

# Stochastic Event Counter for Discrete-Event Systems under Unreliable Observations

Tae-Sic Yoo and Humberto E. Garcia

**Abstract**—This paper addresses the issues of counting the occurrence of special events in the framework of partially-observed discrete-event dynamical systems (DEDS). First, we develop a noble recursive procedure that updates active counter information state sequentially with available observations. In general, the cardinality of active counter information state is unbounded, which makes the exact recursion infeasible computationally. To overcome this difficulty, we develop an approximated recursive procedure that regulates and bounds the size of active counter information state. Using the approximated active counting information state, we give an approximated minimum mean square error (MMSE) counter. The developed algorithms are then applied to count special routing events in a material flow system.

## I. INTRODUCTION

The failure/fault analysis of discrete-event dynamical systems has received attentions from academia and industries since the seminal work [8] was published. The framework presented in [8] 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.

Among the subsequent extensions and improvements of [8], recent developments on the detection problem of special events accounting for sensor unreliability and stochastic aspects in discrete-event systems include [6], [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. In [9], the authors 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 [8]. Also presented in [9] is the procedure of building the stochastic diagnoser<sup>1</sup> that bears a similar structure of the logical diagnoser of [8]. The transitions of the stochastic diagnoser include probability transition matrixes that can be used to update the probability distribution on the state estimate. 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]. 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], a counting strategy accommodating stochastic automata and unreliable observations was presented. However, the counting strategy of [10] 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 this paper, we attempt to fully utilize the probabilistic aspect of stochastic automata in developing algorithms for special event counting.

The main contributions of this paper are summarized below.

- A noble recursive procedure for updating counter information state is given;
- An approximated recursive procedure for updating counter information state is developed;
- An approximated MMSE conditional expectation counter is given;
- We apply the developed methodology to a material flow system where some special routing events are counted dynamically.

The rest of the paper is organized as follows. In Section II, we provide necessary notation and definitions. Section III defines MMSE counter and gives the associated counter information state. Also given in Section III is a recursive procedure for sequentially updating counter information state. In Section IV, we develop an approximated algorithm for the recursive procedure given in Section III. An approximated MMSE counter is described in Section IV as well. In Section V, we give an illustrative material flow system application where the developed counters provide the dynamic estimates on the number of occurrences of special routing events.

We assume in the remainder of this paper that the reader is familiar with terminologies typical of DEDS.

## II. PRELIMINARIES

In this section, we define the model of DEDS under consideration and related necessary notation. We consider

T.-S. Yoo and H.E. Garcia are with the Idaho National Laboratory, P.O. Box 1625, Idaho Falls, ID 83415-6180, USA. Email: {Tae-Sic.Yoo, Humberto.Garcia}@inl.gov

<sup>1</sup>The stochastic diagnoser is not a stochastic automaton. Therefore, the results presented in [6], [9] do not contradict each other.

a stochastic automaton as the system model. A stochastic automaton is a quadruple

$$A = (X, \Sigma, a, \pi_0)$$

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:

$$a : X \times \Sigma \times X \rightarrow [0, 1]$$

where,  $\forall x \in X$ ,

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

In practice, sensors are seldom perfect. Observation information from sensors thus can be unreliable. We model the unreliability of observation information in the following manner. Let  $\Delta := \{y_1, y_2, \dots, y_{n_y}\}$  be the set of distinctive observation symbols. The set of output symbols is

$$\Delta_* := \Delta \cup \{\epsilon\}$$

where the symbol  $\epsilon$  represents the situation of having no observation. The unreliable output function  $b : \Sigma \times \Delta_* \rightarrow [0, 1]$  satisfies the following:  $\forall \sigma \in \Sigma$ ,

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

The functional value  $b(\sigma, y)$  is the conditional probability of having output  $y \in \Sigma \times \Delta_*$  when the system executes event  $\sigma \in \Sigma$ .<sup>2</sup>

For any finite set  $S$ ,  $S^*$  denotes the Kleene closure of  $S$ . A run of the system is the sequence of transitions of system states such that

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

where  $x_s^i, x_d^i \in X$  and  $\sigma^i \in \Sigma$ , for  $i = 1, \dots, n$ . For brevity, let us have the following sequence notation:

$$\{(x_s^i, \sigma^i, x_d^i)\}_{i=1}^n := (x_s^1, \sigma^1, x_d^1) \dots (x_s^n, \sigma^n, x_d^n).$$

The above run of the system is called feasible if and only if the following conditions satisfy conjunctively:

