2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Dynamic Sensor Activation for Event Diagnosis

Weilin Wang, Stéphane Lafortune, Anouck R. Girard, and Feng Lin

***Abstract*— We consider the problem of dynamic sensor activation for event diagnosis in partially-observed discrete-event systems. The observing agent is able to activate sensors dynamically during the evolution of the system. The sensor activation policy is the function that describes which sensors are to be activated after an observed string of events. The sensor activation policy must achieve the requirements of the property of diagnosability previously defined for discrete event systems. A policy is said to be minimal if there is no other policy, with strictly less sensor activation, that achieves diagnosability. For the purpose of computing minimal policies, we define language partition methods that lead to efficient computational algorithms. Specifically, we define "window-based" language partitions that lead to scalable algorithms for computing minimal policies. By increasing the size of the window in this class of partitions, one is able to refine the solution space over which minimal solutions are computed.**

***Index Terms*— discrete event systems, supervisory control, sensor activation, event diagnosis**

I. Introduction

We are interested in the problem of event diagnosis in partially-observed discrete-event systems. The objective is to perform model-based inferencing to detect the occurrence of significant unobservable events such as faults. On-line diagnosis is driven by the observed sequences of events. In many applications, these observations are limited or costly. Such limitations include, for example, availability of sensors and their life span, battery power, as well as computation and communication resources. Therefore, there is significant motivation to use sensors economically in system diagnosis.

Relevant work in the discrete event formalism started with the sensor selection problem. Under the constraint that diagnosability or observability be preserved, the objective of sensor selection is to minimize the *set of events* that need to be observed; see, e.g., [1]–[3]. However, by assuming a given sensor is always activated or never activated for all occurrences of an event, the sensor selection problem excludes the possibility for the observing agent to decide dynamically when to activate or deactivate the various sensors.

The motivation for turning sensors on and off dynamically is that sensors are often operated in an adversarial environment where available resources are limited or costly. For example, in a unmanned aerial vehicle system, making a measurement may cost hours of flight; in a radar system, emitting radar signals can be dangerous since they can be used to detect the position of the radar station. Moreover, the life span of a sensor is often dependent on its measurement frequency. In several applications, security concerns may motivate the minimization of communications with sensing devices. These considerations have motivated recent research on the problem of dynamic sensor activation for discrete event systems [4], [5]. The objective in these works is to minimize some cost function related to the measurement frequency of event occurrences so that the property of diagnosability is preserved. There has also been related work on the problem of minimization of the frequency of communication of event occurrences in distributed discrete event systems with multiple agents [6], [7].

The approaches in [4], [5] have high computational cost. We adopt a different approach to dynamic sensor activation where the problem is formulated in a manner where a trade-off between computational cost and refinement of the solution space is captured. In recent work reported in [8], polynomial algorithms (in the state space representation of the system) for calculating minimal sensor activation policies are presented for achieving the properties of observability and coobservability in controlled discrete event systems. In this paper, our focus is the property of diagnosability. A significant difference with the work in [8] is the introduction in this paper of the notions of *language partitions* and *window partitions* for characterizing the search space over which optimal sensor activation policies are to be searched. This leads to the formulation of scalable algorithms for minimizing sensor activation while ensuring that diagnosability of the system is preserved. In [8], the solution space is fixed by the transition structure of the automaton modeling the system. Here, the use of window partitions permits the relationship between the size of the solution space and the size of the problem to be treated as an input to the algorithms. So the algorithms will be adjustable upon different hardware and software environments, and upon the size of the problem. Running our algorithms over a finer solution space can result in a better solution, but at the price of more computational effort. For fixed computational resources, this technique ensures a desirable balance between high solution accuracy and fast processing time.

We borrow some features from the communication problems and sensor activation problem described in [6]–[8] for the problem setting. The notion of feasibility is used to capture the consistency between the agent's observation of the system and its decisions on sensor activation. The optimality criterion is logical: a sensor activation policy is optimal (minimal) if any less sensor activations during the dynamic evolution of the system renders a correct solution

Weilin Wang and Anouck R. Girard are with the Department of Aerospace Engineering, Stéphane Lafortune is with the Department of EECS, University of Michigan, Ann Arbor, MI 48109, {weilinw, anouck, stephane}@umich.edu. Feng Lin is with the Department of ECE, Wayne State University, Detroit, MI 48202, flin@ece.eng.wayne.edu

978-1-4244-4524-0/09/$25.00 ©2009 AACC

incorrect, i.e., diagnosability is lost.

This paper is organized as follows. Section II presents the precise description of our sensor activation model and its relation to the property of diagnosability. In Section III, we specialize the problem to the computation of sensor activation policies over a finite search space delimited by a partition of the language model of the system. We first present a general class of language partitions and then specialize this class to so-called window partitions based on automata representations of the language model. Section IV presents our algorithms for computing minimal sensor activation policies over a given window partition in order to achieve diagnosability; illustrative examples of these algorithms are given in Section V. Section VI concludes the paper. Due to space limitations, all proofs have been omitted; they are available from the authors.

