

# Sequential Window Diagnoser for Discrete- Event Systems Under Unreliable Observations

Allerton Conference 2009

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

September 2009

The INL is a  
U.S. Department of Energy  
National Laboratory  
operated by  
Battelle Energy Alliance



This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint should not be cited or reproduced without permission of the author. This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, or any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, of any information, apparatus, product or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights. The views expressed in this paper are not necessarily those of the United States Government or the sponsoring agency.

# Sequential Window Diagnoser for Discrete-Event Systems Under Unreliable Observations

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

**Abstract**—This paper addresses the issue of counting the occurrence of special events in the framework of partially-observed discrete-event dynamical systems (DEDS). Developed diagnosers referred to as sequential window diagnosers (SWDs) utilize the stochastic diagnoser probability transition matrices developed in [9] along with a resetting mechanism that allows on-line monitoring of special event occurrences. To illustrate their performance, the SWDs are applied to detect and count the occurrence of special events in a particular DEDS. Results show that SWDs are able to accurately track the number of times special events occur.

## I. INTRODUCTION

The failure/fault analysis of discrete-event dynamical systems has received attentions from academia and industries since the seminal work [7] was published. The framework presented in [7] deals with the detection of special events where the finite-state automaton describes the dynamic of the system and the associated partial-observation is assumed to be reliable. Later, this work was extended in [11] for diagnosing behaviors of interest in discrete event systems. The behaviors of interest in this work were modeled as languages generated by the discrete event model.

Among the subsequent extensions and improvements of [7], recent developments on the detection problem of special events accounting for sensor unreliability and stochastic aspects in discrete-event systems include [6], [8], [9], [1]. In [6], the authors show that, in general, the observer of a finite-state stochastic automaton cannot be represented by another finite-state stochastic automaton. Assuming reliable sensors, the authors in [8] introduced the notions of stochastic diagnosability that incorporate the stochastic automaton describing the behavior of the system. These notions of diagnosability relax that of deterministic automaton introduced in [7]. Also presented in [8] is the procedure of building the stochastic diagnoser<sup>1</sup> that bears a similar structure of the logical diagnoser of [7]. The transitions of the stochastic diagnoser include probability transition matrices that can be used to update the probability distribution on the state estimate. Later, this work is extended to unreliable sensors in [9]. In [1], the authors present a method of calculating the observation likelihood of the stochastic automaton. The authors then used the developed procedure to decide the most-likely stochastic automaton explaining the observed output sequence among the candidate stochastic automata.

One can find attempts addressing special events with repeatable nature in [3], [5], [10], [4], [13]. Intermittent

or non-persistent faults are repetitive in nature and can autonomously reset. The issue of detecting whether or not a resetting has occurred was addressed in [3]. In [5], the authors addressed fault counting problems and introduced several notions of diagnosability that capture the various counting capabilities of special events.

Though results on probabilistic detection/diagnosis for stochastic automata and event counting for deterministic automata are available, results on the “probabilistic counting” of special events for stochastic automata are limited at best. In [10], [13], a counting strategy accommodating stochastic automata and unreliable observations was presented. However, the counting strategy of [10], [13] is deterministic in that the presented counting algorithm searches the minimum count of the associated state estimate rather than using the probabilistic distribution of state estimate of the stochastic automaton; essentially, it deals with possibility rather than probability. In [12], the authors fully utilized the probabilistic aspect of stochastic automata in developing algorithms for special event counting. In this paper, we develop *sequential window diagnosers* (SWDs) utilizing the notions of state probability vector and stochastic diagnoser probability transition matrices from [9] together with a resetting mechanism that allows for counting the occurrence of special events. The transition matrices can be calculated and stored offline before the monitoring begins. Special event counting involves only simple vector-matrix multiplications, which is computationally simpler than the recursive calculations in [12]. The main contributions of this paper are summarized below.

- A procedure for counting the occurrence of special events for DEDS under unreliable observations is developed based on the stochastic diagnoser probability transition matrices introduced in [9] and a resetting mechanism.
- We apply the developed methodology to a material flow system where special routing events are counted dynamically.

The rest of the paper is organized as follows. In Section II, necessary notation and definitions are introduced. Section III briefly reviews sensor output automata and stochastic diagnoser probability transition matrices, on which SWDs are based. The SWDs are also defined in this section. In Section IV, we give an illustrative material flow system application where SWD estimates the number of occurrences of special routing events under unreliable observations. Finally, Section V concludes the paper. We assume in the remainder of this paper that the reader is familiar with terminologies

<sup>1</sup>The stochastic diagnoser is not a stochastic automaton. Therefore, the results presented in [6], [8] do not contradict each other.

typical of DEDS.

## II. PRELIMINARIES

The monitoring architecture considered in this paper is shown in Fig. 1. The monitored plant is modeled as a

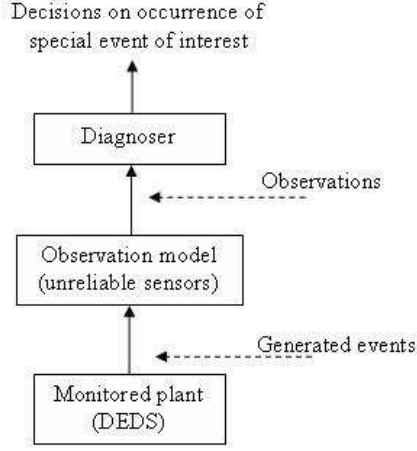


Fig. 1. Monitoring architecture

DEDS, which generates events. The events are observed by unreliable sensors, which give rise to the observation model. Finally, the diagnoser uses the observations from the unreliable sensors to infer the occurrence of special events. In Sections II-A and II-B, the DEDS and observation models are described in detail, respectively. The diagnoser is described in Section III.