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

An output run of the system is the sequence of output symbols such that

$$\{o^i\}_{i=1}^n := o^1 o^2 \dots o^n \in (\Delta_*)^*$$

where  $o^i \in \Delta_*$  for  $i = 1, \dots, n$ . The output run  $\{o^i\}_{i=1}^n$  is called feasible if and only if there is a feasible run of the system  $\{(x_s^i, \sigma^i, x_d^i)\}_{i=1}^n$  such that

$$b(\sigma^i, o^i) > 0, \forall i \in \{1, \dots, n\}.$$

<sup>2</sup>One can also model insertion or false alarm by including the self-loop with insertion events to the stochastic automata  $A$  and updating  $b$  appropriately to include the insertion events.

An observation run of the system is the sequence of observation symbols such that

$$\{y^i\}_{i=1}^m := y^1 y^2 \dots y^m \in \Delta^*$$

where  $y^i \in \Delta$  for  $i = 1, \dots, m$ . The observation run  $\{y^i\}_{i=1}^m$  is called feasible if and only if there is a feasible output run of the system  $\{o^i\}_{i=1}^n \in (\Delta_*)^*$  such that

$$P_\Delta(\{o^i\}_{i=1}^n) = \{y^i\}_{i=1}^m$$

where  $P_\Delta : (\Delta_*)^* \rightarrow \Delta^*$  is a plain projection function that removes  $\epsilon$  symbol from the output runs of the system.

Given the above stochastic model, we are interested in sequentially estimating the number of occurrences of event  $f \in \Sigma$  with observation sequences from  $\Delta$ .<sup>3</sup> We call these sequential estimation functions as *counters* in this paper. Formally, a counter is a function  $C : \Delta^* \rightarrow \mathbb{R}_*$  where  $\mathbb{R}_*$  is the set of non-negative real numbers. We denote  $N_t(f)$  a random variable on the number of transitions incurred by event  $f$  up to the moment when the  $t^{\text{th}}$  observation is available.

We note that the above stochastic model is similar to a Hidden Markov Model (HMM) [7] in that it is doubly-stochastic; i.e., the dynamics of the underlying system and the associated observations are both stochastic. A major distinction is that the above stochastic model explicitly accounts for “no observation” represented by the  $\epsilon$  symbol. In HMM, one usually assumes that stochastic observations are available and ready to be processed if and only if the underlying Markov model makes a transition. In this sense, the presented framework of stochastic automata and unreliable output functions subsumes HMM.

An example is given below illustrating the notation presented above.

*Example 1:* Consider the stochastic automaton  $A$  and the associated unreliable output function  $b$  described in Fig. 1.

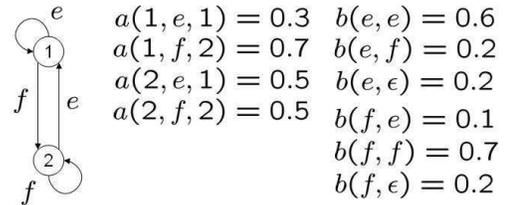


Fig. 1. Stochastic automaton and unreliable output function

Consider a sample run of the system and corresponding output and observation runs below.

System run	:	(1, e, 1)(1, f, 2)(2, f, 2)(2, e, 1)...
Output run	:	e e $\epsilon$ f ...
Observation run	:	e e f ...

Note that the third transition (2, f, 2) of the sample run of the system does not generate an observation symbol in

<sup>3</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 [8] is straightforward.

$\Delta$ . Counters will use the observation run  $ee f \dots$  to update functional values. Also note that system transitions incurred by  $f$  occurred twice while  $ee f$  is observed. Therefore,  $N_3(f) = 2$  for the above system run.

### III. INFORMATION STATE OF COUNTERS

We are interested in designing counters that minimize the mean square error. Formally, for any function  $g_t$  of the sequence of random observations from  $\Delta$  of length  $t$ , say  $\{Y_k\}_{k=1}^t := (Y_1, Y_2, \dots, Y_t)$ , we want to find a function that minimizes the mean square error as below:

$$\arg \min_{g_t} E[(N_t(f) - g_t)^2].$$

It is well-known that the conditional expectation of  $N_t(f)$  given  $Y_1^t$  is a minimizer for the above optimization argument. That is,

