INL/CON-10-19141
PREPRINT

Sensor Configuration Selection for Discrete-Event Systems Under Unreliable Observations

6th IEEE Conference on Automation Science and Engineering

Wen-Chiao Lin
Tae-Sic Yoo
Humberto E. Garcia

August 2010

Sensor Configuration Selection for Discrete-Event Systems under Unreliable Observations

Wen-Chiao Lin, Tae-Sic Yoo, and Humberto E. Garcia

***Abstract*— Algorithms for counting the occurrences of special events in the framework of partially-observed discrete-event dynamical systems (DEDS) were developed in previous work. Their performances typically become better as the sensors providing the observations become more costly or increase in number. This paper addresses the problem of finding a sensor configuration that achieves an optimal balance between cost and the performance of the special event counting algorithm, while satisfying given observability requirements and constraints. Since this problem is generally computational hard in the framework considered, a sensor optimization algorithm is developed using two greedy heuristics, one myopic and the other based on projected performances of candidate sensors. The two heuristics are sequentially executed in order to find best sensor configurations. The developed algorithm is then applied to a sensor optimization problem for a multi-unit-operation system. Results show that improved sensor configurations can be found that may significantly reduce the sensor configuration cost but still yield acceptable performance for counting the occurrences of special events.**

I. Introduction

This paper considers the monitoring architecture shown in Fig. 1 for failure/fault analysis of discrete-event dynamical systems (DEDS). In previous work, various algorithms for

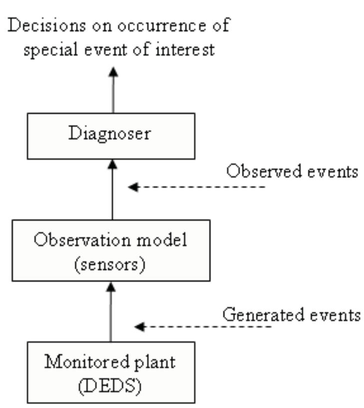

Fig. 1. Monitoring architecture

implementing the diagnoser in Fig. 1 have been developed. In particular, the work in [1] deals with detection of special events assuming that a finite-state automaton describes the DEDS, that sensors are reliable, and that failures/faults are permanent. This work is extended in [2] for diagnosing behaviors of interest in discrete event systems. Later, subsequent extensions and improvements of [1] address the problem of the detection of special events accounting for sensor unreliability and stochastic aspects in the DEDS [3]–[6]. Counting of the occurrences of intermittent or non-persistent faults that are repetitive in nature and can autonomously reset is addressed in [7]–[14]. In particular, the issue of detecting whether or not a resetting has occurred is addressed in [7], and [8] addresses fault counting problems and introduced several notions of diagnosability that capture the various counting capabilities of special events. Counting of the occurrences of special events assuming a deterministic finite-state automaton with partial observations is addressed in [9]. While [10], [11] present a deterministic counting strategy for accommodating stochastic automata with unreliable observations, [12]–[14] develop algorithms that fully utilize the probabilistic aspects of stochastic automata.

The work mentioned above focuses on developing diagnoser algorithms for analyzing the behavior of DEDS (e.g., detecting special event occurrences) given the sensor configurations. The costs of the sensor configurations may vary with the number of sensors deployed, their quality, their impact on operation, and difficulty of installation for example. In particular, sensor configurations that cost more consist of more sensors and of sensors with better quality, and, typically, they give rise to better diagnoser performance. This paper considers the problem of finding optimal sensor configurations that balance the cost of the configuration and the performance of a given diagnoser, while satisfying cost constraints and performance requirements. Without loss of generality, the diagnosers considered in this paper are implemented using stochastic counters (SCs) [12], [14], where the monitored DEDS is modeled as a stochastic automaton, with unreliable sensors under partial observations. Since the monitored system is an automaton and the sensors are unreliable, the optimization problem considered here falls into the category considered in [15]. Therefore, the optimization problem is computationally hard, and a sensor optimization algorithm based on heuristics is developed for solving it. The developed algorithm utilizes two greedy heuristics, one myopic and the other taking into account projected performances of candidate sensors. The latter heuristic is similar to the approach for optimal sensor selection developed in [16] for heterogeneous sensor networks. However, it has been accordingly modified to address the problem considered here. The developed algorithm is here applied to a sensor optimization problem on a multi-unit-operation system.

Other work on optimal senor selection for DEDS includes

W.-C. Lin, T. Yoo, and H.E. Garcia are with the Idaho National Laboratory, P.O. Box 1625, Idaho Falls, ID 83415-3605, USA. Email: {Wen-Chiao.Lin, Tae-Sic.Yoo, Humberto.Garcia}@inl.gov

[15], [17]–[20]. While [15] addresses computational issues regarding sensor selection for DEDS modeled as finite automata, [17], [18] considers optimal sensor selection for satisfying observability properties in Petri nets. Reference [19] considers the optimization problem in the framework of Fig. 1 assuming, unlike this paper, that sensors are reliable. Finally, [20] addresses optimal sensor selections for supervisory control.

The rest of this paper is organized as follows. Section II gives a brief review of the monitoring architecture shown in Fig. 1 including the algorithm for SCs. The sensor optimization problem is formulated in Section III, while a sensor optimization algorithm to solve the problem is developed in Section IV. In Section V, the developed algorithm is applied to a multi-unit-operation system. Section VI concludes the paper. We assume the reader is familiar with the terminology typical of DEDS.