### A. DEDS Model

The monitored DEDS in Fig 1 is a stochastic automaton  $SA$ , which is defined as a quadruple:

$$SA = (X, \Sigma, a, \pi_0) \quad (1)$$

where  $X := \{x_1, x_2, \dots, x_{n_x}\}$  is the finite state space,  $\Sigma := \{\sigma_1, \sigma_2, \dots, \sigma_{n_\sigma}\}$  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 below:

$$a : X \times \Sigma \times X \rightarrow [0, 1], \quad (2)$$

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  $\forall x \in X$ ,

$$\sum_{i=1}^{n_\sigma} \sum_{j=1}^{n_x} a(x, \sigma_i, x_j) = 1, \quad (3)$$

i.e., the occurrence of a new transition is certain from every state. Transition probability function can be extended to strings according to the equation,

$$a(x_i, \sigma s, x_j) = \sum_{x' \in X} a(x_i, \sigma, x') a(x', s, x_j). \quad (4)$$

The probability of a given string occurring when the  $SA$  is in state  $x_i$  is defined as

$$Pr(x_i, s) \triangleq \sum_{x' \in X} a(x_i, s, x'). \quad (5)$$

The language generated from state  $x_i$  is

$$L(SA, x_i) = \{s \in \Sigma^* : Pr(x_i, s) > 0\}, \quad (6)$$

where for any finite set  $S$ ,  $S^*$  denotes the Kleene closure of  $S$ .

### B. Observation Model

Next, the observation model in Fig. 1 is considered. In practice, sensors are seldom perfect, and hence, observations from sensors are unreliable. The unreliable observations are modeled in the following manner. Let  $\Delta := \{y_1, y_2, \dots, y_{n_y}\}$  be the set of distinctive observation symbols. We denote the set of observation symbols at the sensor outputs as

$$\Delta_* := \Delta \cup \{\epsilon\}, \quad (7)$$

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

$$b(\sigma, \epsilon) + \sum_{i=1}^{n_y} b(\sigma, y_i) = 1. \quad (8)$$

The functional value  $b(\sigma, y)$  is the conditional probability of having output  $y \in \Sigma \times \Delta_*$  when the system executes event  $\sigma \in \Sigma$ .

Consider the following sequence of state transitions by the DEDS in Fig. 1:

$$(x_s^1, \sigma^1, x_d^1) \dots (x_s^n, \sigma^n, x_d^n) \dots \in (X \times \Sigma \times X)^*, \quad (9)$$

where  $x_s^i, x_d^i \in X$  and  $\sigma^i \in \Sigma$  for  $i > 0$  and  $(x_s^i, \sigma^i, x_d^i)$  denotes that the DEDS makes a state transition from  $x_s^i$  to  $x_d^i$  by executing the event  $\sigma^i$ . Obviously, the following has to be satisfied:

- $\pi_0(x_s^1) > 0$ ;
- $a(x_s^i, \sigma^i, x_d^i) > 0, \forall i > 0$ ;
- $x_s^i = x_d^{i+1}, \forall i > 0$

The sequence of generated events for the DEDS in Fig. 1 corresponding to the state transitions in (9) is denoted by the string:

$$s = \sigma^1 \sigma^2 \dots \sigma^n \dots \in \Sigma^*. \quad (10)$$

Given the sequence of events,  $s$ , there are many possible sequences of output symbols for the sensors modeled by (7) and (8). A particular sequence of output symbols can be denoted by

$$o = o^1 o^2 \dots o^n \dots \in (\Delta_*)^*, \quad (11)$$

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

$$y = y^1 y^2 \dots y^m \dots \in \Delta^*, \quad (12)$$

where

$$P_\Delta(o) = y, \quad (13)$$

and  $P_\Delta : (\Delta^*)^* \rightarrow \Delta^*$  is a plain projection function that removes  $\epsilon$  symbol from  $o$ .

### C. Special Event

In the remainder of the paper, we are interested in developing a diagnoser (see Fig. 1) for estimating the number of occurrences of some special event  $f \in \Sigma$  (e.g., a fault or a failure) given the available observations in (12).<sup>2</sup> If the special event  $f$  is an element of a string  $s$ , we write  $f \in s$ .

### D. Example

An example is given below illustrating the notations presented above. Consider the stochastic automaton  $SA$  and the associated unreliable output function  $b$  described in Fig. 2.

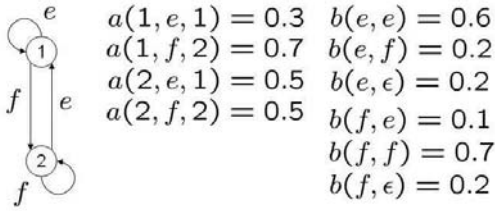


Fig. 2. Stochastic automaton and unreliable output function

Consider a sample state transition of the system and corresponding sequences of generated events and observations.