II. Sensor Activation for Diagnosability

A. Sensor Activation Model

We first present a general language-based model for sensor activation during the evolution of the system. We assume that the system behavior is described by the prefix-closed language L over event set E. Let $E = E_{uo} \cup E_o$, where E_{uo} is the subset of E whose occurrences cannot be observed and E_o is the subset of E whose occurrences can be observed. However, an occurrence of event $e \in E_o$ is observed only if the sensor for event e is *activated* at the time of that occurrence. Formally, the set of observable events whose sensors are activated at a given time in the evolution of the system is described by the sensor activation mapping

$$\omega : L \rightarrow 2^{E_o} \tag{1}$$

Given two sensor activation mappings ω' and ω'', $\omega' \subseteq \omega''$ means that, for all $s \in L$, $\omega'(s) \subseteq \omega''(s)$, while $\omega' \subset \omega''$ means that $\omega' \subseteq \omega''$ and there exists $s \in L$, such that $\omega'(s) \subset \omega''(s)$. $\omega = \omega' \cup \omega''$ means that, for all $s \in L$, $\omega(s) = \omega'(s) \cup \omega''(s)$.

Given ω, we use induction to define the information mapping $\theta^{\omega} : L \rightarrow E_o^*$ as follows. For the empty string ε,

$$\theta^{\omega}(\varepsilon) = \varepsilon, \tag{2}$$

and for all $se \in L, e \in E$,

$$\theta^{\omega}(se) = \begin{cases} \theta^{\omega}(s)e & \text{if } e \in \omega(s) \\ \theta^{\omega}(s) & \text{otherwise} \end{cases} \tag{3}$$

In words, after the occurrence of s, the next event e is *known* to the agent iff the agent activates the sensor for e after the occurrence of s.

Given language L and sensor activation policy ω, let θ^{ω} be the corresponding information mapping. The set of confusable string pairs in L, denoted by $\mathcal{T}_{conf}(\omega)$, is defined as

$$\mathcal{T}_{conf}(\omega) = \{(s,t) \in L \times L : \ \theta^{\omega}(s) = \theta^{\omega}(t)\} \tag{4}$$

We note here that for all $s \in L$, we have $(s,s) \in \mathcal{T}_{conf}(\omega)$.

It is important to clarify that not all arbitrary sensor activation policies ω will be "feasible" based on the information available to the agent. To guarantee feasibility, it is required that any two sequences of events that are indistinguishable to the agent must be followed by the same activation policy for the same event. Namely, activation policy ω must be "compatible" with the information mapping θ^{ω} that is built from it. Formally, ω is said to be *feasible* if

$$\begin{aligned} &(\forall\, e \in E)(\forall\, se, s'e \in L)\ \theta^{\omega}(s) = \theta^{\omega}(s') \\ &\Rightarrow\ [e \in \omega(s) \Leftrightarrow e \in \omega(s')] \end{aligned} \tag{5}$$

In principle, to check feasibility, we first calculate θ^{ω} from ω, and then check if (5) holds.

For the prefix-closed language $L' \subseteq L$, we define $\omega|_{L'} = \omega \cap (L' \times 2^{E_o})$, where $\omega \cap (L' \times 2^{E_o})$ means that we are restricting ω to the smaller domain of the prefix-closed sublanguage L'. And, correspondingly, we define information mapping $\theta|_{L'} = \theta \cap (L' \times E_o^*)$, where $\theta \cap (L' \times E_o^*)$ means that we are restricting θ to the smaller domain of the prefix-closed sublanguage L'.

$\omega|_{L'}$ is said to be *feasible* if

$$\begin{aligned} &(\forall\, e \in E)(\forall\, se, s'e \in L')\ \theta^{\omega}|_{L'}(s) = \theta^{\omega}|_{L'}(s') \\ &\Rightarrow\ [e \in \omega|_{L'}(s) \Leftrightarrow e \in \omega|_{L'}(s')] \end{aligned} \tag{6}$$

Clearly, if ω is feasible, then $\omega|_{L'}$ is also feasible.

Feasibility is used to capture the interdependence of the agent's observation of the system and the sensor activation policies. That is, in general, the determination of when to activate sensors depends on the "observation" of the system, and at the same time it also affects the "observation" of the system. An example that illustrates such interdependency is given as follows.