II. Brief Review of Observation Platform

A. Monitored plant and observation model

The monitored plant in Fig. 1 is modeled as a stochastic automaton,

$$SA = (X, \Sigma, a, \pi_0), \tag{1}$$

where $X := \{x_1, x_2, \ldots, x_{n_x}\}$ is the finite state space, $\Sigma := \{\sigma_1, \sigma_2, \ldots, \sigma_{n_\sigma}\}$ is the set of events, and $\pi_0 := \{\pi_0(x_i) : x_i \in X\}$ is the initial probability distribution of the system. The state transition probability function a is defined as $a : X \times \Sigma \times X \to [0, 1]$, where, $a(x_i, \sigma, x_j)$ denotes the conditional probability that, given the system is in state $x_i \in X$, $\sigma \in \Sigma$ occurs and transitions the system to state $x_j \in X$. Moreover, to insure that the system is live, we assume $\forall x \in X$,

$$\sum_{i=1}^{n_\sigma} \sum_{j=1}^{n_x} a(x, \sigma_i, x_j) = 1, \tag{2}$$

i.e., the occurrence of a new transition is certain from every state. The interest is to detect and count the number of occurrences of a special event $f \in \Sigma$, which may represent a fault or an anomaly. As shown in Fig. 1, diagnosers are used to accomplish this task using observations from unreliable sensors. Assume that there is a given pool of available sensors, $\{s_1, s_2, \ldots, s_p\}$, from which sensors are chosen for observing the monitored DEDS. Let $\Delta := \{y_1, y_2, \ldots, y_{n_y}\}$ be the set of distinctive observation symbols generated from a sensor configuration $S \subseteq U$. We denote the set of observation symbols at the sensor outputs as

$$\Delta_* := \Delta \cup \{\epsilon\}, \tag{3}$$

where the symbol ϵ indicates that an event has been executed but no observation is reported. The event output function $b : \Sigma \times \Delta_* \to [0, 1]$ satisfies the following: $\forall \sigma \in \Sigma$,

$$b(\sigma, \epsilon) + \sum_{i=1}^{n_y} b(\sigma, y_i) = 1. \tag{4}$$

The functional value $b(\sigma, y)$ is the conditional probability of having output $y \in \Delta_*$ when the system executes event $\sigma \in \Sigma$. For example, assume that the given monitored system generates an event σ. Characterizing $b(\sigma, \epsilon)$, $b(\sigma, \sigma)$, and $b(\sigma, \delta)$ (with $\sigma \neq \delta$) equal to 0.1, 0.7, and 0.2 indicates the probabilities of misdetection, correct classification, and misclassification, respectively, for this sensor. The set of observation symbols, Δ, and the function b both depend on the sensor configuration, S. To further illustrate the observation model, suppose that SA executes the following event sequence: $s = \sigma^1\sigma^2 \ldots \sigma^n \ldots \in \Sigma^*$. Given s and sensor configuration, $S \subseteq U$, there are many possible sequences of output symbols for the unreliable observations modeled by (3) and (4). A particular sequence of output symbols can be denoted by $o = o^1 o^2 \ldots o^n \ldots \in (\Delta_*)^*$, where $b(\sigma^i, o^i) > 0$ for $i > 0$. Finally, the sequence of observations available to the diagnoser in Fig. 1 is denoted by $y = y^1 y^2 \ldots y^m \ldots \in \Delta^*$, where $P_\Delta(o) = y$ and $P_\Delta : (\Delta_*)^* \to \Delta^*$ is a plain projection function that removes ϵ symbol from o. Hence, y is the output sequence o with the symbol ϵ eliminated. Note that y_i denotes the ith symbol in Δ, while y^i denotes the ith observed symbol corresponding to the sequence of generated events.

B. Stochastic counters

A diagnoser for the monitoring architecture in Fig. 1 is designed based on the model of the DEDS (i.e., the stochastic automaton in (1)), the sensor observation model (i.e., the output function b in (4)) corresponding to the given sensor configuration, and the special event, f, to be counted. As the observations, y^i, $i = 1, 2, \ldots$, become sequentially available, the SC estimates the number of times f has occurred. In particular, when the mth observation becomes available, the estimate is calculated based on the observations y^1 through y^m and is denoted by $c(m)$.

Let $N_m(f)$ denote the random variable of the number of times f has occurred up to the mth available observation and $\{Y^i\}_{i=1}^m = Y^1 Y^2 \ldots Y^m$ denote the sequence of random observations. The goal of the SCs developed in [12], [14] is to calculate $c(m)$ as a function of $\{Y^i\}_{i=1}^m$ such that the mean squared error,

$$E[(N_m(f) - c(m))^2], \tag{5}$$

is minimized. The conditional expectation,

$$c(m) = E[N_m(f) | \{Y^i\}_{i=1}^m], \tag{6}$$

is a solution to minimizing (5). The SCs utilize a recursive algorithm to calculate (6). Specifically, the following information states are considered:

- $Prob(X_m | \{Y^i\}_{i=1}^m)$: conditional probability of system state just after the mth observation becomes available given the observation history up to the mth available observation;
- $E[N_m(f) | X_m, \{Y^i\}_{i=1}^m]$: conditional mean of the number of times f has occurred given the current system state just after the mth observation becomes

available and given the observation history up to the mth available observation;

- $E[(N_m(f))^2|X_m, \{Y^i\}_{i=1}^m]$: conditional second moment of the number of times f has occurred given the current system state just after the mth observation becomes available and given the observation history up to the mth available observation;