State transitions	:	(1, e, 1)(1, f, 2)(2, f, 2)(2, e, 1)...
s	:	e f f e ...
o	:	e e $\epsilon$ f ...
y	:	e e f ...

Note that the third state transition  $(2, f, 2)$  of the system does not generate an observation symbol in  $\Delta$ . Thus, system transitions incurred by  $f$  occurred twice, while  $eeff$  is observed (i.e.,  $f$  is observed only once). The sequential window diagnoser will use the observations,  $eeff \dots$ , to logically infer the number of occurrences of  $f$ .

## III. SEQUENTIAL WINDOW DIAGNOSERS

In this paper, the diagnoser in Fig. 1 is the sequential window diagnoser (SWD) described in this section. SWDs are based on the stochastic diagnoser probability transition matrices developed in [9] and a resetting mechanism that permits counting special event occurrences. A brief discussion of the development of the probability transition matrices is given here for completeness.

<sup>2</sup>For simplicity, we only consider a single event  $f \in \Sigma$  to be counted. Extension to multiple events or multiple types of multiple events in the sense of [7] is straightforward.

### A. Sensor Output Automaton

We start with the sensor output automaton (SOA), which is necessary to define the stochastic diagnoser probability transition matrices.

The SOA for a DEDS under unreliable observations modeled in Section II is a stochastic automaton denoted by:

$$SOA = (X^{SOA}, \Sigma^{SOA}, a^{SOA}, \pi_0^{SOA}), \quad (14)$$

where the constituent elements are explained as follows. The state space,  $X^{SOA}$ , and initial probability distribution,  $\pi_0^{SOA}$ , are identical to those of  $SA$ , i.e.,  $X^{SOA} = X$  and  $\pi_0^{SOA} = \pi_0$ . The event set,  $\Sigma^{SOA} = \Delta \cup \Delta_f \cup \{\epsilon, \epsilon_f\}$ , consists of two versions of the output symbols  $\Delta \cup \{\epsilon\}$ : one corresponding to “normal” events and the other corresponding to the “special” event  $f$ .

The probability transition functions  $a^{SOA}$  are constructed according to the following equations:

$$a^{SOA}(x_1, y, x_2) = \sum_{\sigma \in \Sigma / \{f\}} (a(x_1, \sigma, x_2) \times b(\sigma, y)) \quad (15)$$

if  $y \in \Delta \cup \{\epsilon\}$ , and

$$a^{SOA}(x_1, y_f, x_2) = a(x_1, f, x_2) \times b(f, y) \quad (16)$$

if  $y_f \in \Delta_f \cup \{\epsilon_f\}$ .

A deterministic observation model for the SOA is defined as:

$$b^{SOA}(y, y) = 1 \quad \text{if } y \in \Delta \cup \{\epsilon\}, \quad (17)$$

$$b^{SOA}(y_f, y) = 1 \quad \text{if } y_f \in \Delta_f \cup \{\epsilon_f\}. \quad (18)$$

Thus, “normal” events in  $\Delta \cup \{\epsilon\}$  are observed directly, while “special” events in  $\Delta_f \cup \{\epsilon_f\}$  appear identical to their corresponding normal events.

To construct stochastic diagnoser probability transition matrices, define the set of all strings generated from state  $x_i$  whose only observable event is the final event  $\delta \in \Delta \cup \Delta_f$  as

$$L_\delta(SOA, x_i) \triangleq \{s \in L(SOA, x_i) : s = u\delta, u \in (\{\epsilon\} \cup \{\epsilon_f\})^*\} \quad (19)$$

The SOA, along with its observation model, combines the information provided by the model of the DEDS, its associated unreliable observations, and special event  $f$ .

### B. Stochastic Diagnoser Probability Transition Matrices

The stochastic diagnoser probability transition matrices developed in [9] are described below with modified notions for simplicity. We note here that the stochastic diagnoser in [9] is not a stochastic automaton. The reason is that the state transition probabilities of the stochastic diagnoser cannot be characterized by usual Markov chain probability transition matrices (where each row sums up to one) but by the matrices described below.

Consider the DEDS in (1) under unreliable observations with an  $SOA$  in (14). For  $m \geq 0$ , define the vector,

$$\phi(m) = [p_1^N(m), p_2^N(m), \dots, p_{n_x}^N(m), p_1^F(m), \dots, p_{n_x}^F(m)], \quad (20)$$

where  $p_i^N(m)$  (or  $p_i^F(m)$ ) denotes the conditional probability that the DEDS is in state  $x_i \in X$  and that  $f$  has not been (or has been) executed given  $m$  observations and given initial probability  $\phi(0) = [\pi_0, 0, 0, \dots, 0]$ . Recall that  $\pi_0$  is the initial state distribution probability given in (1).