Example 1: The system is modeled by the regular expression $L = PC[(e+\varepsilon)(ae)^*]$ where the notation PC denotes the prefix-closure operation. (We use PC instead of the more common overbar notation to avoid confusion with later notations.) Let $E = E_o = \{a, e\}$. The agent can freely choose to deactivate the sensor for any one but exactly one event occurrence for either event a or event e, and it is easy to verify that these policies are feasible. Now, suppose the agent initially deactivates the sensor for event e (after empty string ε). Consequently, the agent cannot distinguish strings ε and e. By feasibility, such ambiguity precludes the agent from using different policies for the occurrence of event a after strings $(ae)^n$ and $e(ae)^n$, as well as event e after strings $(ae)^n a$ and $e(ae)^n a$, where $n \in \mathbb{N}$. Moreover, if the agent further deactivates the sensor for the occurrence of event a after string ε, then, by feasibility, it has to deactivate sensors for all occurrences of all events. ■

B. Diagnosability with Information Mappings

In the context of sensor activation, the agent no longer observes an occurrence of an observable event unless the corresponding sensor is activated when such an event happens. However, whenever a sensor activation policy is given, the agent's observation of the system is captured by a corresponding information mapping, as stated in Section II-A. We extend the standard definition of *diagnosability* of discrete event systems to account for information mappings.

Let $E_f \subseteq E_{uo}$ denote the set of "fault" events that must be diagnosed. The objective is to identify the occurrence, if any, the fault events, while tracking the observable strings generated by the system. For such purpose, the set of fault events is partitioned into disjoint sets corresponding to different fault types:

$$E_f = E_{f1} \dot\cup \ldots \dot\cup E_{fl} \tag{7}$$

We denote this partition by Π_f. Hereafter, the meaning of "a fault of type F_i has occurred" is that some event in the set E_{fi} has occurred.

Hereafter, $s \in \Psi(E_{fi})$ denotes that the last event of a string $s \in L$ is a fault event of type f_i. That is,

$$\Psi(E_{fi}) = \{s\sigma_f \in L : \sigma_f \in E_{fi}\} \tag{8}$$

L/s denotes the postlanguage of L after s, i.e.

$$L/s = \{t \in E^* : st \in L\} \tag{9}$$

With a slight abuse of notation, we write $E_{fi} \in s$ to denote that $PC(s) \cap \Psi(E_{fi}) \neq \emptyset$. A language L is live if

$$((\forall\, s \in L, n \in \mathbb{N})\, \exists\, t \in E^*) \|t\| = n \wedge st \in L \tag{10}$$

where $\|t\|$ denotes the length of t and $\mathbb{N}$ is the set of nonnegative integers. We assume that L is live when diagnosability is considered.

In the context of a given information mapping $\theta : E^* \to E_o^*$, the definition of diagnosability in [9] is restated as follows, where θ^{-1} is the inverse mapping of θ defined in the usual manner.

A prefix-closed and live language L is said to be diagnosable with respect to θ and Π_f on E_f if the following holds:

$$\begin{aligned}&(\forall\, i \in \Pi_f)(\exists\, k_i \in \mathbb{N})(\forall\, s \in \Psi(E_{fi}))\\ &(\forall\, t \in L/s)[\|t\| \geq k_i \Rightarrow D]\end{aligned} \tag{11}$$

where the diagnosability condition D is

$$(\forall \mu \in L) \mu \in \theta^{-1}\theta(st) \Rightarrow E_{fi} \in \mu \tag{12}$$

A prefix-closed and live language L is said to be diagnosable with respect to ω if it is diagnosable with respect to θ^{ω}.

The definition of diagnosability says that, for any string in the system which contains any type of faulty events, the diagnostic engine can distinguish that string from strings without that type of faulty events within finite delay.

C. Monotonicity of Diagnosability in Sensor Activation

In this section, we formally analyze our problem of minimization of sensor activation.

The first theorem establishes the monotonicity of feasible sensor activation policies.

Theorem 1: Given a prefix-closed language L with two sensor activation policies ω' and ω'', if ω' and ω'' are both feasible, i.e., they both satisfy (5), then

$$\omega' \supseteq \omega'' \Rightarrow \mathscr{T}_{conf}(\omega') \subseteq \mathscr{T}_{conf}(\omega'') \tag{13}$$

The second theorem discusses the union of two feasible policies, which is a direct result of the monotonicity property.

Theorem 2: Consider a prefix-closed language L and two feasible sensor activation policies ω' and ω'' for it. Then, $\omega = \omega' \cup \omega''$ is also feasible.

The next theorem states that monotonicity holds for diagnosability under feasible sensor activation policies.

Theorem 3: Suppose E_o and $E_{fi}, i = 1, \ldots, l$ are given. Let sensor activation policies ω' and ω'' be feasible and $\omega' \subseteq \omega''$. Then, L diagnosable under ω' implies that L is diagnosable under ω''.

III. Language Partitions for Sensor Activation

To construct a finite solution space for the sensor activation problem, we present a method to partition a language into a finite number of subsets. A general language partition is presented first, followed by a window partition for an automaton model as a specific case.

A. General Language Partitions

The definitions of ω and θ in Section II-A are language-based. In such a model, for a system containing loops, the domain of our sensor activation policy is infinite. It is desirable to limit the richness of the sensor activation policy to a finite domain. For doing so, we partition the language L into a finite number of disjoint subsets, with the requirement that all strings within the same subset have the same last event. Then, we restrict the richness of possible sensor activation policies by requiring that all strings within each one of these subsets must have the same sensor activation decision regarding their common last event. We call such a partition a *language-based partition (LBP)*.