$$E[N_t(f) | \{Y_k\}_{k=1}^t] \in \arg \min_{g_t} E[(N_t(f) - g_t)^2].$$

The state space of counters considered in this paper is  $X \times \mathbb{Z}_*$  where  $\mathbb{Z}_*$  is the set of non-negative integers. Let  $\alpha_t(x_i, c)$  denote the probability that the system is at state  $x_i$  while executed the special event  $f$   $c$ -times given that realized observation sequence is  $\{y_{i_k}\}_{k=1}^t$  and the system state is initially distributed according to  $\pi_0$ . With this, we set the information state of counters as all possible special event counts of all system states:

$$Z_t := \{\alpha_t(x_i, c) : x_i \in X, c \in \mathbb{Z}_*\}$$

where

$$\sum_{i=1}^{n_s} \sum_{c=0}^{\infty} \alpha_t(x_i, c) = 1.$$

We call that  $\alpha_t(x_i, c)$  is *active* if  $\alpha_t(x_i, c) > 0$ . The set of active counts after observing  $\{y_{i_k}\}_{k=1}^t$  is

$$C_t := \{c : \alpha_t(x_i, c) > 0, x_i \in X, c \in \mathbb{Z}_*\}.$$

The initial information state of counters is

$$Z_0 = \{\alpha_0(x_i, c) : i = 1, \dots, n_s, c \geq 0\}$$

where

$$\alpha_0(x_i, c) = \begin{cases} \pi_0(x_i) & \text{if } c = 0; \\ 0 & \text{if } c > 0. \end{cases}$$

The above simply implies that the system is initially distributed following  $\pi_0$  while the special event  $f$  has never been executed.

We find a recursive form of updating information state  $Z_t$  from  $Z_{t-1}$ : for  $c_1 \geq 0$  and  $x_i \in X$ ,

$$\alpha_t(x_i, c) = \sum_{c'=0}^c \sum_{j=1}^{n_s} \alpha_{t-1}(x_j, c') \lambda(x_i, c - c' | x_j, y_{i_t}) \quad (1)$$

where  $\lambda(x_i, c_2 | x_j, y_{i_t})$  denotes the probability that the system reaches state  $x_i$  while executing event  $f$   $c_2$ -times along the sequence of transitions when we start in state  $x_j$  and observed  $y_{i_t}$ . Note that information state is updated only if new observation from  $\Delta$  is available. The above recursion implies that, when we receive a new observation,  $y_{i_t}$ , we

can compute  $\alpha_t(x_i, c)$  by adding all probabilities to reach state  $x_i$  from previous information state while executing the special event  $c - c'$  times given that we observed  $y_{i_t}$ . Then, the conditional expectation of special event count given the observation sequence is

$$E[N_t(f) | \{y_{i_k}\}_{k=1}^t] = \sum_{c=0}^{\infty} c \sum_{i=1}^{n_s} \alpha_t(x_i, c).$$

In the next subsection, we develop a procedure for computing  $\lambda(x_i, c | x_j, y_{i_t})$ .

#### A. Computation of $\lambda(x, c | x', y)$

For brevity, let us introduce the following notation. For  $o \in \Delta_*$ ,

$$p_{i,j}^{o,1} := a(x_i, f, x_j) b(f, o) \text{ and}$$

$$p_{i,j}^{o,0} := \sum_{\sigma \in \Sigma \setminus \{f\}} a(x_i, \sigma, x_j) b(\sigma, o).$$

Intuitively,  $p_{i,j}^{o,1}$  is the probability of reaching  $x_j \in X$  by executing  $f$  event with output  $o \in \Delta_*$  when we start in state  $x_i$ . On the other hand,  $p_{i,j}^{y,0}$  is the probability of reaching  $x_j \in X$  with output  $o \in \Delta_*$  while the system does not execute  $f$  when we start in state  $x_i$ .

We denote the probability of reaching state  $x_j$  with observation  $y \in \Delta$  while executing  $f$  events  $c$ -times along the transitions when we start in state  $x_i$  by  $\lambda(x_j, c, y | x_i)$ . For brevity, we will use a short handed notation  $\lambda_{i,j}^y(c)$  for  $\lambda(x_j, c, y | x_i)$ . Note that  $\lambda_{i,j}^y(c)$  differs from  $\lambda(x_j, c | x_i, y)$  in that observation  $y$  is not conditioned. The reach probability  $\lambda_{i,j}^y(c)$  satisfies the following set of recursive equations: For all  $1 \leq i, j \leq n_s$ ,  $y \in \Delta$ , and  $c \geq 0$ ,

$$\lambda_{i,j}^y(c) = \sum_{k=1}^{n_s} p_{i,k}^{\epsilon,0} \lambda_{k,j}^y(c) + \sum_{k=1}^{n_s} p_{i,k}^{\epsilon,1} \lambda_{k,j}^y(c-1) + p_{i,j}^{y,c} \quad (2)$$

