control policy

Using the dynamic program from the previous subsection, we calculate an optimal control policy for the automaton in Fig. 4. The form of the resulting calculations is very similar to that used for timed automata in the previous section.

Table 5 An optimal control and observation policy for the automaton in Fig. 4

$\mathcal{F}_2 - \mathcal{F}_1$	Disable	Observe	$\mathcal{F}_1 - \mathcal{F}_0$	Disable	Observe	$\mathcal{F}_0$	Disable
ϵ	$\{b, d\}$	a	a	$\{\sigma_{c_3}\}$	b	c	$\{b, d\}$
			$a + c$	—	—	ac_1b	$\{c\}$
						ac_2b	$\{d\}$
						$ac_1b + ac_2b$	$\{c, d\}$
						$ac_1b + c$	$\{b, c, d\}$
						$ac_2b + c$	$\{b, d\}$
						$ac_1b + ac_2b + c$	$\{b, c, d\}$

Table 5 shows an optimal control policy for the automaton in Fig. 4. For each information state in $\mathcal{F}_0$, we need not observe any events as our choice of control action will disable all events and prevent any further observations. The cost incurred for each information state is the cost of disabling all feasible events.

Once these costs have been calculated, we determine the costs for those reachable information states in $\mathcal{F}_1 - \mathcal{F}_0$. The information state $a + c$ is illegal for the following reason. If the event b is enabled, then if the string executed by the system is c, the trace cb outside the specification is enabled and we occur an infinite penalty; however, if we disable b and the true string is a, then the system will deadlock and we will incur an infinite stopping penalty.

For the information states a, the choice of control and observation action is not unique as we may choose to observe b and thus not need to disable c and d, or we could not observe b and then be required to disable c and d. We calculate the cost of the information state a using the following equation:

$$V(a) = \min_{u \in 2^{\Sigma_c \times \Sigma_{co}}} \left\{ c_u + \max_{\sigma \in \Sigma} \left[V\big(\hat{\delta}_u(a, \sigma)\big), V_s\big(\hat{\delta}_u(a, \epsilon)\big) \right] \right\}. \tag{76}$$

3.8 Active acquisition for diagnosis

The development of the diagnosis problem is simpler than that of the control problem as the inputless nature of a system under diagnosis simplifies the information structure. An observation policy is a function $g : \mathcal{F}_T \to 2^{\Sigma_{co}}$, where $g(\pi)$ indicates which events should be observed given that the information available to the policy maker is the information state π. No control actions need to be determined.

In the absence of controllable events, the control projection of a string contains no more information than the standard projection, as the control alphabet reduces to the singleton symbol $1_\emptyset$. Therefore we define σ-fields using the standard projection. For $n = 0...T$, we define

$$X_n = \left\{ s \in P_L^{-1}[P(K)] : \max_{t \in P(K)/P(s)} \|t\| = n \right\}, \tag{77}$$

and we define the sequence of σ-fields as follows:

$$\mathcal{F}_t = \sigma\big(\cup_{n=0}^{t} X_n \big). \tag{78}$$

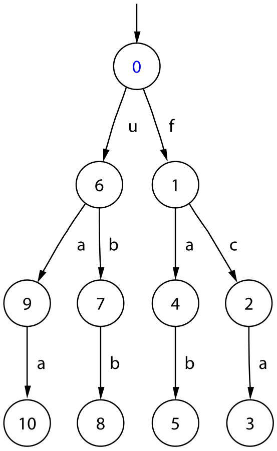


Fig. 5 An automaton used to illustrate the active acquisition method for diagnosis for acyclic, untimed systems. This automaton is identical to that shown in Fig. 2, except that there is no longer a fixed amount of time between events. $\Sigma_{\text{uo}} = \{f, u\}$, $\Sigma_{\text{co}} = \{a, b, c\}$, $\Sigma_{\text{f}} = \{f\}$, and $T = 3$

Since no events can be disabled, the information state transition function can be reduced to $\hat{\hat{\delta}}_u : \mathcal{F}_t \times \Sigma_{u,\text{obs}} \cup \epsilon \to \mathcal{F}_{t+1}$ by:

$$\hat{\hat{\delta}}_u(\pi, \sigma) = \{st\sigma : s \in \pi \wedge t \in (\Sigma_{u,\text{unobs}})^*\} \tag{79}$$

$$\hat{\hat{\delta}}_u(\pi, \epsilon) = \{st : s \in \pi \wedge t \in \Sigma^*_{u,\text{unobs}} \wedge \Gamma(\delta(x_0, st)) = \emptyset\}. \tag{80}$$

The main difference in the active acquisition technique between the control and diagnosis problems is in how illegal information states are assigned. In the diagnosis problem, an information state is illegal it indicates that we are uncertain as to whether or not a failure has occurred when the process has terminated. We thus define a cost for all $\pi \in \mathcal{F}_0$ as follows.

$$V(\pi) = \begin{cases} 0 & \text{if } \pi \text{ is F-certain} \\ \infty & \text{otherwise.} \end{cases} \tag{81}$$

The dynamic programming equation is as before is a simplified version of Eq. 75:

$$V(\pi) = \min_{u \in 2^{\Sigma_{\text{co}}}} \left\{ c_u + \max_{\sigma \in \Sigma \cup \epsilon} V\big(\hat{\delta}_u(\pi, \sigma)\big) \right\}, \pi \in \mathcal{F}_T. \tag{82}$$

Since no events are controllable in the diagnosis problem, the system cannot stop as the result of any action we choose to implement. Because we only stop when the system terminates, we do not need to consider an additional penalty for stopping too soon as we do in the case of supervisory control. The costs $V(\pi)$ for various information states $\pi \in \mathcal{F}_T$ are determined by the method describe in Section 3.6.

3.9 Diagnosis example

Consider the automaton in Fig. 5. This automaton has the same structure as that of Fig. 2, except we no longer assume that events occur at "ticks of the clock."

To define the generating sets $\{X_n\}$, we start with those traces where no continuation is possible; for this example those traces are *ac* and *bd*, so

$$X_0 = \{uaa, \ fab, ubb, \ fca\}. \tag{83}$$

Similarly, X_1 consists of those traces where one more observation is possible, and X_2 is the set of traces where exactly two more observations are possible. The sets X_0 through X_2 are enumerated in Table 6. Note that these partitions are equal to the sets used to generate the σ-fields in the untimed case.

These sets are used to generate the reverse filtration of σ-fields used for the dynamic program. Using Eq. 78, the filtration σ-fields $\mathcal{F}_0 \ldots \mathcal{F}_2$ are created. The reachable elements of these σ-fields are enumerated in Table 7.

Note that while our σ-field $\mathcal{F}_0$ is identical to the σ-field $\mathcal{F}_3$ from the timed example, the set of reachable elements in $\mathcal{F}_2 - \mathcal{F}_1$ is much larger than any of the timed σ-fields.

Table 6 Construction of the field generating sets X_n

X_2	X_1	X_0
ϵ	$fa + ua$	fab
	ub	uaa
	fc	ubb
		fca

Table 7 Reachable elements of the generated σ-fields $\mathcal{F}_n$

$\mathcal{F}_2 - \mathcal{F}_1$	$g^*(\pi)$	$V(\pi)$	$\mathcal{F}_1 - \mathcal{F}_0$	$g^*(\pi)$	$V(\pi)$	$\mathcal{F}_0$	$V(\pi)$
ϵ	$\{b, c\}$	4	$ua + fa$	$\{b\}$	1	fab	0
			ub	$\emptyset$	0	uaa	0
			fc	$\emptyset$	0	ubb	0
			$ua + fa + ub$	—	∞	fca	0
			$ua + fa + ubb$	$\{b\}$	1	$fab + uaa$	∞
			$ua + fa + fc$	—	∞	$fab + ubb$	∞
			$ua + fa + fca$	$\{a\}$	4	$fab + fca$	0
			$ub + fab$	$\{b\}$	1	$uaa + ubb$	0
			$ub + uaa$	$\emptyset$	0	$uaa + fca$	∞
			$ub + fc$	$\{b\}$	1	$ubb + fca$	∞
			$ub + fca$	$\{b\}$	1	$fab + uaa + ubb$	∞
			$fc + fab$	$\emptyset$	0	$fab + uaa + fca$	∞
			$fc + uaa$	$\{a\}$	4	$fab + ubb + fca$	∞
			$fc + ubb$	$\{a\}$	4	$uaa + ubb + fca$	∞
			$ua + fa + ub + fc$	—	∞	$fab + uaa + ubb + fca$	∞
			$ua + fa + ub + fca$	—	∞		
			$ua + fa + ubb + fc$	—	∞		
			$ua + fa + ubb + fca$	—	∞		
			$ub + fab + uaa$	—	∞		
			$ub + fab + fc$	$\{b\}$	1		
			$ub + uaa + fc$	$\{a\}$	4		
			$ub + fab + fca$	$\{b\}$	1		
			$ub + uaa + fca$	—	∞		
			$fc + fab + uaa$	—	∞		
			$fc + fab + ubb$	—	∞		
			$fc + uaa + ubb$	$\{a\}$	4		
			$ub + fc + uaa + fab$	—	∞		
			$ub + fca + uaa + fab$	—	∞		
			$fc + uaa + fab + ubb$	—	∞		

To determine an optimal policy, we simply solve Eq. 82 for the untimed case just as we solved Eq. 21 in the timed case.

4 Cyclic automata

In this section we remove the assumption that our automata must be acyclic and consider the general class of automata where neither assumption (A1) nor (A2) need be satisfied; we still require assumption (A3) to hold. The methodology of the previous sections cannot be directly applied in this case because constructing σ-fields in the same manner for cyclic automata would results in an infinite sequence of σ-fields, and there would be no set of "final" information states from which to initialize a dynamic programming solution. We present two methods of working around this problem: we describe how the set of string-based information states can be reduced to a finite set of "diagnoser states;" we also demonstrate how limited lookahead methods as described in Section 2.6 can be applied in the cyclic case.