The stochastic diagnosis probability transition matrices form a set of  $2n_x \times 2n_x$  matrices denoted by  $\Phi^{SD}$ . These matrices update  $\phi(m)$  as observations become sequentially available and are described below. Each  $\delta \in \Delta$  is associated with a matrix in  $\Phi^{SD}$  defined by

$$\Phi^{SD}(\delta) = \begin{bmatrix} \Phi^{NN}(\delta) & \Phi^{NF}(\delta) \\ 0_{n_x \times n_x} & \Phi^{FF}(\delta) \end{bmatrix}, \quad (21)$$

where  $0_{n_x \times n_x}$  is the  $n_x \times n_x$  zero matrix and  $\Phi^{NN}(\delta)$ ,  $\Phi^{NF}(\delta)$ , and  $\Phi^{FF}(\delta)$  are defined by

$$\begin{aligned} \Phi_{ij}^{NN}(\delta) &= \sum_{s \in L_\delta(SOA, x_i): f \notin s} a^{SOA}(x_i, s, x_j), \\ \Phi_{ij}^{NF}(\delta) &= \sum_{s \in L_\delta(SOA, x_i): f \in s} a^{SOA}(x_i, s, x_j), \\ \Phi_{ij}^{FF}(\delta) &= \sum_{s \in L_\delta(SOA, x_i)} a^{SOA}(x_i, s, x_j). \end{aligned} \quad (22)$$

Using the expressions given by (22) directly to calculate the entries of  $\Phi^{SD}$  is not practical for two major reasons. First, there may be cycles of unobservable events in the SA. Second, for each  $x \in X^{SOA}$  and  $\sigma \in \Sigma^{SOA}$ , there may be non-unique  $x', x'' \in X^{SOA}$  such that  $a^{SOA}(x, \sigma, x') > 0$  and  $a^{SOA}(x, \sigma, x'') > 0$ . The above reasons will make (22) difficult to evaluate directly. A matrix equation to calculate the entries of  $\Phi^{SD}$  based on the absorption probabilities of Markov chains is derived in [9] and described below. Define the matrix  $Q$  as the following. For each  $\delta \in \Delta \cup \{\epsilon\}$ , let

$$Q_{ij}(\delta) = a^{SOA}(x_i, \delta, x_j), \quad (23)$$

and, similarly, for each  $\delta_f \in \Delta_f \cup \{\epsilon_f\}$ , let

$$Q_{ij}(\delta_f) = a^{SOA}(x_i, \delta_f, x_j). \quad (24)$$

For each  $\delta$ , we combine  $Q_{ij}(\delta)$  and  $Q_{ij}(\delta_f)$  to give

$$\hat{Q}(\delta) = \begin{bmatrix} Q(\delta) & Q(\delta_f) \\ 0_{n_x \times n_x} & Q(\delta) + Q(\delta_f) \end{bmatrix}. \quad (25)$$

Matrix  $\Phi^{SD}(\delta)$  is then the solution to the matrix equation

$$\Phi^{SD}(\delta) = \hat{Q}(\delta) + \hat{Q}(\epsilon)\Phi^{SD}(\delta). \quad (26)$$

The matrices  $\Phi^{SD}(\delta)$  are precalculated and stored offline for use by the sequential window diagnoser.

From the development in [8] and [9], given the  $(m+1)$ st observation  $\delta \in \Delta$ ,  $\phi(m+1)$  can be calculated from  $\phi(m)$  as follows:

$$\phi(m+1) = \frac{\phi(m)\Phi^{SD}(\delta)}{\|\phi(m)\Phi^{SD}(\delta)\|}, \quad (27)$$

where  $\|\bullet\|$  denotes the 1-norm.

### C. Sequential Window Diagnoser Algorithm

Let  $c(m)$  denote the estimated count for the number of times special event  $f$  has been executed after the  $m$ th observation becomes available. The algorithm for SWDs is described next.

**S1:** Select a *false alarm tolerance*  $\alpha$ . Set count  $c(0) = 0$ .

**S2:** Initialize the state probability vector in (20) such that

$$\phi(0) = [\pi_0, 0, 0, \dots, 0]. \quad (28)$$

**S3:** Suppose the next observation is the  $(m+1)$ st,  $m \geq 0$ , observation. As this observation becomes available, update

$$\begin{aligned} \phi(m+1) &= \\ &[p_1^N(m+1), p_2^N(m+1), \dots, p_{n_x}^N(m+1), \\ & p_1^F(m+1), p_2^F(m+1), \dots, p_{n_x}^F(m+1)] \end{aligned}$$

according to (27), i.e.,

$$\phi(m+1) = \frac{\phi(m)\Phi^{SD}(\delta)}{\|\phi(m)\Phi^{SD}(\delta)\|}.$$

If the summation

$$\sum_{i=1}^{n_x} p_i^F(m+1) > 1 - \alpha, \quad (29)$$

do the following: set

$$c(m+1) = c(m) + 1, \quad (30)$$

and *reset*  $\phi(m+1)$  such that

$$p_i^N(m+1) = p_i^N(m+1) + p_i^F(m+1), \quad (31)$$

and

$$p_i^F(m+1) = 0, \quad (32)$$

for  $1 \leq i \leq n_x$ . If (29) does not hold, set

$$c(m+1) = c(m). \quad (33)$$

**S4:** Go back to **S3**.

We note that the reset equations (31) and (32) mean that, after an estimated occurrence of  $f$  is counted, the probability that  $f$  has occurred is reset to zero, while the probability that the DEDS is in state  $x_i \in X$  is preserved. Hence, the diagnoser is still able to estimate the system state after the reset.