In the above set of equations,  $p_{i,k}^{\epsilon,0} \lambda_{k,j}^y(c)$  implies the probability of the following scenario. First, the system moves from  $x_i$  to  $x_k$  (with 1 step) while avoiding the execution of the special event and generating no observable symbol. Then, from  $x_k$ , the system reaches  $x_j$  while generating observation  $y$  and executing the special event  $c$ -times. Similarly,  $p_{i,k}^{\epsilon,1} \lambda_{k,j}^y(c-1)$  implies the probability of the following scenario. First, the system moves from  $x_i$  to  $x_k$  (with 1 step) while executing the special event and generating no observable symbol. Then, from  $x_k$ , the system reaches  $x_j$  while generating observation  $y$  and executing the special event  $c-1$ -times. The quantity  $p_{i,j}^{y,c}$  is the probability that the system moves from  $x_i$  to  $x_j$  (with 1 step) while executing the special event  $c$ -times and generating observation  $y$ . Note that  $\lambda_{i,j}^y(-1) = 0$  because negative counting is not possible. Also note that  $p_{i,j}^{y,c} = 0$ , if  $c \geq 2$ , because the system only can execute at most one special event per system execution. As the above scenarios are all disjoint and include all possible paths to  $\lambda_{i,j}^y(c)$ , by adding up these probabilities, we can compute  $\lambda_{i,j}^y(c)$  as in (2).

Let  $I_n$  be the  $n \times n$  identity matrix. Rewriting (2) gives the following set of equations:

$$\begin{aligned}\Lambda_0^y &= (I_{n_s} - P_0^\epsilon)^{-1} P_0^y; \\ \Lambda_1^y &= (I_{n_s} - P_0^\epsilon)^{-1} (P_1^\epsilon \Lambda_0^y + P_1^y); \\ \Lambda_c^y &= [(I_{n_s} - P_0^\epsilon)^{-1} P_1^\epsilon]^c \Lambda_1^y, \quad \forall c \geq 2.\end{aligned}\quad (3)$$

where the elements at the  $i^{\text{th}}$  column and the  $j^{\text{th}}$  row are:

$$\Lambda_c^y(i, j) := \lambda_{i,j}^y(c), \quad P_0^y(i, j) := p_{i,j}^{y,0}, \quad \text{and} \quad P_1^y(i, j) := p_{i,j}^{y,1}.$$

We note that

$$\sum_{k=0}^{\infty} \lambda_{i,j}^y(k) = \lambda_{i,j}^y$$

where  $\lambda_{i,j}^y$  denotes the probability of reaching state  $x_j$  with one observation step while observing  $y \in \Delta$  when we start in state  $x_i$ . We can set up the following set of equations for  $\lambda_{i,j}^y$ . For all  $1 \leq i, j \leq n_s, y \in \Delta$ ,

$$\lambda_{i,j}^y = \sum_{k=1}^{n_s} p_{i,k}^\epsilon \lambda_{k,j}^y + p_{i,j}^y,$$

where  $p_{i,j}^o$  is the probability of reaching  $x_j \in X$  with output  $o \in \Delta_*$  when we start in state  $x_i$ . Formally, for  $o \in \Delta_*$ ,

$$p_{i,j}^o := \sum_{\sigma \in \Sigma} a(x_i, \sigma, x_j) b(\sigma, o).$$

Solving the above set of equations gives

$$\Lambda^y = (I_{n_s} - P^\epsilon)^{-1} P^y$$

where the elements at the  $i^{\text{th}}$  column and the  $j^{\text{th}}$  row are:

$$\Lambda^y(i, j) := \lambda_{i,j}^y, \quad P^\epsilon(i, j) := p_{i,j}^\epsilon, \quad \text{and} \quad P^y(i, j) := p_{i,j}^y.$$

Note that  $P^0 = P_0^o + P_1^o$ , for  $o \in \Delta_*$ .

With marginalization, we have that, for  $x_i, x_j \in X, y \in \Delta$ , and  $c \in \mathbb{Z}_*$ ,

$$\lambda(x_j, c | x_i, y) = \frac{\lambda_{i,j}^y(c)}{\sum_{j=1}^{n_s} \lambda_{i,j}^y}. \quad (4)$$

We provide next an example illustrating the computation procedures presented above.