4.1 Problem formulation

We formulate the active acquisition of information problem for diagnosis of cyclic systems in a manner similar to previous sections. For ease of notation we restrict attention to the case where there is only one failure type, although these results can be extended to the case of multiple failure types.[1]

As in the acyclic untimed case, we define an observation policy

$$g : \mathcal{F}_\infty \to 2^{\Sigma_{co}} \tag{84}$$

where

$$\mathcal{F}_\infty = \lim_{T\to\infty} \mathcal{F}_T = 2^{P_L^{-1}[P(\mathcal{L}(G))]}, \tag{85}$$

and $\mathcal{F}_T$ is the space of information states for the diagnosis problem of an acyclic untimed automaton where the maximum string length is T. The σ-field $\mathcal{F}_\infty$ is the space of information states for the diagnosis problem under consideration. Thus if $\pi \in \mathcal{F}_\infty$ is the information available to the policy maker at a certain stage, then $g(\pi) \in 2^{\Sigma_{co}}$ specifies the set of costly observable events that must be observed at that stage. Since the automaton is cyclic, there may be information states $\pi \in \mathcal{F}_\infty$ generated from arbitrarily long sequences of observations. In Section 4.2 we show how to compress the set of possible information states so as to ensure that the domain of an observation policy is finite.

To proceed with the formulation of the problem of active acquisition of information for the diagnosis of cyclic untimed automata, we need to introduce Definitions 11–16 which are the extensions of Definitions 4–6 for acyclic timed automata.

Definition 11 An information state π is *F-certain* if $f \in s$ for all $s \in \pi$.

[1] In the case of multiple failure types, Definition 11 can be written as: an information state π is l-safe if $f \in l \Rightarrow f \in s$ for all $s \in \pi$ and $f \notin l \Rightarrow f \notin s \wedge f \notin L/s$ for all $s \in \pi$.

Definition 12 An information state π is *N-safe* if $f \notin s$ and $f \notin L/s$ for all $s \in \pi$.

Definition 13 An information state π is *safe* if π is F-certain or N-safe.

These definitions can be illustrated using Fig. 3. In the language generated by the system in the figure, the information state $\pi_2 = fab + fcad$ is F-certain, as each string in π_2 contains the failure event f. The information state $\pi_2 = ubb + uaa$ is N-safe because not only does no string in π_2 contain a failure event, no failure event is possible following any string in π_2. Both information states π_1 and π_2 are safe.

If the system is in a safe information state, we need not make any more observations since we are certain as to the failure mode in the current information state and in all future state. If the information state is not safe, we must choose an action that ensures that another event will eventually be observed or else we may never diagnose the failure.

Definition 14 An information state π is *non-diagnosable* if $\exists M \in \mathbb{N}$ such that for all $n \geq M, \exists t \in L/\pi$ such that $\|t\| = n$ and the information state obtained by implementing any policy g along t is uncertain.

Definition 15 A language $\mathcal{L}(G)$ is *diagnosed* by an observation policy g if, for all $s \in \mathcal{L}(G)$, the information state reached by implementing g along s is never non-diagnosable.

Definition 16 Let H denote the set of all policies that diagnose $\mathcal{L}(G)$. The language $\mathcal{L}(G)$ is *diagnosable* if H is non-empty, i.e., if there exists a policy that diagnoses $\mathcal{L}(G)$.

Define the performance criterion:

$$J(g) = \sup_{s \in \mathcal{L}(G)} \{c^g(s) + K^g(s)\}, \tag{86}$$

where $c^g(s)$ denotes the cost of implementing policy g along the trajectory s and

$$K^g(s) = \begin{cases} \infty & \text{if the information state } \pi^g(s) \text{ reached by implementing } g \text{ along } s \text{ is} \\ & \text{non-diagnosable} \\ 0 & \text{otherwise.} \end{cases} \tag{87}$$

The performance criterion is thus the maximum total cost of policy g along any string of arbitrary length.

The active acquisition of information problem for diagnosis of cyclic systems, is defined as follows.

Problem CD Find a policy $g^* \in H$ such that

$$J(g^*) = \inf(J(g) | g \in H) \text{ and } J(g^*) < \infty. \tag{88}$$

4.1.1 Solution existence

Without loss of generality, we assume all observable events have a non-zero cost, i.e. $\Sigma_{co} = \Sigma_o$. Solution existence for Problem CD can be determined using the following definition and theorem.

Definition 17 A language $\mathcal{L}(G)$ is strictly logically diagnosable with respect to Σ_o and Σ_f if:

$$(\exists N \in \mathbb{N}) \left(n > N \Rightarrow \hat{D}^N(s) = 1 \vee D^F(s) = 1 \right), \tag{89}$$

where the functions $\hat{D}^N$ and D^F are defined as:

$$\hat{D}^N(s) = \begin{cases} 1 & \text{if } P_L[P^{-1}(s)] \text{ is } N\text{-safe} \\ 0 & \text{otherwise,} \end{cases} \tag{90}$$

$$D^F(s) = \begin{cases} 1 & \text{if } P_L[P^{-1}(s)] \text{ is } F\text{-certain} \\ 0 & \text{otherwise.} \end{cases} \tag{91}$$

Strict logical diagnosability is a more stringent condition than the logical diagnosability of Sampath et al. (1995). A system is strictly logically diagnosable if when all possible events are observed, the system will surely transition to a safe information state after a bounded number of states. Thus, a finite-cost observation policy that diagnoses the failure can be easily determined: we must observe all possible events until N events have been observed; at that point the system will be in a safe information state and no more observations will have to be made at all. In the next theorem we show that strict logical diagnosability is a necessary and sufficient condition for the existence of a solution to Problem CD.

Theorem 7 *A finite-cost observation policy exists if and only if $\mathcal{L}(G)$ is strictly logically diagnosable with respect to Σ_o and Σ_f.*

Proof (Sufficiency) Suppose $\mathcal{L}(G)$ is strictly logically diagnosable with respect to Σ_o and Σ_f. Consider the following observation policy g. For any $s \in \mathcal{L}(G)$ let $\pi \in \mathcal{F}_\infty$ be the information state resulting when g is implemented along s. If $\|s\| \leq N$ set $g(\pi) = \Sigma_o$; otherwise set $g(\pi) = \emptyset$. The cost of this policy is no greater than $Nc(\Sigma_{co})$, and thus a finite-cost observation policy exists.

(Necessity) Now suppose that $\mathcal{L}(G)$ is not strictly logically diagnosable with respect to Σ_o and Σ_f. Then for all $m \in \mathbb{N}$, $\exists s \in \mathcal{L}(G)$ such that $\|s\| = m$ and neither $\hat{D}^N(s) = 1$ nor $D^F(s) = 1$. Suppose the system executes such a string s of arbitrary length. The initial observation action must contain at least one event along s; suppose that σ_1 is the first such event. The information state reached after σ_1 is observed must be unsafe since neither $\hat{D}^N(s) = 1$ nor $D^F(s) = 1$. Therefore we must choose an observation action that contain at least one more event along s. However, after the observation of the second event σ_2, we must remain in an unsafe state. Since this process can be repeated indefinitely without making a diagnosis, there must be an infinite cost along s, and thus there is no finite-cost observation policy. □

The condition of strict logical diagnosability is too severe for most problems as it disallows the possibility of the system running in a normal, "unsafe," state for an indefinitely long time. Were a system to run in such a state indefinitely, a diagnosis at infinite observation cost would be incurred; however, the number of events required to occur for this cost to be incurred would also be infinite. Therefore, it would be more realistic to find a criterion for solution existence coinciding with the concept of diagnosability introduced in Sampath et al. (1995). The definition of diagnosability is given below.

Definition 18 A language $\mathcal{L}(G)$ is *logically diagnosable* with respect to Σ_o, Σ_f if

$$(\exists n \in \mathbb{N})[\forall s \in \Psi(\Sigma_{f_i})](\forall t \in L/s)[\|t\| \geq n \Rightarrow D^F(st) = 1]. \tag{92}$$

To formulate the diagnosis problem so that finite-cost solution existence corresponds to logical diagnosability, consider a performance criterion where future costs are discounted at a rate $\beta < 1$:

$$J_\beta(g) = \left\{ \max_{s \in \mathcal{L}(G)} \sum_{t=0}^{\|s\|} \beta^t c_t^g(s) \right\}. \tag{93}$$

The discounted active acquisition of information problem for diagnosis of cyclic systems is defined as follows.

Problem DCD Find a policy $g^* \in H$ such that

$$J_\beta(g^*) = \inf(J_\beta(g) | g \in H). \tag{94}$$

The conditions for the existence of a solution to Problem DCD coincide with Definition 18.

Theorem 8 *A language $\mathcal{L}(G)$ is diagnosable at finite discounted cost if and only if it is logically diagnosable with respect to Σ_o and Σ_f.*

Proof (Sufficiency) Sufficiency will be shown by contradiction. Suppose $\mathcal{L}(G)$ is logically diagnosable. Consider the observation policy g defined by $g(\pi) = \Sigma_o$ for all $\pi \in \mathcal{F}_\infty$; the cost of this policy is $\sum_{\sigma \in \Sigma_o} \frac{\nu(\sigma)}{1-\beta}$. The information state reached by implementing g along any s is $\pi(s) = P_L^{-1}[P(s)]$. Suppose $\pi(s)$ were non-diagnosable. Then by Definition 18, there exists $N \in \mathbb{N}$ such that there also exists $t \in L/s$ such that $\|t\| = N$ and $P_L^{-1}[P(st)]$ is uncertain; therefore, at least one string in $P_L^{-1}[P(st)]$ contains a failure event.

Furthermore, for all $n \in \mathbb{N}$ there exists $u \in L/st$ such that $\|u\| = n$ and $P_L^{-1}[P(stu)]$ in uncertain. Therefore, a continuation u of arbitrary length can be appended to the failure event, thus contradicting the assumption of logical diagnosability. Therefore, no such $\pi(s)$ can be reached, and thus $\mathcal{L}(G)$ is diagnosable at finite cost.