An SWD, therefore, starts with an initial belief of state probabilities given by the vector in (28). As observations become available, it updates this state probability vector according to (27) until the conditional probability (given the observations) that  $f$  has occurred (calculated by the



summation in the left hand side of (29)) is greater than  $1 - \alpha$ . Then, the SWD increases the estimated count by one via (30), and resets state probability vector by (31) and (32). After the reset, the probability that  $f$  has occurred is zero (i.e., the summation on the left hand side of (29) is zero), while the probability that the DEDES is in a particular state  $x_i \in X$  remains the same before and after the resetting. The above process of updating the estimated count, then, is repeated with the state probability vector after reset.

The parameter,  $\alpha$ , is the upper bound of the probability that the estimated count is increased via (30) but  $f$  has not been executed since the last increase of the count. This parameter also determines the amount of evidence required for increasing the estimated count. It can be concluded from S3 of the SWD algorithm that, if a very small  $\alpha$  is chosen, a large number of observations has to be available before the estimated count is increased. In this case,  $f$  may have been executed more than once before the estimated count is increased by one, and the SWD tends to undercount. Conversely, if  $\alpha$  is very large, a small number of observations (possibly from events irrelevant to execution of  $f$ ) may cause the estimated count to increase. In this case,  $f$  may not have been executed when the estimated count is increased, and the SWD tends to overcount. Selection of the best  $\alpha$  for optimal performance is largely dependent on the DEDES and its sensor configuration (for generating the observations). There is currently no closed form expression for finding the best  $\alpha$  for a given system. A good  $\alpha$  is found by comparing computer simulation results using various values for  $\alpha$ .

Finally, we refer to the observations in between increases of the estimated count as observation windows. The SWD utilizes the state probability update (27) in the windows to sequentially update estimated count. Hence, the name *sequential window diagnosers*.

#### IV. APPLICATION

For the application considered here, the monitored plant of Fig. 1 is a material flow system depicted in Fig. 3. This

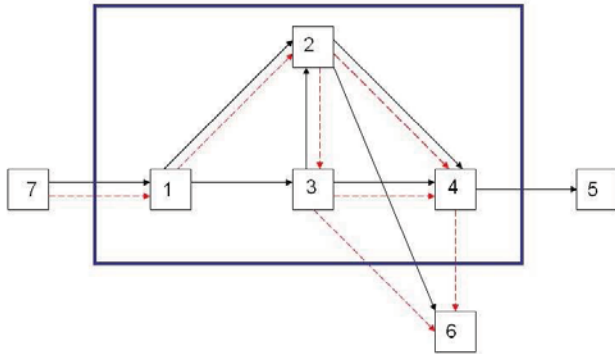


Fig. 3. Material Flow System

system corresponds to a manufacturing facility where two types of materials are processed. The possible routes of the

first (second) type of material is depicted with arrowed plain (dotted) lines.

The numbered rectangles represent Input/Working/Output stations where materials is processed and transfer to/from. Station 7 represents an input station where materials enter the facility. Stations 1, 2, 3, and 4 are internal material processing stations. Stations 5 and 6 are output stations where the processed materials leave the facility.

The next subsection describes the monitored plant as a DEDES considered in Section II-A.

##### A. Stochastic Automata System Model

A set of automata is used to model the material flows among stations. For example, the automaton in Fig. 4 captures the material flow at station 1. State  $(W_i, j, k)$  means that internal material processing station  $i$  has  $j$  number of material type 1 and  $k$  number of material type 2. Event  $(i, j, k)$  means that a material of type  $k$  is moved from station  $j$  to station  $i$ . For simplicity, we assume one buffering capacity for all internal material processing stations.<sup>3</sup> Therefore, for instance, after event  $(1, 7, 1)$ , the feasible events for  $W_1$  are the transportation events of material type 1 from station 1 to some other stations, that is,  $(2, 1, 1)$  and  $(3, 1, 1)$ . The automata models  $O_5$  and  $O_6$  for output stations 5 and 6 are depicted in Figs. 5 and 6, respectively. As one can observe, output stations 5 and 6 are assumed to take materials indefinitely. For input station 7, we assume that there are infinite number of materials to be processed. The automaton model of input station 7 is denoted by  $I_7$  and depicted in Fig. 7.

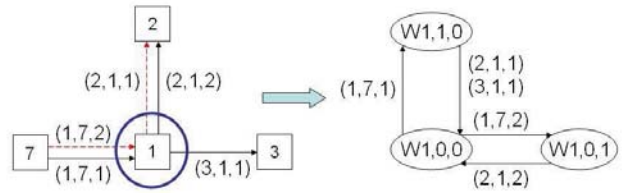


Fig. 4. Working Station Automaton Model:  $W_1$



Fig. 5. Output Station Automaton Model:  $O_5$

The global system model is constructed by composing all component models:

$$SA := W_1 || W_2 || W_3 || W_4 || O_5 || O_6 || I_7$$

where  $||$  is the parallel composition operator as in [2].

<sup>3</sup>Modeling buffering capabilities may amount to introduce more states depending on the given material processing policies (FIFO, LIFO, etc.) and define appropriate transitions.