*Example 2:* Consider the stochastic automaton described in Fig. 1. Then, we have

$$\begin{aligned}P_0^\epsilon &= \begin{pmatrix} 0.06 & 0 \\ 0.1 & 0 \end{pmatrix}, \quad P_1^\epsilon = \begin{pmatrix} 0 & 0.14 \\ 0 & 0.1 \end{pmatrix}, \\ P_0^e &= \begin{pmatrix} 0.18 & 0 \\ 0.3 & 0 \end{pmatrix}, \quad P_1^e = \begin{pmatrix} 0 & 0.07 \\ 0 & 0.05 \end{pmatrix}, \\ P_0^f &= \begin{pmatrix} 0.06 & 0 \\ 0.1 & 0 \end{pmatrix}, \quad \text{and} \quad P_1^f = \begin{pmatrix} 0 & 0.49 \\ 0 & 0.35 \end{pmatrix}.\end{aligned}$$

With (3), we have that, for  $e \in \Delta$ ,

$$\begin{aligned}\Lambda_0^e &= (I_2 - P_0^\epsilon)^{-1} P_0^e = \begin{pmatrix} 0.1915 & 0 \\ 0.3191 & 0 \end{pmatrix}; \\ \Lambda_1^e &= (I_2 - P_0^\epsilon)^{-1} (P_1^\epsilon \Lambda_0^e + P_1^e) = \begin{pmatrix} 0.0475 & 0.0745 \\ 0.0367 & 0.0574 \end{pmatrix}; \\ \Lambda_{c+1}^e &= [(I_2 - P_0^\epsilon)^{-1} P_1^\epsilon]^c \Lambda_1^e = \begin{pmatrix} 0 & 0.1489 \\ 0 & 0.1149 \end{pmatrix}^c \Lambda_1^e.\end{aligned}$$

For  $f \in \Delta$ , we have that

$$\begin{aligned}\Lambda_0^f &= (I_2 - P_0^\epsilon)^{-1} P_0^f = \begin{pmatrix} 0.0638 & 0 \\ 0.1064 & 0 \end{pmatrix}; \\ \Lambda_1^f &= (I_2 - P_0^\epsilon)^{-1} (P_1^\epsilon \Lambda_0^f + P_1^f) = \begin{pmatrix} 0.0158 & 0.5213 \\ 0.0122 & 0.4021 \end{pmatrix}; \\ \Lambda_{c+1}^f &= [(I_2 - P_0^\epsilon)^{-1} P_1^\epsilon]^c \Lambda_1^f = \begin{pmatrix} 0 & 0.1489 \\ 0 & 0.1149 \end{pmatrix}^c \Lambda_1^f.\end{aligned}$$

We have that

$$P^\epsilon = P_0^\epsilon + P_1^\epsilon, \quad P^e = P_0^e + P_1^e, \quad \text{and} \quad P^f = P_0^f + P_1^f.$$

Then, we can compute the following.

$$\begin{aligned}\Lambda^e &= (I_2 - P^\epsilon)^{-1} \cdot P^e = \begin{pmatrix} 0.2452 & 0.0841 \\ 0.3606 & 0.0649 \end{pmatrix}; \\ \Lambda^f &= (I_2 - P^\epsilon)^{-1} \cdot P^f = \begin{pmatrix} 0.0817 & 0.5889 \\ 0.1202 & 0.4543 \end{pmatrix}.\end{aligned}$$

With (4), we can compute  $\lambda(x_j, c | x_i, y)$ , for  $x_j, x_i \in \{1, 2\}$ ,  $y \in \{e, f\}$ , and  $c \in \mathbb{Z}_*$ .

#### IV. APPROXIMATING COUNTER INFORMATION STATE

Note that Example 2 of the previous section gives that  $\lambda(x_j, c | x_i, e) > 0$  for all  $x_j, x_i \in \{1, 2\}$ ,  $y \in \{e, f\}$ , and  $c \in \mathbb{Z}_*$ . Then, for any initial distribution  $\pi_0$ , we have that  $\alpha_t(x, c) > 0$  for all  $x \in X$  and  $c \in \mathbb{Z}_*$  and  $t \in \mathbb{Z}$ . Thus, exact recursion in (1) incurs infinite active counter state, which makes exact recursion computationally infeasible. To overcome this difficulty, we give an approximation scheme that regulates and bounds the set of active counter information states. First, we begin with the following formulation that finds the minimum count explaining counter information state approximately within the given factor  $0 < \delta_1 \approx 0$  for each observation symbol  $y \in \Delta$ :

$$N_{i,j}^y(\delta_1) = \min \left( N : 1 - \frac{\sum_{c=0}^N \lambda_{i,j}^y(c)}{\lambda_{i,j}^y} < \delta_1 \right).$$