(Necessity) If $\mathcal{L}(G)$ is not logically diagnosable, $\exists s \in \mathcal{L}(G)$ such that for all $n \in \mathbb{N}$, there exists $t \in L/s$ such that $\|t\| > n$ and $P_L^{-1}[P(st)]$ is uncertain. The information state reached by implementing any policy along st must contain at least all members of the set $P_L^{-1}[P(st)]$ and therefore must be uncertain. Therefore, the original information state $\pi_0 = \epsilon$ is non-diagnosable by Definition 14; since this initial information state is reachable under every policy, $\mathcal{L}(G)$ must be non-diagnosable. □

4.2 Solution methods

A cyclic automaton generates an infinite number of *string-based* information states in the σ-field $2^{P_L^{-1}[P(\mathcal{L}(G))]}$. In order to derive an optimal policy in the same way in the case of acyclic automata, we reduce the string-based information states to diagnoser states, as the set of diagnoser states is guaranteed to be finite (Sampath et al. 1995).

Recall from Sampath et al. (1995) that a diagnoser state is an element of the set $Q_o = 2^{X_o \times \Delta}$, where X_o is the set of states of the system reachable via an observable event and Δ is the set of failure labels indicating which failure events may have occurred in the system.

For each element in $2^{\mathcal{L}(G)}$, the diagnoser state associated with that element can be computed by the function $q : 2^{\mathcal{L}(G)} \rightarrow Q_o$

$$q(\pi) = \bigcup_{s \in \pi} (\delta(x_o, s),\, LP(x_0, s)). \tag{95}$$

This mapping allows the infinite set of string-based elements in $2^{\mathcal{L}(G)}$ to be reduced to a finite set of diagnoser states, or *state-based information states*. We can calculate optimal policies using diagnoser states instead of string-based elements as a result of the following theorem.

Theorem 9 *If multiple information states in $\mathcal{F}_\infty$. map to the same diagnoser state, the same sequence of observation actions is optimal for any string after that diagnoser state.*

Proof Let π_1 be a string-based information state such that $q(\pi_1) = q_1$. Suppose we implement the action $u \in 2^{\Sigma_{co}}$ and the event σ is observed. The new string-based information state will be:

$$\hat{\delta}_u(\pi_1, \sigma) = \{st\sigma : s \in \pi_1 \wedge t \in \Sigma_{u,\text{unobs}}\}. \tag{96}$$

The diagnoser state corresponding to this information state is given by:

$$\begin{aligned}
q\big(\hat{\delta}_u(\pi_1, \sigma)\big) &= \bigcup_{st\sigma \in \hat{\delta}_u(\pi_1, \sigma)} (\delta(x_0, st\sigma),\, LP(x_0, st\sigma)) && (97)\\
&= \bigcup_{st\sigma : s \in \pi_1 \wedge t \in \Sigma_{u,\text{unobs}}} (\delta(x_0, st\sigma),\, LP(x_0, st\sigma)) && (98)\\
&= \bigcup_{s \in \pi_1} \bigcup_{t\sigma : t \in L/s \cap \Sigma_{u,\text{unobs}}} [\delta(\delta(x_0, s), t\sigma),\, LP(LP(x_0, s), t\sigma] && (99)\\
&= \bigcup_{(x,l) \in q_1} \bigcup_{t\sigma : L_x(G) \cap \Sigma_{u,unobs}} (\delta(x, t\sigma),\, LP(x, t\sigma)), && (100)
\end{aligned}$$

where $L_x(G)$ denotes the language generated by G starting from the state x.

This expression indicates that if the current diagnoser state q_1 is known, the succeeding diagnoser state depends only on the action u, not on the string-based information state π_1.

Now suppose π_2 is a string-based information state distinct from π_1, but $q(\pi_2) = q_1$. If we implement the action u at the information state π_2, the resulting diagnoser state will be the same as it would be if we had implemented that action at π_1.

Now consider a sequence $u^* := (u_1^*, u_2^*, \ldots)$ of optimal actions starting at π_1 along a string t. This sequence of actions will create a sequence of diagnoser states that will reach a safe state when the system is diagnosed. If u^*were not also optimal for π_2 along t, there would be a less expensive sequence u' of actions along t that would generate a sequence of diagnoser states. However, since t must be feasible after both π_1 and π_2 (since they map to the same diagnoser state and hence to the same set of states in the system), u' would also be feasible after π_1, contradicting our statement that the sequence u^* is optimal. Therefore if two string-based information states map to the same diagnoser state, the same set of actions will be optimal for both information states. Furthermore, when determining an optimal policy, we need only consider an optimal action for each diagnoser state, as opposed to each string-based information state. □

The reduction of information states in $\mathcal{F}_\infty$ to diagnoser states ensures that an optimal policy need only be calculated for a finite number of information states; that is, we can describe an optimal observation policy as a function $g : Q_o \to 2^{\Sigma_{co}}$, Q_o being the finite set of potential diagnoser states. However, the reduction of set of strings to diagnoser states sacrifices the sequentiality inherent in the strings; there is no inherent "filtration" of diagnoser states that we can use as we have in the case of acyclic systems.

However, there are certain diagnoser states for which we can assign a cost *a priori* just as we assign costs to information states in the final maximal σ-field $\mathcal{F}_T$ in the case of untimed systems (Section 3.6). If a state is safe, we are sure that no more observations are needed after reaching such a state and can assign zero cost to such a state. Furthermore, we can test all remaining diagnoser states to see if the they are non-diagnosable (Yoo and Lafortune 2002b; Jiang et al. 2001) and assign infinite cost to any non-diagnosable state.

For all $q \in Q_o$, define

$$V(q) = \begin{cases} 0 & \text{if } q \text{ is safe} \\ \infty & \text{if } q \text{ is non-diagnosable.} \end{cases} \tag{101}$$

We state what it means for a state-based information state to be diagnosable in the following definition.

Definition 19 A state-based information state π is diagnosable if the language generated by the automaton $G' = (X \cup x', \Sigma \cup \{f, n\}, \delta', x')$ is diagnosable, where:

$$\delta'(x', f) = x \quad \text{if } (x, F)) \in \pi \tag{102}$$

$$\delta'(x', n) = x \quad \text{if } (x, N)) \in \pi \tag{103}$$

$$\delta'(x, \sigma) = \delta(x, \sigma) \quad \text{if } x \neq x'. \tag{104}$$

In short, a state-based information state π is diagnosable if the original automaton with initial state π instead of x_0 is diagnosable. To apply standard diagnosability results, we append a new initial state to the automaton G and add unobservable transitions to this state that bear the failure labels associated with each component of the diagnoser state.

In order to determine whether a particular diagnoser state has a zero or infinite cost, we need only test these conditions for diagnoser states q_d consisting of at most two components because if q_d is non-diagnosable, any diagnoser state that is a superset of q_d will also be non-diagnosable.

The minimum worst-case costs of the remaining diagnoser states can be determined using the following dynamic programming equations.

$$V(q) = \min_{u \in 2^{\Sigma_{\mathrm{co}}}} \left\{ c_u + \max_{\sigma \in \Sigma_{\mathrm{o}}} V\big(\hat{\delta}_u(q, \sigma)\big) \right\}, q \in Q_o. \tag{105}$$

For Problem DCD, the equation that needs to be solved is given by

$$V(q) = \min_{u \in 2^{\Sigma_{\mathrm{co}}}} \left\{ c_u + \max_{\sigma \in \Sigma_{\mathrm{o}}} \beta^t V\big(\hat{\delta}_u(q, \sigma)\big) \right\}, q \in Q_o, \tag{106}$$

where the exponent t is defined as:

$$t = \min_{w \in \Sigma_{u), \mathrm{unobs}}} \{ \| w\sigma \| : w\sigma \in L/\pi \}. \tag{107}$$

Future costs in Eq. 106 are thus discounted according to the minimum number of events that may have occurred between the current observation and the next observation.

The dynamic programming Eqs. 105–107 that, together with Eq. 101, solve Problems CD and DCD, respectively, are sets of algebraic equations that in general must be solved simultaneously for all diagnoser states. Such solutions appear in the literature as characteristics of the free-time problem in stochastics (cf. Chapter 4 of Kushner 1971).

4.2.1 Example

Figure 6 shows an example of a cyclic automaton, where the costs of each observable event are given by $\nu(a) = 1$, $\nu(b) = 2$, $\nu(c) = 3$, and $\nu(d) = 4$. The results of the two-component diagnosability tests are shown in Table 8.

For example, since the diagnoser state $\{(3, N), (4, F)\}$ has infinite cost, any diagnoser state containing both $(3, N)$ and $(4, F)$ will also have infinite cost. Also, if two diagnoser states have zero cost and the same label, their union will have the same label, e.g. since $\{(3, N), (4, N)\}$ and $\{(8, N), (9, N)\}$ both have zero cost and bear only the label N, the diagnoser state $\{(3, N), (4, N), (8, N), (9, N)\}$ also has zero cost.

Fig. 6 An automaton used to illustrate the active acquisition method for cyclic systems

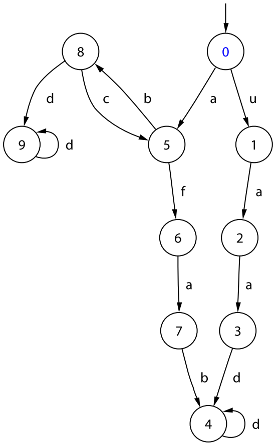


Table 8 Results of diagnosability tests for reachable two-component diagnoser states

	$4F$	$7F$	$0N$	$2N$	$3N$	$4N$	$5N$	$8N$
$7F$	0	—	—	—	—	—	—	—
$0N$	?	?	—	—	—	—	—	—
$2N$	?	?	?	—	—	—	—	—
$3N$	∞	?	?	0	—	—	—	—
$4N$	∞	?	?	0	0	—	—	—
$5N$	?	∞	∞	?	?	?	—	—
$8N$	∞	?	?	0	0	0	?	—
$9N$	∞	?	?	0	0	0	?	0

In Fig. 6, a finite-cost solution exists for Problem DCD but not for Problem CD. To see this, consider the cost of the information state $q = \{(8, N\}$ under Problem DCD.