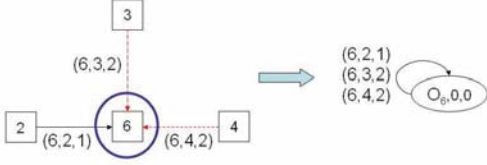


Fig. 6. Output Station Automaton Model:  $O_6$



Fig. 7. Input Station Automaton Model:  $I_7$

Constructing transition probability function  $a$  may require the assessments of experts and/or samplings from system operations. Here, for simplicity, we assume that the transition probability is equally-likely for all active events of a given state  $x$ . That is,  $\forall x \in X, \sigma \in \Sigma$ ,

$$a(x, \sigma, x') = \frac{1}{|act(x)|},$$

where  $act(x) = \{\sigma \in \Sigma : \exists x' \in X \text{ s.t. } a(x, \sigma, x') > 0\}$  and  $|act(x)|$  denotes the cardinality of  $act(x)$ .

Assume that, initially, the working stations of the material flow system do not have materials being processed. That is, the initial probability distribution is  $\pi_0(x) = 1$  when

$$\pi_0(x) = \begin{cases} 1 & \text{if } x = (W_{1,0}, \dots, W_{4,0}, O_{5,0}, O_{6,0}, I_{7,\infty}) \\ 0 & \text{o.w.} \end{cases}$$

where

$$\begin{aligned} W_{i,0} &:= (W_i, 0, 0), \quad i = 1, \dots, 4; \\ O_{i,0} &:= (O_i, 0, 0), \quad i = 5, 6; \\ I_{7,\infty} &:= (I_7, \infty, \infty). \end{aligned}$$

## B. Sensor Models

In this subsection, we describe the observation model in Fig 1 for this application.

Three types of sensors can be installed at internal working stations,  $W_1, W_2, W_3$ , and  $W_4$ , and output stations,  $O_5$  and  $O_6$ . These three sensor types are motion sensors, tag sensors, and material type sensors. Motion sensors are able to identify the first attribute of the executed event (i.e., identify that a material is transferring to the sensor's location). Tag sensors are able to identify the second attribute of the executed event (i.e., identify the previous location of the material transferring to the sensor's location). Finally, material type sensors are able to identify the third attribute of the executed event (i.e., identify the material type of a material transferring to the sensors's location). In this specific application, only motion and material type sensors are considered. Unidentified attributes of events are marked with \*. We consider the following two sensor configurations.

**(Case 1)** Internal working stations,  $W_1, W_2$ , and  $W_3$

are installed with motion sensors only, while  $W_4$  and  $O_6$  are installed with both motion and material type sensors. The sensor characteristics are such that, for  $i = 1, 2, 3$ ,

$$b_1((i, j, k), (i', *, *)) = \begin{cases} 0.87 & \text{if } i' = i \ (\checkmark_1), \\ 0.01 & \text{if } i' \in \{1, 2, 3, 4\} \setminus \{i\} \ (\dagger_1), \end{cases}$$

$$b_1((i, j, k), \epsilon) = 0.1 \ (\ddagger_1)$$

For  $i = 4, 6$ ,

$$b_1((i, j, k), (i, *, k')) = \begin{cases} 0.8 & \text{if } k' = k \ (\checkmark\checkmark_1), \\ 0.1 & \text{if } k' \in \{1, 2\} \setminus \{k\} \ (\dagger\dagger_1), \end{cases}$$

$$b_1((i, j, k), \epsilon) = 0.1 \ (\ddagger\ddagger_1).$$

For  $i \notin \{1, 2, 3, 4, 6\}$ ,  $b_1((i, j, k), \epsilon) = 1$ .

**(Case 2)** Motion sensors are installed at all internal working stations, while no material type sensors are installed. The sensor characteristics are such that, for  $i \in W := \{1, 2, 3, 4\}$ ,

$$b_2((i, j, k), (i', *, *)) = \begin{cases} 0.87; & \text{if } i' = i; \ (\checkmark_2) \\ 0.01, & \text{if } i' \in W \setminus \{i\}; \ (\dagger_2) \end{cases}$$

$$b_2((i, j, k), \epsilon) = 0.1 \ (\ddagger_2).$$

For  $i \notin \{1, 2, 3, 4\}$ ,  $b_2((i, j, k), \epsilon) = 1$ .

Above,  $\checkmark_i$  represents the probability of detection for stations with only motion sensors in Case  $i$ ;  $\dagger_i$  is to model the probabilities of misclassifications for stations with only motion sensors in the Case  $i$ ;  $\ddagger_i$  is for the probability of misdetection for stations with only motion sensors in the Case  $i$ ; finally,  $\checkmark\checkmark_1$ ,  $\dagger\dagger_1$ , and  $\ddagger\ddagger_1$  are used to model the probabilities of detection, misclassifications, and misdetection for the stations in Case 1 with both motion and material type sensors. It is obvious to see that the sensor configuration in Case 1 gives more information about the executed events than the configuration in Case 2.