As shown in [14], these information states can be calculated recursively as the observations sequentially become available. In particular, $c(m)$ in (6) can be calculated by,

$$\begin{aligned} c(m) = E[N_m(f)|\{Y^i\}_{i=1}^m] = \\ \sum_{j=1}^{n_x}\{E[N_m(f)|X_m = x_j, \{Y^i\}_{i=1}^m = \{y^i\}_{i=1}^m]\times \\ Prob(X_m = x_j|\{Y^i\}_{i=1}^m = \{y^i\}_{i=1}^m)\}, \end{aligned} \quad (7)$$

where $\{y^i\}_{i=1}^m$ and x_j denote the realizations of $\{Y^i\}_{i=1}^m$ and X_m, respectively. Moreover, the conditional variance,

$$\begin{aligned} Var[N_m(f)|\{Y^i\}_{i=1}^m] = \\ \{\sum_{j=1}^{n_x} E[(N_m(f))^2|X_m = x_j, \{Y^i\}_{i=1}^m = \{y^i\}_{i=1}^m]\times \\ Prob(X_m = x_j|\{Y^i\}_{i=1}^m = \{y^i\}_{i=1}^m)\}- \\ E[N_m(f)|\{Y^i\}_{i=1}^m = \{y^i\}_{i=1}^m]^2, \end{aligned} \quad (8)$$

can also be calculated. The conditional variance in (8) is the mean squared error between the true count and the count estimated by the SC regarding the occurrence of the given special event. It also represents the uncertainty of the estimated special event count, and will be used to determine the performance of the SC for a given $S \subseteq U$. Let $Var(n)$ denote the variance calculated by (8) after the DEDS has executed n events. Given $S \subseteq U$, the performance of the SC is measured by the *normalized variance*,

$$\beta(S) = \lim_{n\to\infty} \frac{Var(n)}{n}, \quad (9)$$

where $\beta(S)$ depends on S and (for fixed S) is the same for any realizations of the executed events and observations. Based on simulations, the convergence of (9) holds for the practical cases considered here and in previous work [12]–[14]. Furthermore, in general, if $S_1 \subseteq S$,

$$\beta(S_1) \geq \beta(S), \quad (10)$$

i.e., a sensor configuration with more sensors gives rise to less normalized variance. Note that when there are no sensors (i.e., $S = \emptyset$), there is no way to estimate the number of occurrences of f. In this case, we set $\beta(S) = \infty$. The value, $\beta(S)$, has the following physical interpretation when $S \neq \emptyset$. It was shown in [12] that $N_m(f)$ given $\{Y^i\}_{i=1}^m$ converges to a normal distribution as the number of observations (and, hence, event executions) become large. Suppose that, on average, the system executes n_{100} events before f has been executed for the 100th time, and let

$$\gamma_{100}(S) = 2 \times \sqrt{n_{100} \times \beta(S)}. \quad (11)$$

Assuming n_{100} large enough, $\gamma_{100}(S)$ is twice the standard deviation after n_{100} event executions. Then, as f is executed the 100th time, the estimated count $c(m)$ should be within $100 \pm \gamma_{100}(S)$ with probability 0.95 (by assuming a 2σ rule). The value $\gamma_{100}(S)$ is referred to as the *uncertainty* and is used to compare performances of SCs in Section V.

Finally, if the true number of special event occurrences is known (e.g., in off-line simulations), it is possible to evaluate the performance of SCs by comparing this number and the number estimated by the SC. However, unlike the normalized variance, this comparison does not use the statistic information on the system model and sensors and may require large numbers of trials to obtain a good performance measurement. Hence, the normalized variance is chosen here as the performance measure. In addition, since there are no known closed form expressions for the normalized variance, its values are obtained via simulations.

C. Extensions to multiple special events

The description of the SCs above considers only one special event, f. Extension in the sense of [1] to estimate the occurrences of multiple events is straightforward. Let the special events be labeled as f_i, $i = 1, 2, \ldots, N_s$, where N_s is the number of special events. Given a specific sensor configuration, $S \subseteq U$, the normalized variances and uncertainties corresponding to f_i, $i = 1, 2, \ldots, N_s$, are given by $\beta^i(S)$ and $\gamma_{100}^i(S)$, respectively. Each formulation above is then extended to consider multiple f_i.

III. Problem Formulation

Consider the monitoring architecture discussed in Section II and that there are N_s special events. For a sensor configuration, $S \subseteq U$, its sensor configuration cost is

$$ct(S) = \sum_{s\in S} ct(s), \quad (12)$$

where $ct(S)$ and $ct(s)$ denote the costs of the configuration S and a particular sensor $s \in S$, respectively. Note that $ct(s)$ is a compound measure for sensor s that includes metrics such as monetary cost, vulnerability, and intrusiveness of operation. Typically, $ct(S^1) \geq ct(S^2)$ implies $\beta^i(S^1) \leq \beta^i(S^2)$, for $i = 1, 2, \ldots, N_s$, i.e., more expensive sensor configurations often provide better monitoring performance. The sensor optimization problem here is to find a sensor configuration that achieves an optimal balance between cost of the configuration and performance of the SC, while satisfying constraints on the sensor configuration cost and SC performance. For this purpose, the following loss index is considered:

$$I(S) = \sum_{i=1}^{N_s} c_i \cdot \beta^i(S) + c \cdot ct(S) \quad (13)$$

for $S \subseteq U$, where $c \geq 0$ and $c_i \geq 0$, $i = 1, 2, \ldots, N_s$, are weighting factors that indicate the importance of each term in $I(S)$. Note that optimization measures are given by $\beta^i(S)$ and $ct(S)$, while the optimization criterion is given by (13). The sensor optimization problem is formulated into the following minimization problem for $I(S)$: Find

$$S^* := \arg\min\{I(S) : S \subseteq U\}, \tag{14}$$

subject to

$$\beta^i(S) \leq \beta^{i*}, \text{ for } i = 1, 2, \ldots, N_s \text{ and } ct(S) \leq ct^*, \tag{15}$$

where

- ct^* is the maximum cost desired for S;
- β^{i*} indicates the maximum normalized variance tolerable for the diagnoser in estimating the number of occurrences of special event f_i.

As mentioned in Section I, the sensor optimization problem considered here is computationally hard, and a heuristic search algorithm is used instead for solving it.