In the information state $q = \{(8, N\}$, only the two actions $\{c, d\}$ and $\{a, b, d\}$ and actions that are supersets of those actions are admissible in that they prevent the system from entering a non-diagnosable state. The equation to find an optimal action for q is therefore:

$$V(8N) = \min\{c + d + \beta V(5N), a + b + d + \beta^2 V(8N)\}. \tag{108}$$

We now need to consider the cost of the information state $\{5, N\}$. Using the same arguments as above, the only two actions we need to consider at $\{5, N\}$ are $\{a, b\}$ and $\{a, c, d\}$. The equation to find an optimal action at $\{5, N\}$ is:

$$V(5N) = \min\{a + b + \beta V(8N), a + c + d + \beta^2 V(5N)\}. \tag{109}$$

If we solve these equations simultaneously, we find that the optimal action at $\{8, N\}$ is $\{a, b, d\}$ and the optimal action at $\{5, N\}$ is $\{a, b\}$. The cost of the information state $\{8, N\}$ is then:

$$V(8N) = \frac{7}{1 - \beta^2}. \tag{110}$$

If $\beta < 1$, the cost of $\{8, N\}$ is finite. However, if we consider Problem CD, β is equal to exactly one and the cost of diagnosing the failure from this information state becomes infinite. The loop between states 5 and 8 means that it is possible for an arbitrarily large number of observations to be necessary, thus the worst-case undiscounted observation cost must be infinite (Table 9).

Table 9 Calculation of optimal observation actions for the diagnoser states {(5,N)} and {(8,N)}

$g(5N)$	$g(8N)$	$V(5N)$	$V(8N)$
$\{a, b\}$	$\{c, d\}$	$\frac{a+b+\beta(c+d)}{1-\beta^2}$	$\frac{c+d+\beta(a+b)}{1-\beta^2}$
$\{a, b\}$	$\{a, b, d\}$	$a + b + \frac{\beta(a+b+d)}{1-\beta^2}$	$\frac{a+b+d}{1-\beta^2}$
$\{a, c, d\}$	$\{c, d\}$	$\frac{a+c+d}{1-\beta^2}$	$c + d + \frac{\beta(a+c+d)}{1-\beta^2}$
$\{a, c, d\}$	$\{a, b, d\}$	$\frac{a+c+d}{1-\beta^2}$	$\frac{a+b+d}{1-\beta^2}$

4.3 Limited lookahead algorithms for cyclic systems

Another technique to overcome the difficulties inherent in cyclic systems is to consider a limited lookahead method similar to the one proposed in Section 2.6 for acyclic timed automata. By restricting our attention to a finite lookahead horizon, we no longer need to make the switch from string-based to state-based information states, as the limited lookahead ensures that only a finite number of strings are considered at each stage.

However, in applying the limited lookahead method to cyclic systems, we must take note of a fine distinction that did not appear in acyclic automata; namely, the distinction between *preserving the property of diagnosability* and the actual *act of diagnosing the failure*. To see this difference, consider the example in Fig. 7 and suppose that $\Sigma_{\text{fo}} = \{c\}$, $\Sigma_{\text{f}} = \{f\}$, $\Sigma_{\text{co}} = \{a, b\}$, and that the limited lookahead horizon is $T' = 2$. Suppose we apply the limited lookahead algorithm for acyclic automata defined in Eqs. 37–40 without modification. At each stage, the locally optimal action is always to observe only c, as it will always be possible to pay to observe a and b beyond the lookahead horizon. Such a policy ensures that the failure event is always diagnosable, but the actual diagnosis can be put of indefinitely. The "procrastination" characteristic described for the acyclic timed model is no longer held in check by the existence of a final, finite, deadline for diagnosis.

In order to ensure the diagnoses are made in a timely fashion, we consider a surrogate problem wherein we introduce a penalty for the delay in diagnosis occurring in uncertain information states. The delay in diagnosis for an information state π is defined as

$$\text{delay}(\pi) = \max_{s \in \pi}(\|t\| : s = uft). \tag{111}$$

We require the delay penalty function $R : \mathbb{N} \to \mathbb{R}^+$ to have the following properties: (1) R is non-decreasing in $\mathbb{N}$, and (2) $\exists n \in \mathbb{N}$ such that $R(n) \geq c(\Sigma_{\text{co}})$. The first condition ensures that the penalty for delaying a diagnosis increases as the delay increases, while the second ensures that if the diagnosis has been delayed a sufficient length of time, it becomes optimal to make whatever observations are necessary to complete the diagnosis.

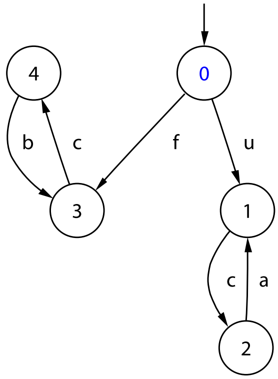


Fig. 7 An automaton where applying the acyclic limited lookahead approach directly results in the failure never being diagnosed

When assigning costs at the lookahead horizon, we consider two cases. In the first case we assign a cost based solely on the diagnosis delay before the horizon. All costs required to make a diagnosis that are incurred beyond the horizon are disregarded, even if all observations beyond the horizon must be made. In this case we assign a zero cost to an information state is if it is diagnosable at the horizon.

In the second case, while determining whether or not the information state is diagnosable, we also determine the worst-case diagnosis delay after the horizon when all observations are made using the method of Yoo and Garcia (2003). By considering the delay after the lookahead horizon, in general we reduce the delay in diagnosis as the penalty function R increases more rapidly.

To construct the maximal σ-fields $\mathcal{F}'_t$ used in the limited lookahead algorithm, we first create the automaton $G_{T'}$ which generates all strings in $\mathcal{L}(G)$ of length T' of less. For $n = 0 ... T'$, we define a sequence of partitions using the method for acyclic untimed automata:

$$X'_n = \left\{ s \in P_L^{-1}[P(G_{T'})] : \max_{t \in P(G_{T'}))/P(s)} \|t\| = n \right\}, \tag{112}$$

and we define the sequence of σ-fields as follows:

$$\mathcal{F}'_t = \sigma\left(\cup_{n=0}^{t} X'_n\right). \tag{113}$$

We then assign a cost to all information states in $\mathcal{F}'_0$:

$$V'(\pi) = \begin{cases} 0 & \text{if } \pi \text{ is F-certain} \\ R(\text{delay}(\pi) + \text{diagdelay}(\pi)) & \text{if } \pi \text{ is diagnosable} \\ \infty & \text{if } \pi \text{ is non-diagnosable.} \end{cases} \tag{114}$$

We then determine the actions for all information states π in $\mathcal{F}'_{T'}$ using the following dynamic program

$$V'(\pi) = R(\text{delay}(\pi)) + \min_{u \in 2^{\Sigma_{co}}} \left\{ c_u + \max_{\sigma \in \Sigma \cup \epsilon} V'\left(\hat{\delta}_u(\pi, \sigma)\right) \right\}, \tag{115}$$

and proceeding as in Section 3.6. Upon solving this equation for $V'(\epsilon)$, we implement the observation action and when an event is observed, we generate a new information state $\hat{\pi}$and then construct a new sequence of σ-fields starting from $\hat{\pi}$.

We construct the automaton $G_{T',\hat{\pi}}$, which generates those strings that are continuations of strings of $\hat{\pi}$ of length T' or less. We then construct the sequence of partitions:

$$X'_{\pi,n} = \left\{ st \in P_L^{-1}[P(G_{T'})] : s \in \hat{\pi} \wedge \max_{t \in P(G_{T'}))/P(s)} \|t\| = n \right\}, \tag{116}$$

We define the sequence of σ-fields

$$\mathcal{F}'_t = \sigma\left(\cup_{n=0}^{t} X'_{\hat{\pi},n}\right), \tag{117}$$

solve the corresponding dynamic program, and so on. Returning to the example in Fig. 7, suppose $\nu(a) = 1.25$, $\nu(b) = 2$, and that the delay penalty function is given by:

$$R(n) = \begin{cases} \frac{n}{3} & n \leq 12 \\ 4 & \text{otherwise.} \end{cases} \tag{118}$$

If we do not consider delay beyond the lookahead horizon, the penalty for an information state of the form $uc(ac)^k + fc(bc)^k$ is

$$R\big(\mathrm{delay}\big(uc(ac)^k + fc(bc)^k\big)\big) = R\big(\|c(bc)^k\|\big) = R(2k+1) = \begin{cases} \frac{2k+1}{3} & k \leq 5 \\ 4 & \text{otherwise.} \end{cases} \tag{119}$$

The action at $\pi = \epsilon$ is still to observe only c, as the penalty for the information state $uc + fc$ is $\frac{1}{3}$. At $\pi = uc + fc$, the action is still to observe only c, as the penalty for the information state $\pi = ucac + fbcb$ is 1, less than the cost of observing either a or b. However, at that information state, the action chosen by the algorithm is to observe a, as the cost of observing a and diagnosing the failure is 1.25, less than the delay cost occurred at the information state $\pi = ucacac + fbcbcb$, which is $\frac{5}{3}$.

If we consider the delay beyond the lookahead horizon, instead of waiting until $\pi = ucac + fbcb$ to observe a and make the diagnosis, the decision to observe a is made at $\pi = uc + fb$. At this point, the cost of delaying the observation is given by

$$R(\mathrm{delay}(ucac + fbcb) + \mathrm{diagdelay}(ucac + fbcb)) = R(3+1) = \frac{4}{3}, \tag{120}$$

as we would need to wait for one event beyond the horizon to make the diagnosis. Thus by considering the delay after the horizon, the diagnosis is made more promptly.

4.4 Problem formulation for stochastic, cyclic automata

In an analogous manner to the section on timed, acyclic automata, we now consider the active acquisition of information problem for stochastic, cyclic automata. An observation policy g and the space of information states for the problem formulated in this section are defined in exactly the same way as in the case of logical cyclic untimed automata. To precisely formulate the diagnosis problem, we restate the definitions for logical, cyclic automata in the stochastic framework. These definitions are conceptually equivalent to Definitions 11–16.

Definition 20 An information state π is *F-certain* if $\Pr(s : f \in s \mid s \in \pi) = 1$.