## C. Performance Measure

Different methods may be proposed to evaluate the performance of SWDs. However, the performance measure described below seems to work best. Suppose the system runs for  $N$  event executions with  $n$  occurrences of a special event,  $f$ . Let executed events be sequentially indexed from one through  $N$ . Let  $i_k$  denote the index of the event, where  $f$  is executed the  $k$ th time. Let  $T(i)$  denote the true number of times  $f$  has been executed and  $\hat{T}(i)$  denote the value of the estimated count,  $c(m)$ , just after the execution of the  $i$ th event. The *average normalized error*,

$$ANE = \frac{1}{n} \sum_{k=1}^n \frac{|T(i_k) - \hat{T}(i_k)|}{T(i_k)}, \quad (34)$$

is used to compare the performances of the SWDs. In this expression, for each occurrence of  $f$ , the error of the count estimate is calculated and normalized by the true number of occurrences of  $f$ . This value is then averaged over all occurrences of  $f$ .

#### D. Simulation Results

Suppose we are interested in estimating the number of times the event  $(2, 1, 1)$  has been executed, i.e., the special event  $f$  is  $(2, 1, 1)$ . Simulations results for sensor configurations in Cases 1 and 2 are shown, respectively, in Figs. 8 and 9, where  $T(i)$  and  $\hat{T}(i)$  are plotted against the number of executed events. The simulations were terminated when  $f$  was executed for the 200th time. As shown in the figures, the

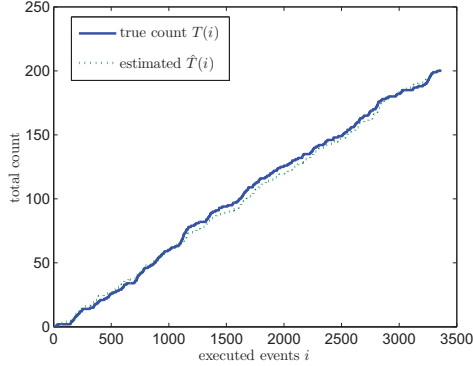


Fig. 8. SWD results for sensor configuration in Case 1 with  $\alpha = 0.29$

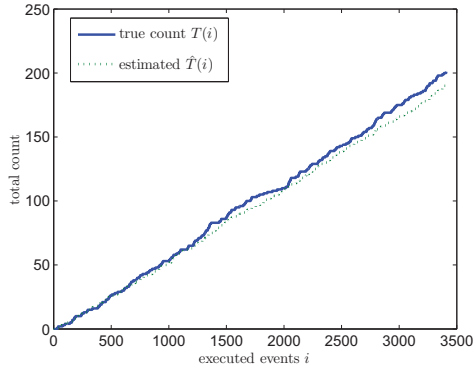


Fig. 9. SWD results for sensor configuration in Case 2 with  $\alpha = 0.36$

SWDs track the number of times that  $f$  occurred accurately. The values of  $\alpha$  for simulations generating the figures were chosen by comparing performances of SWDs for a range of values of  $\alpha$ . Tables I and II show the ANE averaged over 20 trails for each listed candidate value of  $\alpha$  for Cases 1 and 2, respectively. The ANE corresponding to values of  $\alpha$  not listed in Table I are larger than those corresponding to values of  $\alpha$  listed. The same is true for Table II. The values of  $\alpha$

TABLE I

ANE AVERAGED OVER 20 TRAILS FOR VARIOUS  $\alpha$  FOR CASE 1

$\alpha$	0.25	0.26	0.27	0.28	<b>0.29</b>	0.30
ANE	0.0869	0.0764	0.0701	0.0651	<b>0.0636</b>	0.0640
$\alpha$	0.31	0.32	0.33	0.34	0.35	
ANE	0.0659	0.0705	0.0786	0.0923	0.1024	

chosen for simulations generating Figs. 8 and 9 are 0.29 and

TABLE II

ANE AVERAGED OVER 20 TRAILS FOR VARIOUS  $\alpha$  FOR CASE 2

$\alpha$	0.30	0.31	0.32	0.33	0.34	0.35
ANE	0.1458	0.1277	0.1136	0.0981	0.0842	0.0774
$\alpha$	<b>0.36</b>	0.37	0.38	0.39	0.40	
ANE	<b>0.0751</b>	0.0769	0.0835	0.0946	0.1056	

0.36, which yield the least ANE (i.e., 0.0636 and 0.0751) in Tables I and II, respectively. Moreover, as expected, the SWD in Case 1, having more information about the executed events, is able to estimate the number of times  $f$  occurred with less ANE than the SWD in Case 2.

For comparison, Figs. 10 and 11 show the results of stochastic counters developed in [12] for Cases 1 and 2, respectively. The ANE averaged over 20 trails are 0.0490

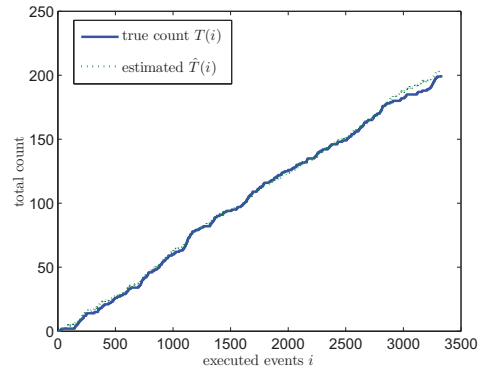


Fig. 10. Stochastic counter results for sensor configuration in Case 1

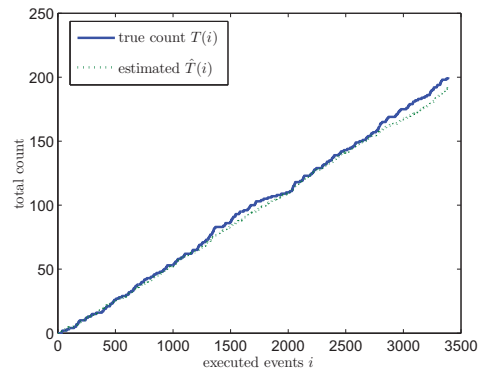


Fig. 11. Stochastic counter results for sensor configuration in Case 2

and 0.070 for Cases 1 and 2, respectively.