Formally, let Δ be a finite set whose elements are subsets of language L. Then, Δ is an LBP if its elements δ_j, $j = 0, 1, \ldots, m$, satisfy all four properties below.

1) $\delta_0 = \{\varepsilon\}$ and $\emptyset \notin \Delta$,
2) $(\forall\, \delta_j \in \Delta \setminus \{\delta_0\}) s, t \in \delta_j \Rightarrow [s = s'e \wedge t = t'e]$ for some $s', t' \in L$ and $e \in E$,
3) $\cup_{j=0}^{m} \delta_j = L$, and
4) $(\forall\, \delta_i, \delta_j \in \Delta)[\delta_i \neq \delta_j \Rightarrow\ \delta_i \cap \delta_j = \emptyset]$.

Its corresponding application to a sensor activation policy is by restricting the sensor activation policy as

$$(\forall\, \delta_i \in \Delta) se, s'e \in \delta_i \Rightarrow [e \in \omega(s) \Leftrightarrow e \in \omega(s')] \tag{14}$$

where $e \in E$ is determined by δ_i. We say that a sensor activation policy is Δ-implementable if it satisfies (14).

The choice of Δ is a trade-off between the degree of refinement in capturing the system dynamics and the computational resources needed for solving problems. With the restriction of LBP Δ, a Δ-implementable sensor activation policy, denoted by Ω, can be represented as a subset of Δ as

$$(\forall\, \delta_i \in \Delta(\setminus\{\delta_0\}))\delta_i \in \Omega \Leftrightarrow [(\forall\, se \in \delta_i)\; e \in \omega(s)] \tag{15}$$

where $e \in E$ is determined by δ_i. In other words, $\Omega \subseteq (\Delta \setminus \{\delta_0\})$ is a set that collects all δ_i, $i = 1, \ldots, m$, in which the sensor is activated for the occurrences of the last event of all strings.

Given language L and sensor activation policy $\Omega \subset \Delta$, let θ be the corresponding information mapping. The set of confusable pairs of elements of Δ, denoted by $T_{conf}(\Omega)$, is defined as

$$\begin{aligned}&T_{conf}(\Omega) = \{(\delta_i,\delta_j) \in \Delta\times\Delta : [(\exists\, se \in \delta_i,\ s'e' \in \delta_j)\\ &\theta(s) = \theta(s')] \vee [\delta_i = \delta_0 \wedge [(\exists\, s'e' \in \delta_j)\theta(s') = \varepsilon]]\\ &\vee[\delta_j = \delta_0 \wedge [(\exists\, s'e' \in \delta_i)\theta(s') = \varepsilon]] \vee \delta_i = \delta_j = \delta_0\}\end{aligned} \tag{16}$$

We note here that for all $\delta_i \in \Delta$, $(\delta_i,\delta_i) \in T_{conf}(\Omega)$.

We define the "unobservable reach" of $\delta_k \in \Delta$ under Ω, denoted by $UR(\delta_k,\Omega)$, to be

$$\begin{aligned}&UR(\delta_k,\Omega) = \{\delta_l \in \Delta :\\ &((\exists\, se, ste' \in L)\ se \in \delta_k, ste' \in \delta_l)\ \theta(s) = \theta(st)\}\end{aligned} \tag{17}$$

In words, $UR(\delta_k,\Omega)$ is a subset of Δ whose elements δ_l contain a string, say ste' for some $e' \in E$, that extends another string s with $se \in \delta_k$ such that, after s, t is unobservable. In the case where $e = e'$, by (5) for feasibility, we need to have the same sensor activation policy for e after strings s and $s' = st$. Then, by (15) for the constraints pertaining to Δ-implementability, we can characterize the feasibility of Ω as follows. Ω is feasible iff

$$\begin{aligned}&(\forall\, \delta_i,\delta_j \in \Delta)(\exists\, se \in \delta_i \wedge s'e \in \delta_j)\\ &[\theta(s) = \theta(s') \Rightarrow [\delta_i \in \Omega \Leftrightarrow \delta_j \in \Omega]]\end{aligned} \tag{18}$$

For any given Δ-implementable sensor activation policy with respect to LBP Δ, the following theorem states that, among all of its feasible and Δ-implementable subpolicies, there is a policy which is a global maximum.

Theorem 4: Consider language L with LBP Δ, and consider sensor activation policy $\Omega \subseteq \Delta$. Then, there exists a maximum feasible sensor activation policy $\Omega^{\uparrow F} \subseteq \Delta$ such that $\Omega^{\uparrow F} \subseteq \Omega$, i.e., for all feasible Ω' with $\Omega' \subseteq \Omega$, we have $\Omega' \subseteq \Omega^{\uparrow F}$. Suppose that Ω_i, $i = 1,\ldots,k$, represent all feasible sensor activation subpolicies of Ω. Then, we have $\Omega^{\uparrow F} = \cup_{i=1}^{k}\Omega_i$. Furthermore, its corresponding set of confusable string pairs is $\mathscr{T}_{conf}(\Omega^{\uparrow F}) = \cap_{i=1}^{k}\mathscr{T}_{conf}(\Omega_i)$.