IV. Solution to Sensor Optimization Problem

A. Motivation

A greedy algorithm is used first to solve the sensor optimization problem formulated in (14). In particular, this algorithm first sets the current sensor configuration, S_{cur}, to $\emptyset$. Then, sensors are added to S_{cur} in a step-by-step manner. The selection of the sensors in each step is based on the following criterion: Find

$$s^* = \arg\min_{s \in (U \setminus S_{cur})} \{\sum_{i=1}^{N_s} c_i \cdot \beta^i(S_{cur} \cup \{s\}) + c \cdot ct(S_{cur} \cup \{s\})\}, \tag{16}$$

subject to

$$ct(S_{cur} \cup \{s\}) \leq ct^*. \tag{17}$$

The sensor, s^*, is the newly selected sensor to be added to S_{cur}. Note that (16) is solved by searching through $s \in U \setminus S_{cur}$. Furthermore, (17) means that sensors costing more than ct^* are not considered. This criterion is based on the instantaneous performance improvement of the candidate sensor and is myopic. Stopping criteria for this greedy algorithm are based on considering whether the constraints in (15) are satisfied and on whether further reduction of the loss in (13) can be achieved in the current step. The greedy algorithm described above does not often perform well. To illustrate, consider the two simplified situations:

(a) $c_i = 0$, for $i = 1, 2, \ldots, N_s$ in (13);
(b) $c = 0$ in (13).

For (a), the optimization problem in (14) reduces to finding the least costly sensor configuration that satisfies (15). The algorithm proceeds by adding the cheapest sensor available in $U \setminus S_{cur}$ in each step. If the algorithm finds a solution, it will stop when the performance requirement in (15) is satisfied for the first time. Since cheap sensors typically provide poor performance, in most cases, the resulting sensor configuration satisfies (15) by using many cheap sensors. However, it may be possible to find a less costly sensor configuration that satisfies (15) by relying on few expensive sensors (but with good performances). A converse observation can be made for (b), where the optimization problem reduces to finding the sensor configuration that minimizes the weighted sum of variances, while satisfying (15). It is often possible to find a sensor configuration satisfying (15) with a smaller weighted sum of variances than the one calculated from the configuration found by the greedy algorithm. In both (a) and (b), the greedy algorithm needs to be improved. In case (a), the algorithm needs to realize that always picking the least costly sensor (usually with poor performance) may end up with an expensive sensor configuration. A similar argument can be made for (b). In general, performance of the above greedy algorithm can be improved if we modify the sensor selection criterion to take into account the *projected* values of the loss index (obtained by adding *fictitious* sensors with the same statistical performances and costs as the candidate sensor so as to avoid the situations described in (a) and (b)) and how far we are from satisfying or violating the constraints in (15). Hence, the proposed heuristic search algorithm for solving (14) also utilizes a sensor selection criterion that minimizes the *projected optimal* values of the loss index.

B. Projected optimal loss index sensor selection criterion

The sensor selection criterion based on minimizing the *projected optimal* values of the loss index is formulated here. To this end, the *projected* loss index and the *projected optimal* loss index are described first. Let S_{cur} stand for the current chosen sensor configuration, and consider a candidate sensor, $s \in U \setminus S_{cur}$. Let $S_{cur} \cup \{s_{1:k}\}$, $k \geq 1$, represent a *fictitious* sensor configuration, $S_{cur} \cup s \cup s \cup \ldots \cup s$, where the sensor, s, is added k times to the current sensor configuration. This fictitious configuration represents the situation, where k sensors with the same statistical performances and costs as s are added to S_{cur}. The idea is to minimize the chance of getting stuck at a local minimum by projecting the value of the loss index assuming sensors with the same statistical performances and costs as the candidate sensor are added. Furthermore, let $S_{cur} \cup \{s_{1:0}\}$ denote S_{cur}. The *projected* cost of $S_{cur} \cup \{s_{1:k}\}$ is given by

$$ct(S_{cur} \cup \{s_{1:k}\}) = \sum_{s' \in S_{cur}} ct(s') + k \cdot ct(s). \tag{18}$$

The *projected* performance of the diagnoser for $S_{cur} \cup \{s_{1:k}\}$ is denoted by $\beta^i(S_{cur} \cup \{s_{1:k}\})$ for f_i, $i = 1, 2, \ldots, N_s$. There is no known closed form expression for calculating the projected performance, and, hence, the heuristic formula,

$$\beta^i(S_{cur} \cup \{s_{1:k}\}) = \left(\frac{\beta^i(S_{cur} \cup \{s\})}{\beta^i(S_{cur})}\right)^k \beta^i(S_{cur}), \tag{19}$$

is used. Since, by (10), $\beta^i(s_{cur} \cup \{s\}) \leq \beta^i(S_{cur})$, (19) indicates that, as more sensor s are used, the projected performance of the diagnoser does not degrade and may improve exponentially. The *projected* loss index for $S_{cur} \cup \{s_{1:k}\}$ is given by

$$I(S_{cur} \cup \{s_{1:k}\}) = \sum_{i=1}^{N_s} c_i \cdot \beta^i(S_{cur} \cup \{s_{1:k}\}) + c \cdot ct(S_{cur} \cup \{s_{1:k}\}). \quad (20)$$

To calculate the *projected optimal* loss index for each candidate sensor, $s \in (U \setminus S_{cur})$, let

$$k_u(s) = \arg\max_{k \geq 0} \{ct(S_{cur} \cup \{s_{1:k}\}) \leq ct^*\}, \quad (21)$$

i.e., $k_u(s)$ is the maximum number of s that can be added to S_{cur} without the cost of $S_{cur} \cup \{s_{1:k}\}$ exceeding ct^*. The value $k_u(s)$ can be found by searching through $k \geq 0$ using (18). Likewise, let

$$k_l(s) = \arg\min_{k \geq 0} \left\{ \wedge_{i=1}^{N_s} \beta^i(S_{cur} \cup \{s_{1:k}\}) \leq \beta^{i*} \right\}, \quad (22)$$

i.e., $k_l(s)$ is the least number of s that should be added for the performance of $S_{cur} \cup \{s_{1:k}\}$ to meet all specified β_i^*. The value $k_l(s)$ can be found by searching through $k \geq 0$ using (19). If $k_l(s) \leq k_u(s)$, define