Definition 21 An information state π is *N-safe* if $\Pr(s : f \notin s \wedge f \notin L/s \mid s \in \pi) = 1$.

Definition 22 An information state π is *safe* if π is F-certain or N-safe.

Definition 23 An information state π is *non-diagnosable* if $\exists N \in \mathbb{N}$ such that for all $n \geq N$, $\exists t \in L/\pi$ such that $\|t\| = n$ and the information state obtained by implementing any policy g along t is uncertain.

Definition 24 A language $\mathcal{L}(G)$ is *surely diagnosed* by an observation policy g if, for all $s \in \mathcal{L}(G)$, the information state reached by implementing g is never non-diagnosable.

Definition 25 Let H denote the set of all policies that diagnose $\mathcal{L}(G)$. The language $\mathcal{L}(G)$ is *surely diagnosable* if H is non-empty, i.e., if there exists a policy that surely diagnoses $\mathcal{L}(G)$.

For stochastic automata we consider the expected cost of an observation policy g instead of the worst-case cost of g. Define the performance criterion:

$$J(g) = E^g \left\{ c^g(s) + K^g(s) \right\}, \tag{121}$$

where $c^g(s)$ denotes the cost of implementing policy g along the trajectory s and $K^g(s)$ is defined in Eq. 87. The performance criterion is thus the expected total cost of policy g.

The active acquisition of information problem, or stochastic cyclic sure diagnosis problem, is defined as follows.

Problem SCSD Find a policy $g^* \in H$ such that

$$J(g^*) = \inf(J(g) | g \in H) < \infty. \tag{122}$$

4.5 Solution existence in the cyclic, stochastic case

As in this section on logical cyclic automata, we assume that all observable events have a non-zero cost of observation; this cost may be arbitrarily close to zero.

Just as in the case of acyclic systems, we first consider conditions necessary and sufficient to ensure that a language can be diagnosed at finite cost. To find such conditions, we consider the previous work on diagnosability of stochastic discrete-event systems (Thorsley and Teneketzis 2005).

4.5.1 Review of stochastic diagnosability

The notions of stochastic diagnosability replace the logically sure statements of the definition of diagnosability for logical automata in Sampath et al. (1995) with probabilistic almost sure statements. Of the two definitions presented, the stricter is A-diagnosability.

Definition 26 (A-diagnosability) A live, prefix-closed language L is A-diagnosable with respect to a projection P and a set of transition probabilities p if

$$\begin{aligned}(\forall \epsilon > 0)(\exists N \in \mathbb{N})(\forall s \in \Psi(\Sigma_{f_i}) \wedge n \geq N)\\ \left\{ \Pr\left(t : D^F(st) = 0 \mid t \in L/s \wedge \|t\| = n\right) < \epsilon \right\},\end{aligned} \tag{123}$$

where the diagnosability condition function D^F is as in Eq. 91.

If a system is A-diagnosable, when a failure occurs, the probability of a continuation that does not allow the failure to be diagnosed approaches zero as the length of the continuation approaches infinity. However, we still need to be logically certain that a failure has occurred in order to call it diagnosed. In the second definition, AA-diagnosability, we weaken the requirement necessary to diagnose a failure.

Definition 27 (AA-Diagnosability) A live, prefix-closed language L is AA-diagnosable with respect to a projection P and a transition probability function p if

$$(\forall \epsilon > 0 \wedge \forall \alpha < 1)(\exists N \in \mathbb{N})$$
$$(\forall s \in \Psi(\Sigma_{f_i}) \wedge n \geq N)\left\{ \Pr\left(t : D_\alpha^F(st) = 0 \mid t \in L/s \wedge \|t\| = n\right) < \epsilon \right\}, \quad (124)$$

where the diagnosability condition function D_α is:

$$D_\alpha^F(st) = \begin{cases} 1 & \text{if } \Pr\left(\omega : \Sigma_{f_i} \in \omega \mid \omega \in P_L^{-1}[P(st)]\right) > \alpha \\ 0 & \text{otherwise.} \end{cases} \quad (125)$$

Thus a system is AA-diagnosable if almost every continuation of a certain length after a failure event leads to a state where we are almost certain that the failure has occurred with probability greater than α, for any α arbitrarily close to, but not equal to, one. Conditions necessary and sufficient to confirm A-diagnosability and sufficient to confirm AA-diagnosability are given in full in Thorsley and Teneketzis (2005); we now highlight a few key points.

The conditions for A- and AA-diagnosability are checked through the construction of a *stochastic diagnoser*. A stochastic diagnoser for a stochastic automaton G is the machine $G_{sd} = (Q_{\text{sd}}, \Sigma_{\text{o}}, \delta_d, q_o, \Phi, \phi_0)$, where

- $Q_{\text{sd}} \subseteq Q_o$ is the set of reachable *logical elements*
- Σ_{o} is the set of observable events in G
- δ_d is the partial transition function of the stochastic diagnoser
- $q_0 = \{x_0, N\}$ is the initial state of the stochastic diagnoser
- Φ is a set of transition probability matrices
- $\phi_0 = [1]$ is the initial probability vector.

The first four elements $(Q_{\text{sd}}, \Sigma_{\text{o}}, \delta_d, q_o)$ of the stochastic diagnoser are the same as in the logical diagnoser described in Section 1.2. The logical diagnoser states are renamed "logical elements" as there are not, in themselves, the states of the stochastic diagnoser. The state of the stochastic diagnoser consists of a logical element and probability distribution among the components of that logical element.

Each logical element consists of a set of *components* of the form (q, x, ℓ), where q denotes the logical element, of the stochastic diagnoser, x denotes the state of the stochastic automaton, and ℓ is a failure label (normal or faulty). The pair (Φ, ϕ_0) allows components to be classified as either transient or recurrent by treating the components as states of a Markov chain. We also define a function $\delta_{\text{comp}} : (Q_{\text{sd}} \times X \times \Delta \times \Sigma) \to (Q_{\text{sd}} \times X \times \Delta)$, which indicates how the component of the stochastic diagnoser that corresponds to the actual state of the original stochastic automaton is updated as events occur. Lastly, we say that a logical element of a stochastic diagnoser is N-safe if every information state $\pi \in \mathcal{F}_\infty$ that such that $q(\pi) = q$ is N-safe.

4.5.2 Strict-A-diagnosability; solution existence for Problem SCSD

In order to find necessary and sufficient conditions for a solution for Problem SCSD to exist, it is necessary to modify the definition of A-diagnosability.

Definition 28 A language L is *strictly-A-diagnosable* if

$$(\forall \epsilon > 0)(\exists N \in \mathbb{N})(\forall n > N)[\Pr(s : D^F(s) = 0 \wedge \hat{D}^N(s) = 0 \mid \|s\| = n) < \epsilon], \quad (126)$$

where the diagnosis condition functions $\hat{D}^N$ and D^F are defined in Eqs. 90 and 91.

This definition is the stochastic analogue to the definition of strict logical diagnosability; it is necessary that, as the number of events that have occurred becomes large, the probability that the system reaches a safe information state approaches one. This definition allows us to state the following theorem.

Theorem 10 *A system is strictly-A-diagnosable if and only if every recurrent component of its associated stochastic diagnoser lies in a certain and safe logical element.*

Proof (Sufficiency) Let $\mathcal{C}$ be the set of components of a stochastic diagnoser, and let $\mathcal{T}_c \in \mathcal{C}$ and $\mathcal{R}_c \in \mathcal{C}$ be the sets of transient and recurrent components, respectively. Suppose that every $q \in Q_d$ that contains a recurrent component (q, x, l_f) such that $\Sigma_{f_i} \in l_f$ is F_i-certain.

By Lemma 1 of Thorsley and Teneketzis (2005), there exists $n \in \mathbb{N}$ such that $\forall c = (q, x, l) \in \mathcal{C}$

$$\Pr(s : \delta_{\text{comp}}(c_0, s) \in \mathcal{T}_c \mid \|s\| = n) < \epsilon. \quad (127)$$

(This lemma states that, in the long run, the stochastic diagnoser will almost surely reach a recurrent component.) Since $\delta(x_0, s)$ is a component of the diagnoser of the system reached by s, this implies that:

$$\Pr(s : \delta_{\text{comp}}(c_0, s) \in \mathcal{R}_c \mid \|s\| = n) > 1 - \epsilon. \quad (128)$$

Therefore, if at least n events have occurred, with probability greater than $1 - \epsilon$, we will reach an element that contains at least one recurrent component.

If the true behavior of the system reaches a recurrent component, then, by assumption, that component is part of a certain and safe logical element. Therefore either $D^F(s) = 1$ or $\hat{D}^N(s) = 1$.

Since the probability of reaching a certain and safe element is at least $1 - \epsilon$,

$$\Pr\left(s : D^F(s) = 1 \vee \hat{D}^N(s) = 1 \mid \|s\| = n\right) > 1 - \epsilon \quad (129)$$

$$\Pr\left(s : D^F(s) = 0 \wedge \hat{D}^N(s) = 0 \mid \|s\| = n\right) < \epsilon. \quad (130)$$

Therefore if every recurrent component lies in a safe, certain logical element, the system is strictly-A-diagnosable.

(Necessity) Necessity will be shown by contradiction. Suppose there exists a string s such that s transitions the system from the initial state to a recurrent component c_R in a logical element that is either certain or unsafe. Let the probability of s be denoted by p_s.

Since c_R is not in a certain, safe logical element, both $D^F(s)$ and $\hat{D}^N(s)$ are equal to zero. Since

$$\Pr\left(t : D^F(st) = 0 \wedge \hat{D}^N(st) = 0 \mid t \in L/s \wedge \|t\| = n - \|s\|\right) = 1, \quad (131)$$

it follows that:

$$\Pr\left(st : D^F(st) = 0 \wedge \hat{D}^N(st) = 0 \mid \|st\| = n\right) = p_s. \tag{132}$$

Since $p_s > 0$, there exists $0 < \epsilon < p_s$ such that

$$\Pr\left(st : D^F(st) = 0 \wedge \hat{D}^N(st) = 0 \mid \|st\| = n\right) > \epsilon. \tag{133}$$

Thus the language is not strictly-A-diagnosable. □

From this theorem, it is clear that strict-A-diagnosability implies A-diagnosability, as the necessary and sufficient condition for A-diagnosability is satisfied whenever the necessary and sufficient condition for strict-A-diagnosability is satisfied. Strict-A-diagnosability is shown to be a necessary and sufficient condition for the existence of a solution to Problem SCSD in the following theorem.