B. Window Partitions for Automata Models

The deterministic finite-state automaton model of an untimed discrete event system is described as

$$G = (X,E,f,x_0) \tag{19}$$

where X is the finite set of states, E is the finite set of events, $f : X\times E \to X$ is the transition function where $f(x,e) = y$ means that there is a transition labelled by event e from state x to state y, and x_0 is the initial state. f is extended to $X\times E^*$ in the usual way: for $se \in \mathscr{L}(G)$ and $e \in E$, $f(x_0,se) = f(f(x_0,s),e)$. $\mathscr{L}(G)$ is used to denote the language generated by G.

By taking advantage of the state representation provided by automata, we present a new method to partition language $\mathscr{L}(G)$, resulting in *window partitions* that will be denoted by Δ^w hereafter. The name "window" comes from the fact that, for any element $\delta_i \in \Delta^w$, whether the string $s \in \delta_i$ or not is determined by both the state reached before the last event in s and the sequence of the last n event occurrences of s.

Let n be a given positive integer. Then, for each $\delta_i \in \Delta^w$ that contains a string se whose length is greater than or equal to n, all strings in $\mathscr{L}(G)$ that visit state $f(x_0,s)$ before the occurrence of the last event and have the same sequence of the last n events, are collected in δ_i. If $\delta_i \in \Delta^w$ contains a string whose length is smaller than n, then δ_i is a singleton. We describe mathematically window partitions as follows.

Let M be a positive integer depending on the system G. The set $\Delta^w = \{\delta_i \subseteq \mathscr{L}(G) : i = 0,\ldots,M\}$ is called an n-Window-Partition of $\mathscr{L}(G)$ if the following is true:

1) For all $u \in \mathscr{L}(G)$, if $\|u\| < n$ then $\{u\} \in \Delta^w$.
2) For all $u,v \in \mathscr{L}(G)$ with $\|u\| \geq n$ and $\|v\| \geq n$, there exists $\delta_i \in \Delta^w$, s.t. $u,v \in \delta_i$, iff there exist $e \in E$ and $s,s',t \in E^*$ with $\|t\| = n-1$, s.t. $u = ste$, $v = s'te$, and $f(x_0,st) = f(x_0,s't)$.

Lemma 1: If Δ^w is an n-Window-Partition, then Δ^w is a Language-Based-Partition (LBP).

In an n-Window-Partition, for all $e \in E$, $x \in X$, and $t \in E^*$ with $s,st \in \mathscr{L}(G)$ and $\|t\| = n-1$, (t,x,e) is used to denote $\{ste \in \mathscr{L}(G) : f(x_0,st) = x\}$. When $n = 1$, (x,e) is used to denote (ε,x,e).

Example 2: Consider G shown in Fig. 1. Then, for $\mathscr{L}(G)$, the 2-Window-Partition is

$$\begin{aligned}\Delta^w = \quad &\{\{\varepsilon\},\{f\},\{d\},\{e\},(b,0,f),(b,0,d),(b,0,e),(d,5,c),\\ &(f,1,a),(c,1,a),(e,2,a),(a,3,e),(e,4,b),(a,4,b)\}\end{aligned}$$

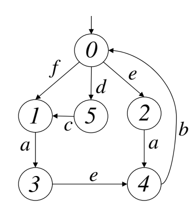

Fig. 1. Example for Window-Partition.

It can be shown that window partitions generalize the (state-based) implementability conditions used in [6], [8]. Such a generalization allows to tradeoff between the amount of computation and the achievable quality of the final solution. Therefore, for fixed computational resources, this technique can ensure a desirable balance between high solution accuracy and fast processing time.

IV. Optimization of Sensor Activation

A. Problem Statement

The problem formulation for dynamic sensor activation with an agent observing/diagnosing the system is as follows.

Given a language L together with an LBP Δ, a specification of sets of faulty events $E_f = E_{f1}\dot\cup\ldots\dot\cup E_{fl}$, and a set of observable events $E_o \subseteq E$ for the agent, suppose that under $\Omega = \{\delta_i \in \Delta : (\exists\, se \in \delta_i)\ e \in E_o\}$, the full-activation policy, the system is diagnosable.

Goal: Find a sensor activation policy $\Omega^* \subset \Delta$ such that:

1) Ω^* is feasible and, under Ω^*, the system is diagnosable.
2) Ω^* is minimal, i.e., there is no other feasible $\Omega' \subset \Omega^*$ under which the system is diagnosable.

The goal is to calculate any sensor activation policy that is a minimal solution. We do not address the selection of one minimal solution over another. This is usually application-dependent and it can be addressed in a second stage, after the above problem has been solved. In this regard, it is imperative to develop effective algorithms for calculating minimal solutions. This is the topic of the next two subsections.