$$k^*(s) = \arg \min_{k \in [k_l(s), k_u(s)]} \{ \sum_{i=1}^{N_s} c_i \cdot \beta^i(S_{cur} \cup \{s_{1:k}\}) + c \cdot ct(S_{cur} \cup \{s_{1:k}\}) \}, \quad (23)$$

i.e., $k^*(s) \in [k_l(s), k_u(s)]$ minimizes the projected loss index in (20) given S_{cur} and s. The value $k^*(s)$ can be found by searching through $k_l(s) \leq k \leq k_u(s)$ using (12) and (19). The *projected optimal* loss index given S_{cur} and s is then

$$I(S_{cur} \cup \{s_{1:k^*(s)}\}) = \sum_{i=1}^{N_s} c_i \cdot \beta^i(S_{cur} \cup \{s_{1:k^*(s)}\}) + c \cdot ct(S_{cur} \cup \{s_{1:k^*(s)}\}). \quad (24)$$

If $k_l(s) > k_u(s)$, $k^*(s)$ and $I(S_{cur} \cup \{s_{1:k^*(s)}\})$ are not defined. Note that the constraints in (15) are used explicitly by (21) and (22). Finally, the projected optimal loss index sensor selection criterion is formulated as

$$s^* = \arg\min \{ \sum_{i=1}^{N_s} c_i \cdot \beta^i(S_{cur} \cup \{s_{1:k^*(s)}\}) + c \cdot ct(S_{cur} \cup \{s_{1:k^*(s)}\}) : s \in U \setminus S_{cur} \text{ and } k^*(s) \text{ defined} \}. \quad (25)$$

The sensor s^* is the newly selected sensor to be added to S_{cur} and can be found by searching through $s \in U \setminus S_{cur}$ using (12), (19), and (23).

Finally, note that for case (a) in Subsection IV-A, $k^*(s)$, if defined, reduces to $k_l(s)$ in (23) and (25). In this manner, always picking the cheapest sensors with similar poor performances and ending up with an expensive sensor configuration can be avoided. A similar argument can be made for case (b), where $k^*(s)$ reduces to $k_u(s)$.

C. Proposed sensor optimization algorithm

Given the weighting factors, c, c_i, $i = 1, 2, \ldots, N_s$, and performance requirements, ct^*, β^{i*}, $i = 1, 2, \ldots, N_s$, the proposed sensor optimization algorithm is as follows:

S1: Set current sensor configuration $S_{cur} = \emptyset$.

S2: Choose s^* according to the myopic criterion in (16) and set P, which stores the previous projected loss, to ∞. If no s^* can be chosen, terminate stating that constraints on the sensor cost is too stringent.

S3: Update $S_{cur} \leftarrow S_{cur} \cup \{s^*\}$. If $S_{cur} = U$, terminate (in this case, there is only one sensor in U and the optimization problem is not well formulated).

S4: Choose s^* according to the projected optimal loss index sensor selection criterion in (25). If no s^* can be chosen, terminate stating that constraints on sensor cost and diagnoser performance are too stringent.

S5: Check if the following holds:

- Stopping criterion 1 (No improvement in projected optimal loss index):

$$I(S_{cur} \cup \{s_{1:k^*(s^*)}\}) > P. \quad (26)$$

- Stopping criterion 2 (Constraints are satisfied for current configuration):

$$\beta^i(S_{cur}) \leq \beta^{i*} \quad \text{for } i = 1, 2, \ldots, N_s. \quad (27)$$

If both stopping criteria hold, terminate algorithm with optimal sensor configuration given by S_{cur}.

S6: Set $P = I(S_{cur} \cup \{s_{1:k^*(s^*)}\})$ and update $S_{cur} \leftarrow S_{cur} \cup s^*$. If $S_{cur} = U$, terminate stating that the algorithm exhausted all selections of the sensors without finding a solution. Otherwise, goto step **S4**.

Note that the first sensor chosen in **S2** is based on the myopic sensor selection criterion, while the remaining sensors chosen are based on the projected optimal loss index sensor selection criterion. The reason for this arrangement is that, since $\beta^i(\emptyset) = \infty$, for $i = 1, 2, \ldots, N_s$, we cannot compute the projected performance via (19) in order to select the first sensor. While other selection criteria may be used for **S2**, the current algorithm works well for most practical/empirical cases considered.

V. Application

A. Multi-unit-operation monitored system

The proposed sensor optimization algorithm is applied to a multi-unit-operation system shown in Fig. 2, which has been derived from an actual facility application. The unit operations UO_i of the system are labeled by numbers 1 through 6. The input ports are squares marked by $I1$, $I2$, and $I3$, while the output ports are marked by $O1$, $O2$, $O3$, and $O4$. The symbols F_i, $i = 1, 2, \ldots, 13$, stand for material flow, which may be a discrete item (e.g., container) or fluid (e.g., solution). The hexagons indicate sensors.

The operations of the system are described below. Input material, F_1, enters the monitored plant via $I1$ and is transferred to UO_1. The outputs of UO_1, F_2 and F_3, are then transferred to UO_2 and UO_4, respectively. Batches of

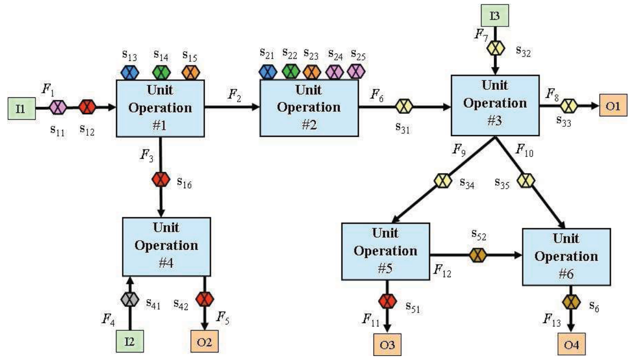

Fig. 2. A multi-unit-operation system

F_3 are stored and processed at UO_4, which are eventually outputted via F_5 through $O2$ after receiving F_4 at UO_4. Likewise, at UO_2, measurements are taken to characterize input material, and the output, F_6, is transferred to UO_3 where it interacts with F_7 entering via $I3$. While UO_3 always outputs F_8 to $O1$, it outputs either F_9 to UO_5 or F_{10} to UO_6. Material in UO_5 can either be transferred out of the monitored plant via $O3$ (i.e., F_{11}) or transferred to UO_6 (i.e., F_{12}). Materials transferred to UO_6 are eventually removed from the monitored plant via $O4$ (i.e., F_{13}).