Although SWDs did not yield better performance than stochastic counters in this particular application, there is one potential advantage of SWDs. Note that the design of both SWDs and stochastic counters are based on knowledge of the system to be observed (the DEDS model) and its sensor configurations (the observation model). Suppose that model inaccuracies (e.g., malfunction of sensors) are discovered only after designed diagnosers are implemented in hardware. In this case, it is likely that both SWDs and stochastic



counters may exhibit degraded performance. Fortunately, it is possible for SWDs to tune the parameter  $\alpha$  to improve performance, while there is no way to correct the performance for stochastic counters without re-designing and re-implementing the hardware (which might be expensive). To capitalize on the additional parameter,  $\alpha$ , it is important to have a good understanding of relations among  $\alpha$ , the system model, the observation model, and the performance measure (e.g., ANE). This understanding may come from numerical experiments or closed form formulae derivations. This work is in progress and results will be reported in future correspondences.

## V. CONCLUSIONS

The SWD algorithm for counting occurrences of special events in partially observed DEDS was developed. Initially, both the probability that special events has occurred and the estimated count of the special event occurrences are set to zero. The algorithm sequentially updates the conditional probability that special events have occurred given available observations until this probability exceeds a threshold determined by the false alarm tolerance parameter. Then, the algorithm increases the estimated count by one, resets the probability that special events occurred to zero, and repeats the above process. The updates of the probability are achieved by simple vector-matrix multiplications using the state probability vectors and stochastic diagnoser probability transition matrices introduced in [9].

As an application, SWDs were applied to a material flow system, where occurrences of an abnormal routing event were counted. The SWDs worked well in tracking the number of times the abnormal routing event has been executed. To fully exploit the potential of SWDs, a work in progress is to explore and describe the relations among of the false alarm tolerance, system model, observation model, and the performance of SWDs.

We note that, in the SWD algorithm, resetting the probability that  $f$  has occurred to zero (via (31) and (32)) after an estimated occurrence of  $f$  is counted is based on heuristics. A better procedure for the reset operation is currently under development.

Finally, developing optimization algorithms for finding a least costly sensor configuration, while maximizing performance of the SWD (in terms of minimizing ANE) is also a work in progress.

## ACKNOWLEDGEMENT

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

## REFERENCES

- [1] E. Athanapoulou, L. Lingxi, and C.N. Hadjicostis. Probabilistic failure diagnosis in finite state machines under unreliable observations. In *Proc. of 2006 8th International Workshop on Discrete Event Systems*, pages 301 – 306, Ann Arbor, MI, July 2006.
- [2] C. G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.
- [3] O. Contant, S. Lafortune, and D. Teneketzis. Diagnosis of intermittent faults. *Discrete Event Dynamic Systems: Theory and Applications*, 14(2):171–202, 2004.
- [4] H. E. Garcia and T. Yoo. Model-based detection of routing events in discrete flow networks. *Automatica*, 4(41):583–594, 2005.
- [5] S. Jiang, R. Kumar, and H. E. Garcia. Diagnosis of repeated/intermittent failures in discrete-event systems. *IEEE Trans. Robotics and Applications*, 19(2):310–323, 2003.
- [6] J. Lunze and J. Schrder. State observation and diagnosis of discrete-event systems described by stochastic automata. *Discrete Event Dyna. Syst.: Theory Appl.*, 11(4):319–369, 2001.
- [7] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete event systems. *IEEE Trans. on Automat. Contr.*, 40(9):1555–1575, September 1995.
- [8] D. Thorsley and D. Teneketzis. Diagnosability of stochastic discrete-event systems. *IEEE Trans. Autom. Control*, 50(4):476–492, 2005.
- [9] D. Thorsley, T. Yoo, and H. E. Garcia. Diagnosability of stochastic discrete event systems under unreliable observations. In *Proceedings of the American Control Conference*, pages 1158 – 1165, 2008.
- [10] T. Yoo and H. E. Garcia. New results on discrete-event counting under reliable and unreliable observation information. In *Proc. of 2005 IEEE International Conference on Networking, Sensing and Control*, 2005.
- [11] T. Yoo and H. E. Garcia. Diagnosis of behaviors of interest in partially-observed discrete-event systems. *Systems & Control Letters*, 57(12):1023–1029, 2008.
- [12] T. Yoo and H. E. Garcia. Stochastic event counter for discrete-event systems under unreliable observations. In *Proceedings of the American Control Conference*, pages 1145 – 1152, 2008.
- [13] T. Yoo and H. E. Garcia. Event counting of partially-observed discrete-event systems with uniformly and nonuniformly bounded diagnosis delays. *Discrete Event Dynamic Systems*, 19(2):167–187, 2009.