B. Main Algorithm for Minimization of Sensor Activation

We present Algorithm MIN-SEN-DIAG for finding a minimal sensor activation policy Ω^* that preserves diagnosability. For the sake of generality, we present this algorithm in the context of LBPs.

Algorithm MIN-SEN-DIAG**:**
INPUT: Language L, partition Δ, set of observable events E_o, and sets of faulty events $E_{fi}, i = 1, \dots, l$.
Step 0: Initialization. Set $\Omega = \{\delta_k \in \Delta : (\exists\, se \in L)\, se \in \delta_k \wedge e \in E_o\}$ and set $D = \emptyset$.
Step 1: Pick a $\delta_k \in \Delta$ with $\delta_k \in \Omega$ but $\delta_k \notin D$. Let $\tilde{\Omega} \leftarrow \Omega \setminus \{\delta_k\}$. Then, calculate the maximum feasible sensor activation policy $\Omega^{\uparrow F}$ of all feasible subsets of $\tilde{\Omega}$.
Step 2: If $(\Delta \setminus \Omega^{\uparrow F}) \cap D = \emptyset$, test diagnosability for policy $\Omega^{\uparrow F}$. If $(\Delta \setminus \Omega^{\uparrow F}) \cap D \neq \emptyset$ or testing diagnosability fails, set $D \leftarrow D \cup \{\delta_k\}$. Otherwise, set $\Omega \leftarrow \Omega^{\uparrow F}$.
Step 3: If $\Omega \neq D$, go to Step 1. Otherwise set $\Omega^* \leftarrow \Omega$ and stop.
OUTPUT: Minimal sensor activation policy Ω^*. ■

Theorem 5: The output of Algorithm MIN-SEN-DIAG is a solution Ω^* of the minimization of sensor activation problem stated in Section IV-A.

The number of iterations of Algorithm MIN-SEN-DIAG is upper bounded by $\|\Delta\|$. However, the overall complexity for the algorithm is also dependent on its two subroutines regarding: (i) the verification of diagnosability and (ii) the calculation of the maximum feasible sensor activation policy. It is at this point that we specialize from LPBs to the case of window partitions obtained from automata models.

We have solved the problem of verification of diagnosability with respect to information mappings and developed algorithms in the case of window partitions of the language. These algorithms are of polynomial complexity in the size of the window partitions. Since these results are of independent interest, and also due to lack of space here, they will be presented elsewhere. In the remainder of this paper, we solve the problem of the second subroutine mentioned above. We present an algorithm for calculating the maximum feasible sensor activation policy in the case of a window partition.

C. Maximum Feasible Subpolicy for Window Partitions

Consider language L and an LBP Δ. For any Δ-implementable sensor activation policy $\Omega \subseteq \Delta$, by Theorem 4, there always exists a maximum feasible subpolicy $\Omega^{\uparrow F}$ of Ω such that any feasible subpolicy of Ω is a subpolicy of $\Omega^{\uparrow F}$. However, the calculation of $\Omega^{\uparrow F}$ depends on how the language L is partitioned. This is certainly a design issue that depends on the modelling formalism chosen for language L. The guideline is that a reasonable class of partitions should balance the computational effort for solving the problem with the desirable degree of refinement of the final solution. For this purpose, we focus on the case of window partitions.

For window partitions, the maximum feasible sensor activation policy $\Omega^{\uparrow F}$ of all feasible policies that are subsets of policy Ω can be found by Algorithm ↑F-WINDOW as follows.

Algorithm ↑F-WINDOW **:**
INPUT: Automaton G, sensor activation policy Ω, and window partition Δ^w.
Step 0: Initially, set $\hat{\Omega} \leftarrow \Omega$ and $T \leftarrow \{(\delta_i, \delta_i) \in \Delta^w \times \Delta^w\}$.
Step 1: If T and $\hat{\Omega}$ keep on growing, recursively set

$$\begin{aligned} &T \leftarrow T \cup \{(\delta_i, \delta_j) \in \Delta^w \times \Delta^w : \\ &(\exists\, (\delta_k, \delta_l) \in T)\ \delta_i \in UR(\delta_k, \hat{\Omega}) \wedge \delta_j \in UR(\delta_l, \hat{\Omega})\} \end{aligned} \tag{20}$$

$$\begin{aligned} &T \leftarrow T \cup \{(\delta_i, \delta_j) \in \Delta^w \times \Delta^w : (\exists (\delta_k, \delta_l) \in T)(\exists se \in \delta_k, \\ &te \in \delta_l)(\exists see', tee'' \in \mathscr{L}(G))[see' \in \delta_i \wedge tee'' \in \delta_j]\} \end{aligned} \tag{21}$$

$$\begin{aligned} &\hat{\Omega} \leftarrow \hat{\Omega} \setminus \{\delta_k \in \hat{\Omega} : (\exists\, \delta_l \in \Delta^w \setminus \hat{\Omega}) \\ &(\exists\, se, te \in \mathscr{L}(G))[(\delta_l, \delta_k) \in T]\ \wedge\ [se \in \delta_l \wedge te \in \delta_k]\} \end{aligned} \tag{22}$$