B. Possible sensors for monitored system

A number of sensors may be deployed to monitor the multi-unit-operation system described above. They are categorized in Table I along with their possible observations.

TABLE I
POSSIBLE SENSORS FOR CONSIDERED MONITORED SYSTEM

Sensors	Discrete event observations (Σ)		
s_{13}, s_{21}	low	normal	
s_{15}, s_{23}	low	normal	high
s_{14}, s_{22}	low	normal	high
s_{12}, s_{16}, s_{42}, s_{51}	low	normal	high
s_{31}, s_{32}, s_{33}, s_{34}, s_{35}	low	normal	high
s_{52}, s_6	transfer	no transfer	
s_{41}	transfer	no transfer	
s_{25}	normal	abnormal	

C. Anomaly patterns of operations

Any form for anomaly patterns can be detected and counted by the proposed DEDS diagnosers [9]. In particular, two anomaly patterns of operations are here considered representing operations that may cause undesirable material to exit improperly. The monitoring challenge is that these patterns (described below assuming perfect sensors) are defined in terms of events separated apart in time and space.

Anomaly pattern $A1$:
A *normal* or *high* property value is provided by s_{12} *and* a *low* property indication is provided by s_{15} *and* a *high* property indication is provided by s_{16} *and* then there is an abnormal event generated by s_{25} (which compares measurements taken by s_{11} and s_{24}). If after three (3) or more instances of these anomalies have been observed, a *high* property indication is provided by s_{42} is received but *no* observation was received from s_{41}, an alarm is triggered.

Anomaly pattern $A2$:
A *low* property indication is provided by s_{31} *and* a *low* or *normal* property indication is provided by s_{33} *and* a *high* property indication is provided by s_{34}. If after three (3) or more instances of these anomalies have been observed, a *high* property indication is provided by s_{51} is received, an alarm is triggered.

D. Modeling of monitored system

In the interest of space, the procedure for modeling the multi-unit-operation system as a DEDS is omitted but outlined here. Each unit operation is first modeled as an automaton. With a slight abuse of notations, let UO_i, $i = 1, 2, \ldots, 6$ indicate the automata corresponding to the unit operations in Fig. 2. Let TV_{ij} denote the true value for the sensor s_{ij}, i.e., TV_{ij} would be the reading of s_{ij} *if* the sensor were perfect. To simplify the automata, events are constructed by aggregating these true values. For example, the automaton, UO_1, is shown in Fig. 3. For this automaton to make a state

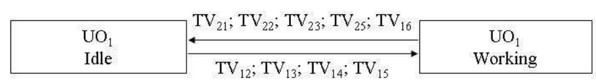

Fig. 3. Automaton for Unit Operation 1

transition from Idle to Working (or from Working to Idle), TV_{13} (or TV_{21}) must be "normal", while the other true values can assume the discrete values corresponding to those in Table I. The other unit operations are modeled similarly.

The technique introduced in [9] is employed for detecting the anomaly patterns. For each anomaly pattern to be detected, an automaton is constructed, where a fictitious unobservable event is executed if the anomaly pattern of operations occur. Let AP_1 and AP_2 denote the automata for anomaly patterns $A1$ and $A2$, respectively, and f_1 and f_2 be their respective fictitious unobservable events. The global system model is constructed by composing all unit operation models, AP_1, and AP_2:

$$UO_1 \| UO_2 \| UO_3 \| UO_4 \| UO_5 \| UO_6 \| AP_1 \| AP_2, \quad (28)$$

where $\|$ is the parallel composition described in [21][1]. The state transition probabilities of the global model is chosen suitable for the simulation study below. In practice, the transition probabilities may be obtained based on past observations. The goal of the diagnoser in Fig. 1 is to estimate the number of occurrences of f_1 and f_2, which correspond to the occurrences of anomaly patterns $A1$ and $A2$, respectively. Notice that only the operations of UO_i, $i = 1, 2, 4$, are

[1]Note that the addition of AP_1 and AP_2 in (28) does not change the behavior described by UO_1 through UO_6. The only effect is that f_1 and f_2 are executed if and only if $A1$ and $A2$ occur, respectively.

relevant to $A1$. Similarly, only UO_i, $i = 3, 5, 6$, are relevant to $A2$. Hence, the diagnoser for detecting and counting the two anomaly patterns can be constructed modularly as shown in Fig. 4, where D_1 is constructed from UO_i, $i = 1, 2, 4$, and AP_1, and estimates the number of occurrences of f_1. Likewise, D_2 is constructed from UO_i, $i = 3, 5, 6$, and AP_2, and estimates occurrences of f_2.

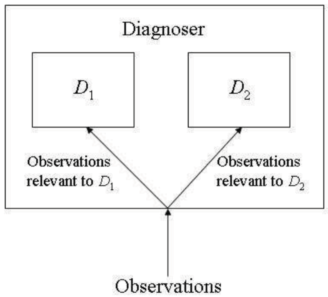

Fig. 4. Modularly constructed diagnoser

E. Sensor reliability and cost assumptions

The costs and characteristics of the available sensors for selections are indicated in Table II, with the probability of misclassification for each particular sensor being equally distributed among all possible misclassifications. For instance, consider s_{14} and suppose that the property that it is measuring is high at a given instance. The probability that this sensor does not give a reading is 0.02 and the probability that it reads "high", "normal", and "low" are 0.94, 0.02, and 0.02, respectively. Finally, from the descriptions of the sensors above, s_{11} and s_{24} are deployed if and only if s_{25} is deployed. Hence, these three sensors together are treated as one and their combined cost and characteristic are shown in the last row of the table.