Theorem 11 *A stochastic automaton is diagnosable at finite expected cost if and only if the automaton is strictly-A-diagnosable when all events in Σ_o are observed.*

Proof (Sufficiency) Suppose that L is diagnosable at finite expected cost, and suppose that the smallest observation cost for an observable event is $\gamma > 0$. Let the cost of diagnosing L be denoted by a constant K. Then for all $N \in \mathbb{N}$,

$$\begin{aligned} K = {} & E\left(c^{g^*}(s) \mid \#^{g^*}_{\text{obs}}(s) \leq N\right) \Pr(s : \#_{\text{obs}}(s) \leq N) \\ & + E\left(c^{g^*}(s) \mid \#^{g^*}_{\text{obs}}(s) > N\right) \Pr(s : \#_{\text{obs}}(s) > N), \end{aligned} \tag{134}$$

where $\#^{g^*}_{\text{obs}}(s)$ denotes the number of events observed by the policy g^* along the string s.

The expected cost if more than $N + 1$ total observations are needed to observed is at least $N\gamma$, so

$$K \geq N\gamma \Pr\left(s : \#^{g^*}_{\text{obs}}(s) > N\right) \tag{135}$$

$$\Pr\left(s : \#^{g^*}_{\text{obs}}(s) > N\right) \leq \frac{K}{N\gamma}. \tag{136}$$

Let $\epsilon > 0$. We can then choose N such that $N > \frac{2K}{\epsilon\gamma}$, and thus

$$\Pr\left(s : \#_{g^*,\text{obs}}(s) > N\right) < \frac{\epsilon}{2}. \tag{137}$$

Since g^* is an optimal observation policy, g^* will only call for more than N observations along the string s if the information state generated by g^* along the string s after N observations is not safe. The set of strings s on which more than N observations are made by g^* is equal in probability to the set of strings where N observations are made and no diagnosis has occurred. It follows that

$$\Pr\left(s : D^F(s) = 0 \wedge \hat{D}^N(s) = 0 \wedge \#_{g^*,\text{obs}}(s) = N\right) < \frac{\epsilon}{2}. \tag{138}$$

Now consider the possible number of unobservable events that may occur between two observed events, and suppose that a diagnosis has yet to be made. Since the system is diagnosable at finite cost, another observation must occur with probability one. Therefore, the expected number of events between each pair of observations is finite, and thus for all $\epsilon > 0$, $N \in \mathbb{N}$, there exists $M \in \mathbb{N}$ such that

$$\Pr\left(s : D^F(s) = 0 \wedge \hat{D}^N(s) = 0 \wedge \|u\| \geq M\right) < \frac{\epsilon}{2(N+1)}, \tag{139}$$

where u denotes a sequence of unobservable events between two observable events. The probability that at least one of $N+1$ consecutive sequences of unobservable events is of length no less than M is therefore less than $\frac{\epsilon}{2}$.

For a string of length $N' = M(N+1)$, Eq. 139 indicates that the probability of less than N events are observed in such a string is less than $\frac{\epsilon}{2}$. Eq. 138 indicates that the probability of not having made a diagnosis given that N events are observed is also less than $\frac{\epsilon}{2}$. The probability that either of these conditions is satisfied is thus less than ϵ, and thus

$$\Pr\left(s : D^F(s) = 1 \vee \hat{D}^N(s) = 1 \mid \|s\| = N'\right) \geq 1 - \epsilon. \tag{140}$$

Therefore, the system is strictly-A-diagnosable.

(Necessity) Necessity will be shown by contradiction using the necessary and sufficient condition for strict-A-diagnosability. Suppose there exists a string s such that s transitions the system from the initial state to a recurrent component c_R in a state that is either uncertain or unsafe. Let the probability of s be denoted by p_s.

Since c_R is not in a certain, safe state, we must choose an observation policy that observes at least one event and thus by assumption has a positive cost. Furthermore, since c_R is recurrent, the probability that the system returns to the logical element containing c_R infinitely often is 1.

Since each time the system reaches this logical element the language is not diagnosed, we must pay a positive cost infinitely often. Since we must pay an infinite cost with probability $p_s > 0$, the expected cost of diagnosing the language is infinite.

Therefore, if the language can be diagnosed at finite expected cost, each recurrent component in its stochastic diagnoser lies in a certain, safe state, and thus a language is diagnosable at finite expected cost only if it is strictly-A-diagnosable. □

4.6 Almost sure diagnosability conditions

We also formulate the active acquisition of information problem for situations where diagnosis is made when the probability of failure is greater than a pre-defined $\alpha < 1$. An observation policy g and the space of information states are defined in the same way as in Section 4.1. For completeness of presentation we restate Definitions 11–16 within the context of the diagnosis problem studied here.

Definition 29 An information state π is *almost-F-certain* if $\Pr(s : f \in s \mid s \in \pi) > \alpha$.

Definition 30 An information state π is *almost-N-safe* if $\Pr(s : f \notin s \wedge f \notin L/s \mid s \in \pi) > \alpha$.

Definition 31 An information state π is *safe* if π is almost-F-certain or almost-N-safe.

Definition 32 An information state π is *uncertain* if $\alpha > \Pr(s : f \in s \mid s \in \pi) > 1 - \alpha$.

Definition 33 An information state π is *non-diagnosable* if $\exists N \in \mathbb{N}$ such that for all $n \geq N, \exists t \in L/\pi$ such that $\|t\| = n$ and the information state obtained by implementing any policy g along t is uncertain in the sense of Definition 32.

Definition 34 A language $\mathcal{L}(G)$ is *almost surely diagnosed* by an observation policy g if, for all $s \in \mathcal{L}(G)$, the information state reached by implementing g is never non-diagnosable.

Definition 35 Let H denote the set of all policies that diagnose $\mathcal{L}(G)$. The language $\mathcal{L}(G)$ is *almost surely diagnosable* if H is non-empty, i.e., if there exists a policy that surely diagnoses $\mathcal{L}(G)$.

The cost associated with any observation policy g is

$$J(g) = \{E(c^g(s) + K^g(s))\}, \tag{141}$$

where $c^g(s)$ denotes the cost of implementing policy g along the trajectory s and $K^g(s)$ is defined in Eq. 87. As is Problem SCSD, the performance criterion is thus the expected total cost of policy g.

The active acquisition of information problem, or stochastic cyclic almost sure diagnosis problem, is defined as follows.

Problem SCASD Find a policy $g^* \in H$ such that

$$J(g^*) = \inf(J(g)|g \in H) < \infty. \tag{142}$$

4.6.1 Strict-AA-diagnosability; solution existence for Problem SCASD

For an optimal finite-cost solution to Problem SCASD to exist, we wish to ensure that, as in the case of Problem SCSD, a diagnosis is almost surely made in a finite amount of time. This motivates the notion of strict-AA-diagnosability.

Definition 36 A language L is strictly-AA-diagnosable if

$$\begin{aligned}&\big(\forall\epsilon > 0 \wedge \forall\alpha < 1\big)(\exists N \in \mathbb{N})\\ &\quad(\forall n > N) \Pr\big(s : D^F_\alpha(s) = 0 \wedge \hat{D}^N_\alpha(s) = 0 \mid \|s\| = n\big) < \epsilon,\end{aligned} \tag{143}$$

where the function $\hat{D}^N_\alpha$ is defined analogously to D^F_α as:

$$D^F_\alpha(st) = \begin{cases} 1 & \text{if } \Pr\big(\omega : \Sigma_{f_i} \in \omega \mid \omega \in P_L^{-1}[P(st)]\big) > \alpha \\ 0 & \text{otherwise.}\end{cases} \tag{144}$$

While strict-A-diagnosability is a more stringent condition that A-diagnosability, this is not the case with their almost sure equivalents. In fact, strict-AA-diagnosability and AA-diagnosability are identical conditions. While it is fairly clear that strict-AA-diagnosability should imply AA-diagnosability, the opposite implication is not intuitively obvious; the idea behind this implication is as follows. If a system is AA-diagnosable and no failure occurs, the probability that the system does not reach a safe normal state becomes arbitrarily small in the long run, as the set of unsafe normal states is transient. Thus if no failure occurs, we will almost surely eventually diagnose that the system is in normal operation, and since the system is AA-diagnosable, we will almost surely eventually diagnose any failure events. We show this equivalence formally in the following theorem.

Theorem 12 *A language is strictly-AA-diagnosable if and only if it is AA-diagnosable.*

Proof (Necessity) Suppose L is strictly-AA-diagnosable, but not AA-diagnosable. We will show this assumption leads to a contradiction.