Repeat this step until $\hat{\Omega}$ and T have converged.
Step 2: Then, set $\Omega^{\uparrow F} \leftarrow \hat{\Omega}$ and $T_{conf}(\Omega^{\uparrow F}) \leftarrow T$.
OUTPUT: The maximum feasible sensor activation policy $\Omega^{\uparrow F}$ of all feasible subpolicies of policy Ω and the corresponding $T_{conf}(\Omega^{\uparrow F})$. ■

To justify Algorithm ↑F-WINDOW we need to characterize the relationship between feasibility and T_{conf}.

Lemma 2: Ω is feasible if, for arbitrary $\delta_k, \delta_l \in \Delta$ with some $se \in \delta_k$ and $te \in \delta_l$ for some $e \in E$,

$$(\delta_k, \delta_l) \in T_{conf}(\Omega) \Rightarrow (\delta_k \in \Omega \Leftrightarrow\ \delta_l \in \Omega) \tag{23}$$

Theorem 6: For a given system G and fixed positive integer n, let Ω be a sensor activation policy corresponding to an n-window partition. Then, the output $\Omega^{\uparrow F}$ of Algorithm ↑F-WINDOW satisfies Theorem 4, i.e., it is the maximum feasible sensor activation policy of all feasible and Δ^w-implementable policies Ω' that satisfy $\Omega' \subseteq \Omega$. Furthermore, the output $T_{conf}(\Omega^{\uparrow F})$ is the set of pairs of confusable elements in Δ^w under $\Omega^{\uparrow F}$.

The number of iterations of Algorithm ↑F-WINDOW is upper bounded by the size of T_{conf} plus the size of $\Omega^{\uparrow F}$. It is further upper bounded by $\|\Delta^w \times \Delta^w\| + \|\Delta^w\|$.

V. ILLUSTRATIVE EXAMPLES

In this section, we illustrate how Algorithm MIN-SEN-DIAG and Algorithm ↑F-WINDOW proceed by examples. We consider for simplicity the case of 1-Window-Partition in Examples 3 and 4. Then, by expressing the solution of Example 4 in 2-Window-Partition and reapplying Algorithm D-MACHINE to it, we show that a refinement of solution space improves the solution quality in Example 5. In this case, the set of transitions of G is the space over which optimization is performed. Instead of writing down all Ω's and D's all the time during the procedure, we use square brackets to show that a transition is removed (i.e., the sensor for that transition is not activated) and use parentheses to show that a transition cannot be removed (i.e., the sensor for that transition must be activated). The subscripts outside the square brackets and parentheses are used to mark the order in which the transitions are removed. Therefore, suppose we are removing the n^{th} transition; the current D is the set of all transitions within parentheses that have a subscript less than n, and the current Ω is set of all the transition of G minus all transitions within square brackets that have a subscript less than n.

Notation: For $(i,j) \in X \times X$, $(i,j) \in T$ means that, for all $e,a \in E$ and for all $(i,e),(j,a)$ such that $f(i,e)$ and $f(j,a)$ is defined, $((i,e),(j,a)) \in T$.

Example 3: This example illustrates Algorithm ↑F-WINDOW. Suppose the system is given by Fig. 1 as for Example 2. We consider 1-Window-Partition with $\Omega = \{(0,e),(1,a),(2,a),(5,c)\}$. By Step 0, set $\hat{\Omega} \leftarrow \Omega$, and $T = \{(x,x) : x \in X\}$.
Recursively apply Step 1 as follows. By (20), set

$$T = \{(0,1),(0,5),(1,5),(3,4),(3,0),(3,1),(3,5),(4,0),(4,1),(4,5)\} \cup \{(x,x) : x \in X\}$$

By (21), set $T \leftarrow T \cup \{(2,4)\}$. Since $(3,0) \in T$, by (22), we have $\hat{\Omega} \leftarrow \hat{\Omega} \setminus \{(0,e)\}$. By (20), $T \leftarrow T \cup \{(0,2),(1,2),(2,5),(3,2),(4,2)\}$. T does not change this time by applying (21). By (22), $\hat{\Omega}$ does not change. Calculations are exhausted for Step 1. Go to Step 2 and return $\Omega^{\uparrow F} \leftarrow \hat{\Omega}$. Finally, $\Omega^{\uparrow F} = \{(1,a),(2,a),(5,c)\}$, and $T = \{(0,1),(0,5),(1,2),(1,5),(2,5),(3,4),(3,0),(3,1),(3,5),(4,0),(4,1),(4,5),(0,2),(3,2),(4,2)\} \cup \{(x,x) : x \in X\}$. ■

Example 4: This example illustrates Algorithm MIN-SEN-DIAG. Suppose the system is given by Fig. 1 as for Example 2. We consider 1-Window-Partition first. Let f be the only faulty event and the only unobservable event. The results of the following iterations are also shown in Fig. 2.