TABLE II
SENSOR RELIABILITY

Sensors	Cost	Prob. of mis-detection	Prob. of correct classification	Prob. of mis-classification
s_{13}, s_{21}	3	0.03	0.94	0.03
s_{15}, s_{23}	1	0.02	0.94	0.04
s_{14}, s_{22}	3	0.02	0.94	0.04
s_{12}, s_{16} s_{42}, s_{51}	7	0.02	0.94	0.04
s_{31}, s_{32} s_{33}, s_{34} s_{35},	11	0.02	0.94	0.04
s_{52}, s_6	4	0.03	0.97	0
s_{41}	5	0	1	0
s_{11}, s_{24}, and s_{25} combined	11	0.03	0.94	0.03

F. Simulation results

The sensor configuration shown in Fig. 2 is the full sensor configuration for the multi-unit-operation system. Here, the algorithm proposed in Section IV is used to find an optimal sensor configuration for the system. The parameters used for running the algorithm are $c = 1$, $c_1 = 4000$, $c_2 = 6000$, $ct^* = 90$, $\beta^{1*} = 0.0027$, and $\beta^{2*} = 0.0035$. After numerous simulations, the systems executes, on average, 6837 and 5305 events for f_1 (anomaly pattern $A1$) and f_2 (anomaly pattern $A2$) to occur 100 times, respectively. Hence, by (11), $\beta^{1*} = 0.0027$ and $\beta^{2*} = 0.0035$ correspond to $\gamma^{1*}_{100} = 8.59$ and $\gamma^{2*}_{100} = 8.62$, respectively. Note that, typically, $\beta^1(S)$ and $\beta^2(S)$ are much smaller than the costs in Table II, and, hence, c_1 and c_2 have to be much larger than c for all terms in the loss index (13) to be comparable. An optimal sensor configuration found by the algorithm consists of the following sensors:

$$s_{15}, s_{16}, s_{41}, s_{31}, s_{34}, s_{51}, s_{52}. \tag{29}$$

Fig. 5 shows the sensor configuration in (29), where sensors selected (and not used) are crossed out. Note that sensors

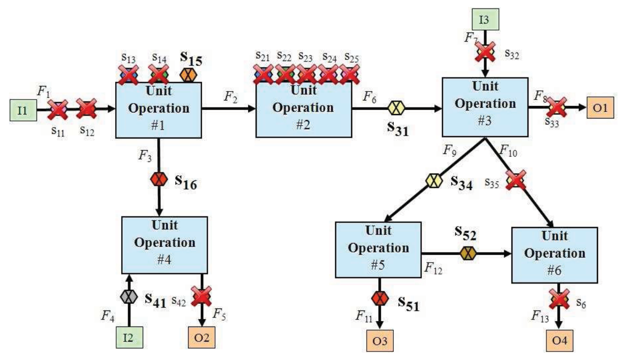

Fig. 5. Computed optimal sensor configuration

s_{15}, s_{16}, and s_{41} are used for counting occurrences of f_1, while s_{31}, s_{34}, s_{51}, and s_{52} are used for f_2. Fig. 6 plots the true and estimated (for both full and optimal sensor configurations) numbers of occurrences of f_1 against the number of event executions, while Fig. 7 plots those for f_2. In both figures, estimates from the full and optimal

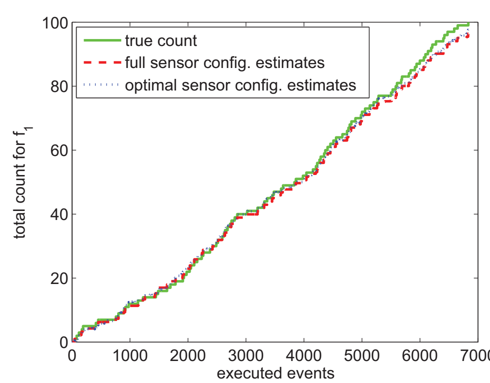

Fig. 6. Estimated vs true count of f_1 as function of executed events

sensor configurations are almost on top of each other. Note that in the particular simulation trial here, the SC tends to undercount the special event occurrences. Table III compares

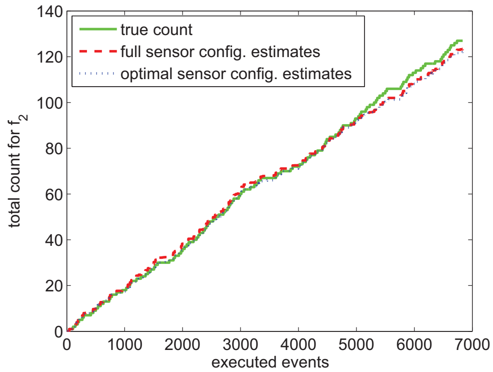

Fig. 7. Estimated vs true count of f_2 as function of executed events

the cost and performance between the full and optimal sensor configurations. From Figs. 6 and 7 and Table III,

TABLE III

COMPARISON BETWEEN FULL AND OPTIMAL SENSOR CONFIGS.

	Full sensor config.	Optimal sensor config.
Cost	121	46
$\beta^1(S)$	0.000636	0.0022
$\gamma^1_{100}(S)$	4.17	7.76
$\beta^2(S)$	0.0016	0.0021
$\gamma^2_{100}(S)$	5.83	6.68

the sensor configuration in (29) achieves cost savings of $\frac{121-46}{121} \approx 61.98\%$, while errors in estimating occurrences of f_1 and f_2 are still reasonably small. The proposed approach for sensor configuration optimization scales well in practice. For example, for the actual facility considered here consisting of 9504 states, the algorithm takes about 12 hours to compute off-line an optimal solution in a 64bit computer with Intel Xeon CPU E5520 running at 2.27 GHz.