Since L is not AA-diagnosable, there exists $s \in \Psi(\Sigma_{f_i})$ such that, for some $\epsilon_1 > 0$ and $\alpha < 1$ such that $\epsilon_1 > 1 - \alpha$, there exists an arbitrarily large n_1 such that

$$\Pr\left(t : D_\alpha^F(st) = 0 \mid t \in L/s \wedge \|t\| = n_1 - \|s\|\right) \geq \frac{\epsilon_1}{p_s}, \tag{145}$$

where p_s is the probability of the string s. Therefore

$$\Pr\left(st : D_\alpha^F(st) = 0 \mid \|st\| = n_1\right) \geq \epsilon_1 \tag{146}$$

Choose ϵ_2 such that $\epsilon_1 > \epsilon_2 > 1 - \alpha$. Since L is strictly-AA-diagnosable, there exists $N \in \mathbb{N}$ such that for $n_2 > N$,

$$\Pr\left(s : D_\alpha^F(s) = 0 \wedge \hat{D}_\alpha^N(s) = 0 \mid \|s\| = n_2\right) < \epsilon_2 - (1 - \alpha) \tag{147}$$

$$\Pr\left(s : D_\alpha^F(s) = 1 \vee \hat{D}_\alpha^N(s) = 1 \mid \|s\| = n_2\right) \geq 1 - \epsilon_2 + (1 - \alpha) \tag{148}$$

$$\Pr\left(s : D_\alpha^F(s) = 1 \mid \|s\| = n_2\right) + \Pr\left(s : \hat{D}_\alpha^N(s) = 1 \mid \|s\| = n_2\right) \geq 1 - \epsilon_2 + (1 - \alpha) \tag{149}$$

$$\begin{aligned}&\Pr\left(s : D_\alpha^F(s) = 1 \wedge f \in s\right) + \Pr\left(s : \hat{D}_\alpha^N(s) = 1 \wedge f \notin s = 1\right)\\ &\quad + \Pr\left(s : D_\alpha^F(s) = 1 \wedge f \notin s\right) + \Pr\left(s : \hat{D}_\alpha^N(s) = 1 \wedge f \in s\right) \geq 1 - \epsilon_2 + (1 - \alpha).\end{aligned} \tag{150}$$

Consider the probability that a string is misdiagnosed as faulty, i.e. $\Pr(s : D_\alpha^F(s) = 1 \wedge f \notin s)$ This probability is equal to

$$\begin{aligned}&\Pr\left(s : D_\alpha^F(s) = 1 \wedge f \notin s \mid \|s\| = n_2\right)\\ &\quad = \Pr\left(s : f \notin s \mid D_\alpha^F(s) = 1 \wedge \|s\| = n_2\right) \Pr\left(s : D_\alpha^F(s) = 1 \mid \|s\| = n_2\right)\end{aligned} \tag{151}$$

$$\leq (1 - \alpha) \Pr\left(s : D_\alpha^F(s) = 1 \mid \|s\| = n_2\right), \tag{152}$$

as the fact that a faulty diagnosis was made indicates that the probability that a fault occurred is greater than α. Similarly, it can be shown that

$$\Pr\left(s : \hat{D}_\alpha^N(s) = 1 \wedge f \in s \mid \|s\| = n_2\right) \leq (1 - \alpha) \Pr\left(s : \hat{D}_\alpha^N(s) = 1 \mid \|s\| = n_2\right). \tag{153}$$

By combining the inequalities (150), (152), and (153), we obtain

$$\begin{aligned}&\Pr\left(s : D_\alpha^F(s) = 1 \wedge f \in s \mid \|s\| = n_2\right)\\ &\quad + \Pr\left(s : \hat{D}_\alpha^N(s) = 1 \wedge f \notin s = 1 \mid \|s\| = n_2\right) \geq 1 - \epsilon_2.\end{aligned} \tag{154}$$

Since both n_1 can be chosen arbitrarily large and n_2 can be any sufficiently large integer, choose $n = n_1 = n_2$. Therefore inequalities (146) and (154) can be added, so

$$\Pr\left(s : D_\alpha^F(s) = 1 \wedge f \in s \mid \|s\| = n\right) + \Pr\left(s : \hat{D}_\alpha^N(s) = 1 \wedge f \notin s + 1 \mid \|s\| = n\right) \\ + \Pr\left(st : D_\alpha^F(st) = 0 \mid \|st\| = n\right) \geq 1 + \epsilon_1 - \epsilon_2. \tag{155}$$

Since these three probabilities are of disjoint events, we have that

$$\Pr\left(s : f \in s \vee \hat{D}_\alpha^N(s) = 1 \wedge f \notin s \mid \|s\| = n\right) \geq 1 + \epsilon_1 - \epsilon_2. \tag{156}$$

However, since $\epsilon_1 > \epsilon_2$, the probability of this event is greater than 1. Therefore we have reached a contradiction, and L cannot be both strictly-AA-diagnosable and not AA-diagnosable. Therefore if L is strictly-AA-diagnosable, L is AA-diagnosable.

(Sufficiency) We shall prove the contrapositive statement. Suppose L is not strictly-AA-diagnosable. Then there exist $\epsilon_1 > 0, \alpha < 1$ such that for all $N \in \mathbb{N}$, there exists $n_1 > N$ such that

$$\Pr\left(s : D_\alpha^F(s) = 0 \wedge \hat{D}_\alpha^N(s) = 0 \mid \|s\| = n_1\right) \geq \frac{\epsilon_1}{1-\alpha}. \tag{157}$$

Let $S = \{s : D_\alpha^F(s) = 0 \wedge \hat{D}_\alpha^N(s) = 0 \wedge \|s\| = n_1\}$. Partition the set S into $S = S_1 \dot\cup S_2 \dot\cup \ldots \dot\cup S_k$, where every string in each S_i has a unique projection s_i. Therefore, since $\hat{D}_\alpha^N(s) = 0$ for all s in each S_i,

$$\Pr\left(s : s \in S_i \wedge f \in s\right) \geq (1-\alpha)\Pr(s : s \in S_i), \tag{158}$$

so

$$\Pr\left(s : D_\alpha^F(s) = 0 \wedge \hat{D}_\alpha^N(s) = 0 \wedge f \in s \mid \|s\| = n_1\right)$$

$$\geq (1-\alpha)\Pr\left(s : D_\alpha^F(s) = 0 \wedge \hat{D}_\alpha^N(s) = 0 \mid \|s\| = n_1\right) \tag{159}$$

$$\geq \epsilon_1 \tag{160}$$

$$\Pr\left(s : D_\alpha^F(s) = 0 \wedge f \in s \mid \|s\| = n_1\right) \geq \epsilon_1. \tag{161}$$

Choose $\epsilon_2 < \epsilon_1$. Again, by Lemma 1 of Thorsley and Teneketzis (2005), there exists $N_2 \in \mathbb{N}$ such that $n_2 > N_2$ implies

$$\Pr\left(f \in s \wedge f \notin \text{the first } N_2 \text{ events of } s \mid \|s\| = n_2\right) < \epsilon_2. \tag{162}$$

Let p_F denote the probability that a failure occurs in the first N_2 events of a string. Choose ϵ such that $\epsilon_1 = \epsilon p_F + \epsilon_2$. Rewrite inequality (161) as

$$\Pr\left(st : D_\alpha^F(st) = 0 \wedge f \in st \wedge f \notin s \mid \|s\| = N_s \wedge \|t\| = n_2 - N_s\right) \\ + \Pr\left(st : D_\alpha^F(st) = 0 \wedge f \in s \mid \|s\| = N_s \wedge \|t\| = n_2 - N_s\right) \geq \epsilon p_F + \epsilon_2. \tag{163}$$

From inequality (162), the first term of the above is less than ϵ_2, and thus

$$\Pr\left(st : D_\alpha^F(st) = 0 \wedge f \in s \mid \|s\| = N_s \wedge \|t\| = n_2 - N_s\right) \geq \epsilon p_F. \tag{164}$$

Divide the probability of st into the probability of s and t:

$$\Pr\left(s : f \in s \mid \|s\| = N_s\right) \Pr\left(t : D_\alpha^F(st) = 0 \mid t \in L/s \wedge \|t\| = n_2 - N_s\right) \geq \epsilon p_F \quad (165)$$

$$\Pr\left(t : D_\alpha^F(st) = 0 \mid t \in L/s \wedge \|t\| = n_2 - N_s\right) \geq \epsilon. \quad (166)$$

□

Having demonstrated that strict-AA-diagnosability and AA-diagnosability are equivalent, we now state the conditions under which Problem SCASD has a solution.

Theorem 13 *A stochastic automation is diagnosable under conditions $\hat{D}_\alpha^N$ and D_α^F for all $\alpha < 1$ at finite expected cost if and only if the automaton is AA-diagnosable when all event in Σ_o are observed.*

Proof (Necessity) If the system is AA-diagnosable, then after $n(\alpha, \epsilon)$ events, a diagnosis to α will be made with probability $1 - \epsilon$. Since for any $\epsilon > 0$, this bound n is finite, the number of events we need to observe to make a diagnosis to α is finite and bounded with probability 1. Therefore the expected cost of the system is finite.

(Sufficiency) The prove for sufficiency is the same is in the case of exact diagnosis.

□

Using the results from Thorsley and Teneketzis (2005), we now state a sufficient condition for a stochastic automaton to be diagnosable with finite expected cost.

Corollary 1 *A stochastic automaton is diagnosable under condition D_α for all $\alpha < 1$ at finite expected cost if the set of recurrent components in each logical element of its stochastic diagnoser is certain.*

4.7 Comments on solution methods

In general, the state estimate of a stochastic automaton is an element of an infinite space, as there may be an infinite number of probability mass functions associated with a given logical diagnoser state (Lunze and Schröder 2001). Thus we cannot perform a reduction from an infinite set of string-based information states to the finite set of state-based information states as in the logical case.

For Problem SCSD, we assign costs to certain information states as follows:

$$\hat{V}(\pi) = \begin{cases} 0 & \text{if } \pi \text{ is safe} \\ \infty & \text{if } \pi \text{ is not strictly-}A\text{-diagnosable,} \end{cases} \quad (167)$$

where an information state $\pi = s_1 + s_2 + \cdots + s_n$ is defined to be strictly-A-diagnosable if the language $L_\pi := \hat{P}(s_1)t_1 + \hat{P}(s_2)t_2 + \cdots + \hat{P}(s_n)t_n$ is strictly-A-diagnosable, where $\hat{P}$ is the projection of Σ onto Σ_{uo}.

An optimal policy can be computed by solving the dynamic programming equations:

$$V(\pi) = \min_{u \in 2^{\Sigma_{co}}} c_u + \sum_{\sigma \in \Sigma_{u,obs}} V(\hat{\delta}_u(\pi, \sigma)) P_u(\sigma \mid \pi), \quad (168)$$

where $P_u(\sigma \mid \pi)$ is the probability that the next observed event is σ, given the current information state of the system is state π and the action u is implemented, and $\hat{\delta}_u$ is given by Eqs. 79–80.