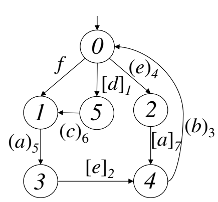

Fig. 2. Illustrative example for Algorithm MIN-SEN-DIAG.

By Step 0, set $\Omega = \{(0,d),(0,e),(5,c),(1,a),(2,a),(3,e),(4,b)\}$ and $D = \emptyset$. Iterate Step 1 to Step 2 for transitions $(0,d)$ and $(3,e)$ in this order, respectively. All of them are removed. We have $\Omega \leftarrow \Omega \setminus \{(0,d),(3,e)\}$.
Try to remove (4,b). By Step 1, set $\tilde{\Omega} \leftarrow \Omega \setminus \{(4,b)\}$. By the calculation of Example 3, corresponding $\Omega^{\uparrow F} = \{(1,a),(2,a),(5,c)\}$. Go to Step 2. The system is not diagnosable under such $\Omega^{\uparrow F}$. Set $D \leftarrow D \cup \{(4,b)\}$.
A removal of $(0,e)$ will force us to remove $(1,a)$. By Step 2, the system becomes not diagnosable. Set $D \leftarrow D \cup \{(0,e)\}$.
A removal of $(1,a)$ will force us to remove $(0,e)$. But $(0,e)$ is already in D. Set $D \leftarrow D \cup \{(1,a)\}$. By similar reason, set $D \leftarrow D \cup \{(5,c)\}$. Iterate Step 1 to Step 3 for transition $(2,a)$. We can find out that $(2,a)$ is removable.
By Step 3, we have $\Omega = \{(0,e),(5,c),(1,a),(4,b)\} = D$.
Set $\Omega^* \leftarrow \Omega$. We have $\Omega^* = \{(0,e),(5,c),(1,a),(4,b)\}$. ■

Example 5: In the 2-Window-Partition case, the policy Ω equivalent to the minimal solution Ω^* of 1-Window-Partition from Example 4 is

$$\Omega = \{\{e\},(b,0,e),(d,5,c),(c,1,a),(f,1,a),(e,4,b),(a,4,b)\}$$

Running Algorithm MIN-SEN-DIAG starting with this Ω, we can find that $(c,1,a)$ is removable and a minimal policy for 2-Window-Partition is

$$\Omega^* = \{\{e\},(b,0,e),(d,5,c),(f,1,a),(e,4,b),(a,4,b)\}$$

The solution is improved by refining the solution space. ■

VI. CONCLUSION

We formulated the problem of dynamic sensor activation in the context of event diagnosis. Algorithms were developed for the optimization of sensor activation policies that preserve the property of diagnosability. We defined the class of window partitions where we were able to trade off between the amount of computations and the achievable quality of the final solution. The problem of how to apply the results in this paper to obtain a global optimal solution for some (quantitative) cost function over dynamic observations remains open.

ACKNOWLEDGMENTS

The research was supported in part by US AFRL grant FA 8650-07-2-3744 and by NSF grants ECCS-06-24821and ECCS-0624828.

REFERENCES

[1] A. Haji-Valizadeh and K. A. Loparo, "Minimizing the cardinality of an event set for supervisors of discrete-event dynamical systems," *IEEE Trans. Automatic Control*, vol. 41, no. 11, pp. 1579–1593, 1996.

[2] S. Jiang, R. Kumar, and H. Garcia, "Optimal sensor selection for discrete-event systems with partial observation," *IEEE Trans. Automatic Control*, vol. 48, no. 3, pp. 369–381, Mar. 2003.

[3] R. Debouk, S. Lafortune, and D. Teneketzis, "On an optimization problem in sensor selection," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 12, no. 4, pp. 417–445, Oct. 2002.

[4] F. Cassez, S. Tripakis, and K. Altisen, "Sensor minimization problems with static or dynamic observers for fault diagnosis," *Seventh International Conference on Application of Concurrency to System Design*, pp. 90–99, 2007.

[5] D. Thorsley and D. Teneketzis, "Active acquisition of information for diagnosis and supervisory control of discrete event systems," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 17, no. 4, pp. 531–586, Dec. 2007.

[6] W. Wang, S. Lafortune, and F. Lin, "On the minimization of communication in networked systems with a central station," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 18, no. 4, pp. 415–443, Sept. 2008.

[7] ——, "Minimization of communication of event occurrences in acyclic discrete event systems," *IEEE Trans. Automatic Control*, vol. 53, no. 9, pp. 2197–2202, Oct. 2008.

[8] ——, "Optimal sensor activation in controlled discrete event systems," in *Proc. 47th IEEE Conf. on Decision and Control*, Dec. 2008.

[9] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete event systems," *IEEE Trans. Automatic Control*, vol. 40, no. 9, pp. 1555–1575, Sept. 1995.