VI. CONCLUSIONS

This paper considered the monitoring architecture in Fig. 1, where the monitored plant is modeled as a stochastic automaton, the sensors are unreliable, and an SC serves as the diagnoser. Since the the special event counting ability of an SC generally becomes better as the cost of the sensor configuration increases, a sensor optimization algorithm was developed to find a sensor configuration that optimally balances cost and the performance the SC, while satisfying given observability requirements. This algorithm adopts two greedy heuristics, one myopic and one based on projected performance of candidate sensors. These heuristics are sequentially executed in order to find optimal sensor configurations. The developed optimization algorithm can be used to compute optimal sensor configuration, although these solutions are not necessarily *the* optimal one. The developed algorithm was applied to a sensor optimization problem for a multi-unit-operation system. The results showed that optimal sensor configurations can be found that may significantly reduce the sensor configuration cost but still yield acceptable performance for counting occurrences of anomaly patterns.

ACKNOWLEDGEMENT

The research reported in this paper was supported by the U.S. Department of Energy contract DE-AC07-05ID14517.

REFERENCES

[1] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, “Diagnosability of discrete event systems,” *IEEE Trans. on Auto. Contr.*, vol. 40, no. 9, pp. 1555–1575, Sept 1995.

[2] T. Yoo and H. E. Garcia, “Diagnosis of behaviors of interest in partially-observed discrete-event systems,” *Systems & Control Letters*, vol. 57, no. 12, pp. 1023–1029, 2008.

[3] J. Lunze and J. Schrder, “State observation and diagnosis of discrete-event systems described by stochastic automata,” *Discrete Event Dyna. Syst.: Theory Appl.*, vol. 11, no. 4, pp. 319–369, 2001.

[4] D. Thorsley and D. Teneketzis, “Diagnosability of stochastic discrete-event systems,” *IEEE Trans. Autom. Control*, vol. 50, no. 4, pp. 476–492, 2005.

[5] D. Throsley, T. Yoo, and H. E. Garcia, “Diagnosability of stochastic discrete event systems under unreliable observations,” in *Proceedings of the American Control Conference*, 2008, pp. 1158 – 1165.

[6] E. Athanasopoulou, L. Lingxi, and C. Hadjicostis, “Probabilistic failure diagnosis in finite state machines under unreliable observations,” in *Proc. of 2006 8th International Workshop on Discrete Event Systems*, Ann Arbor, MI, July 2006, pp. 301 – 306.

[7] O. Contant, S. Lafortune, and D. Teneketzis, “Diagnosis of intermittent faults,” *Discrete Event Dynamic Systems: Theory and Applications*, vol. 14, no. 2, pp. 171–202, 2004.

[8] S. Jiang, R. Kumar, and H. E. Garcia, “Diagnosis of repeated/intermittent failures in discrete-event systems,” *IEEE Trans. Robotics and Applications*, vol. 19, no. 2, pp. 310–323, 2003.

[9] H. E. Garcia and T. Yoo, “Model-based detection of routing events in discrete flow networks,” *Automatica*, vol. 4, no. 41, pp. 583–594, 2005.

[10] T. Yoo and H. E. Garcia, “New results on discrete-event counting under reliable and unreliable observation information,” in *Proc. of 2005 IEEE Int. Conf. on Networking, Sensing and Control*, 2005.

[11] ——, “Event counting of partially-observed discrete-event systems with uniformly and nonuniformly bounded diagnosis delays,” *Discrete Event Dynamic Systems*, vol. 19, no. 2, pp. 167–187, 2009.

[12] ——, “Stochastic event counter for discrete-event systems under unreliable observations,” in *Proceedings of the American Control Conference*, 2008, pp. 1145 – 1152.

[13] W.-C. Lin, H. E. Garcia, D. Thorsley, and T. Yoo, “Sequential window diagnoser for discrete-event systems under unreliable observations,” in *Proc. of 47th Annual Allerton Conf. on Communication, Control, and Computing*, Monticello, IL, Sept. 2009, pp. 668–675.

[14] T. Yoo, D. Thorsley, W.-C. Lin, and H. E. Garcia, “Recursive stochastic event counting for discrete-event systems under unreliable observations,” submitted to *IEEE Tran. on Autom. Control*, 2009.

[15] T. Yoo and S. Lafortune, “NP-completeness of sensor selection problems arising in partially-observed discrete-event systems,” *IEEE Trans. Automat. Contr.*, vol. 47, no. 9, pp. 1495–1499, 2002.

[16] T. Yoo and H. E. Garcia, “Sensor deployment optimization for network intrusion detection,” in *Proc. of 44th Annual Allerton Conf. on Communication, Control, and Computing*, Monticello, IL, Sept. 2006, pp. 996–1004.

[17] L. Aguirre-Salas, O. Begovich, and A. Remírez-Treviño, “Optimal sensor choice for observability in free-choice Petri nets,” in *Proc. of IEEE Int. Symposium on Intelligent Control*, Mexico City, Mexico, Sept. 2001, pp. 270–275.

[18] L. Aguirre-Salas, “Sensor selection for observability in interpreted Petri nets: a generic approach,” in *Proc. of 42nd IEEE Conf. on Decision and Control*, Hawaii, Dec. 2003, pp. 3760–3765.

[19] S. Jiang, R. Kumar, and H. E. Garcia, “Optimal sensor selection for discrete event systems under partial observation,” *IEEE Trans. Automat. Contr.*, vol. 48, no. 3, pp. 369–381, 2003.

[20] K. R. Rohloff, S. Kuller, and G. Kortsarz, “Approximating the minimal sensor selection for supervisory control,” *Discrete Event Dyn. Syst.: Theory and Appl.*, vol. 16, no. 2, pp. 143–170, 2006.

[21] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.