For information states that are either safe or not AA-diagnosable, Eq. 168 gives

$$V(\pi) = \begin{cases} 0 & \text{if } \pi \text{ is safe} \\ \infty & \text{if } \pi \text{ is not } AA\text{-diagnosable,} \end{cases} \tag{169}$$

where an information state π is defined to be AA-diagnosable if the language L_π is AA-diagnosable.

While finding an optimal policy requires considering optimal action for an infinite set of information states, we can find a suboptimal policy by modifying the limited lookahead algorithm of Section VI to minimize the expected observation cost before the lookahead horizon instead of the maximal observation cost.

5 Discussion

This paper provides a framework for formulating and solving various active acquisition problems in discrete-event systems relating to fault diagnosis and supervisory control. For acyclic systems, the problem of finding an optimal observation policy can be solved in two steps: first, an appropriate filtration of σ-fields of information states comprised of sets of strings must be identified, and then a policy can be found using dynamic programming. This two-step process works for both logical and stochastic models. For cyclic systems, in order to ensure that the space of information states is finite, the set of string-based information states is reduced to the set of diagnoser states. The dynamic programming technique results in a set of algebraic equations in the cyclic case.

Further research is necessary in the area of computational efficiency. In acyclic systems, the size of the space of information states is in the worst case doubly exponential with respect to the size of the finite horizon. In cyclic systems, the set of diagnoser states is exponential with respect to the state space of the automaton under consideration. In this paper, we presented limited lookahead algorithms for both classes of systems that provide a first attempt at finding efficient algorithms for optimal and suboptimal solutions. Heuristic methods may be more successful at finding efficient algorithms for particular classes of systems.

Acknowledgements This research was supported in part by NSF Grants CCR-0325571, ECS-0080406 and a grant from the Xerox University Affairs Committee. This research was completed while D. Thorsley was with the Department of Electrical Engineering and Computer Science, University of Michigan. The authors are grateful to the anonymous reviewers whose comments significantly improved the presentation of this paper.

References

Andersland MS, Teneketzis D (1992) Information structures, causality, and non-sequential stochastic control, I: design-independent properties. SIAM J Contr Optim 30(6):1447–1475

Andersland MS, Teneketzis D (1994) Information structures, causality, and non-sequential stochastic control, II: design-dependent properties. SIAM J Contr Optim 32(6):1726–1751

Andersland MS, Teneketzis D (1996) Measurement scheduling for recursive team estimation. J Optim Theory Appl 89(3):615–636

Appadwedula S, Veeravalli VV, Jones DL (2002) Robust and locally-optimum decentralized detection with censoring sensors. In: Proceedings of the 5th international conference on information fusion. Annapolis, MD, USA

Athans M (1972) On the determination of optimal costly measurement strategies for linear stochastic systems. Automatica 8:397–412

Cassandras CG, Lafortune S (1999) Introduction to discrete event systems. Kluwer Academic Publishers, Boston, MA

Debouk R, Lafortune S, Teneketzis D (2002) On an optimization problem in sensor selection. J Discret Event Dyn Syst: Theory and Appl 12:417–445

Ding X, Puterman ML, Bisi A (2002) The censored newsvendor and the optimal acquisition of information. Oper Res 50:517–527

Holloway L, Chand S (1994) Time templates for discrete event fault monitoring in manufacturing systems. In: Proceedings of the 1994 American control conference, pp 701–706

Jiang S, Kumar R, Garcia HE (2003) Optimal sensor selection for discrete-event systems with partial observation. IEEE Trans Syst Man Cybern Part B 30(5):653–660

Jiang S, Huang Z, Chandra V, Kumar R (2001) A polynomial algorithm for testing diagnosability of discrete-event systems. IEEE Trans Automat Contr 46(8):1318–1320

Khanna M (1973) Sampling and transmission policies for controlled Markov processes with costly communication. Ph.D. thesis, Department of Electrical Engineering, University of Toronto

Kumar PR, Varaiya P (1986) Stochastic systems: estimation, identification, and adaptive control. Prentice Hall, Englewood Cliffs, NJ

Kushner HJ (1964) On the optimum timing of observations for linear control systems with unknown initial state. IEEE Trans Automat Contr 9(2):144–150

Kushner HJ (1971) Introduction to stochastic control. Holt, Rinehart, Winston

Lafortune S, Teneketzis D, Sampath M, Sengupta R, Sinnamohideen K (2001) Failure diagnosis of dynamic systems: an approach based on discrete event systems. In: Proceedings of the 2001 American control conference, pp 2058–2071

Lunze J, Schröder J (2001) State observation and diagnosis of discrete-event systems described by stochastic automata. Discret Event Dyn Syst: Theory Appl 11(4):319–369

Meier III L, Peschon J, Dressler RM (1967) Optimal control of measurement subsystems. IEEE Trans Automat Contr 12(5):528–536

Pencolé Y (2000) Decentralized diagnoser approach: application to telecommunication networks. In: Proceedings of the 11th international workshop on principles of diagnosis (DX'00), pp 185–192

Pollard D (2002) A user's guide to measure theoretic probability. Cambridge Univ. Press

Rago C, Willett P, Bar-Shalom Y (1996) Censoring sensors: a low-communication-rate scheme for distributed detection. IEEE Trans Aerosp Electron Sys 32(2):554–568

Rozé L, Cordier M-O (1998) Diagnosing discrete-event systems: an experiment in telecommunication networks. In: Proceedings of the 1998 international workshop on discrete event systems (WODES '98). Published by IEE, London, England. pp 130–137

Sampath M, Sengupta R, Lafortune S, Sinnamohideen K, Teneketzis D (1995) Diagnosability of discrete-event systems. IEEE Trans Automat Contr 40(9):1555–1575

Teneketzis D (1996) On information structures and nonsequential stochastic control. CWI Quarterly 9(3):241–260 (special issue on Systems and Control)

Teneketzis D, Andersland MS (2000) On partial order characterizations of information structures. Math Contr Signals Syst 13:277–292

Thorsley D, Teneketzis D (2005) Diagnosability of stochastic discrete-event systems. IEEE Trans Automat Contr 50(4):476–492

Witsenhausen HS (1971) On information structures, feedback and causality. SIAM J Contr 9(2): 149–160

Witsenhausen HS (1975) The intrinsic model for discrete stochastic control: some open problems. In: Lecture notes in economics and mathematical systems, vol. 107. Springer, Berlin, pp 322–335

Yoo T-S, Garcia HE (2003) Computation of fault detection delay in discrete-event systems. In: Proceedings of the 14th international workshop on the principles of diagnosis (DX-03)

Yoo T-S, Lafortune S (2002a) NP-completeness of sensor selection problems arising in partially observed discrete-event systems. IEEE Trans Automat Contr 47(9):1495–1499

Yoo T-S, Lafortune S (2002b) Polynomial-time verification of diagnosability of partially observed discrete-event systems. IEEE Trans Automat Contr 47(9):1491–1495

David Thorsley received a B.E.Sc. in electrical engineering from the University of Western Ontario, London, Canada, in 2000, and M.S. and Ph.D. degrees in Electrical Engineering: Systems from the University of Michigan, Ann Arbor, in 2002 and 2006, respectively. In summer 2006, he was awarded a postdoctoral fellowship from the Idaho National Laboratory. He is currently a Research Associate in the Department of Electrical Engineering at the University of Washington, Seattle. His research interests are in stochastic control, probability theory and Markov processes, discrete-event systems, and systems biology.

Demosthenis Teneketzis received the diploma in electrical engineering from the University of Patras, Patras, Greece, and the M.S., E.E., and Ph.D. degrees, all in electrical engineering, from the Massachusetts Institute of Technology, Cambridge, in 1974, 1976, 1977, and 1979, respectively. He is currently Professor of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor. In winter and spring 1992, he was a Visiting Professor at the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland. Prior to joining the University of Michigan, he worked for Systems Control, Inc., Palo Alto, CA, and Alphatech, Inc., Burlington, MA. His research interests are in stochastic control, decentralized systems, queueing and communication networks, stochastic scheduling and resource allocation problems, mathematical economics, and discrete-event systems.

Discrete Event Dyn Syst (2007) 17:585–586
DOI 10.1007/s10626-007-0030-3

Active Acquisition of Information for Diagnosis and Supervisory Control of Discrete Event Systems

David Thorsley · Demosthenis Teneketzis

Published online: 16 October 2007

**Erratum to: Discrete Event Dyn Syst
DOI 10.1007/s10626-007-0027-y**

In (Thorsley and Teneketzis 2007), Definition 1 is incomplete. The revised definition below states the additional necessary condition missing from Thorsley and Teneketzis (2007).

Definition 1 An observation policy $g := (g_0, \ldots, g_{T-1})$ is a sequence of functions $g_t : L_T \to 2^{\Sigma_{co}}$ such that for all $t, t = 0, \ldots, T-1$, g_t is measurable with respect to the σ-field $\mathcal{G}_t^g$, defined below (in Definition 3).

Note that for all $s' \in L_t$ and $s, \hat{s} \in \chi_t(s')$, $g_t(s) = g_t(\hat{s})$. The statement "The functions χ_t are used in the following definition," found above Definition 1 in the text, should be ignored when reading the paper.

Definition 3 is unchanged, but we restate it here for convenience.

The online version of the original article can be found at http://dx.doi.org/10.1007/s10626-007-0027-y

D. Thorsley (✉)
Department of Electrical Engineering, University of Washington, Seattle, WA 98195, USA
e-mail: thorsley@ee.washington.edu

D. Teneketzis
Department of EECS, University of Michigan, Ann Arbor, MI 48109, USA
e-mail: teneketzis@eecs.umich.edu

Definition 3 The filtration $\{\mathcal{G}_t^g, t = 0 \ldots T\}$ corresponding to g is

$$\sigma\left(\pi_t : \pi_t \in R_t^g\right), t = 0 \ldots T. \tag{11}$$

Note that $\mathcal{G}_t^g, t = 0 \ldots T-1$, depends on $g_0, g_1, \ldots, g_{t-1}$.

References

Thorsley D, Teneketzis D (2007) Active acquisition of information for diagnosis and supervisory control of discrete event systems. Discret Event Dyn Syst: Theory and Applications 17(4): 531–583