Note that for  $N > N_{i,j}^y(\delta_1)$ ,

$$\frac{\sum_{c=N}^{\infty} \lambda_{i,j}^y(c)}{\lambda_{i,j}^y} < \delta_1.$$

Intuitively, the above inequality means that the probability of getting count beyond  $N_{i,j}^y(\delta_1)$  for transition to state  $x_j \in X$  with observation  $y \in \Delta$  when the system starts at state  $x_i \in X$  is negligible. By finding the maximum among all possible system state pairs, we get the approximated maximum count for observation  $y \in \Delta$  as below:

$$N_{\delta_1}^y = \max (N_{i,j}^y(\delta_1) : 1 \leq i, j \leq n_s).$$

Thus, each time  $y \in \Delta$  is observed, we use  $N_{\delta_1}^y$  in the following manner to approximate  $\lambda(x_j, h | x_i, y)$ :

$$\tilde{\lambda}(x_j, h | x_i, y) = \begin{cases} \frac{\lambda_{i,j}^y(h)}{\sum_{j=1}^{n_s} \sum_{c=0}^{N_{\delta_1}^y} \lambda_{i,j}^y(c)} & \text{if } 1 \leq h \leq N_{\delta_1}^y; \\ 0 & \text{o.w.} \end{cases}$$

With the above scheme, we approximate the recursion (1) in the following manner. First, we have the initial approximated counter information state

$$\tilde{Z}_0 = \{\tilde{\alpha}_0(x_i, c) : x_i \in X, c \in \mathbb{Z}_*\}$$

where

$$\tilde{\alpha}_0(x_i, c) = \begin{cases} \pi_0(x_i) & i = 1, \dots, n_s \text{ and } c = 0; \\ 0 & \text{o.w.} \end{cases}$$

Approximated recursion is defined by replacing  $\lambda$  with  $\tilde{\lambda}$  as below.

$$\tilde{\alpha}_t(x_i, c) = \sum_{c'=0}^{\min(c, N_{\delta_1}^{y_{it}})} \sum_{j=1}^{n_s} \tilde{\alpha}_{t-1}(x_j, c-c') \cdot \tilde{\lambda}(x_i, c' | x_j, y_{it}).$$

Note that, the set of active information states is still not bounded uniformly over  $t$  in general (e.g., Example 2). This makes the recursion (1) with approximated values computationally infeasible again. Therefore, we need another level of approximation scheme to make the recursion (1) computable.

Let us denote the set of active counts at time  $t$  with the above approximation by

$$\tilde{C}_t := \{c : \tilde{\alpha}_t(x_i, c) > 0, x_i \in X, c \in \mathbb{Z}_*\}.$$

Also denote the probability of count by

$$\tilde{\alpha}_t(c) := \sum_{i=1}^{n_s} \tilde{\alpha}_t(x_i, c).$$

We index the elements of  $\tilde{C}_t$  with the following approach. For  $c_i, c_j \in \tilde{C}_t$ ,  $i < j$  if and only if

$$[\tilde{\alpha}_t(c_i) > \tilde{\alpha}_t(c_j)] \vee [(\tilde{\alpha}_t(c_i) = \tilde{\alpha}_t(c_j)) \wedge (c_i < c_j)].$$

Let us denote this complete-ordered set and the  $i^{\text{th}}$  element of this set by  $\tilde{S}_t$  and  $c_i$ , respectively. We find the minimum count that explains counter information state approximately within the given factor  $0 < \delta_2 \approx 0$ :

$$N_{\delta_2}^t := \min \left( N : 1 - \frac{\sum_{i=1}^N \tilde{\alpha}_t(c_i)}{\sum_{c \in \tilde{C}_t} \tilde{\alpha}_t(c)} < \delta_2 \right).$$

To give a hard bound on this set, for a given  $N^* > 0$ , let

$$\overline{N}_{\delta_2}^t := \min(N^*, N_{\delta_2}^t).$$

Note that when  $\overline{N}_{\delta_2}^t = N_{\delta_2}^t$ , for  $\overline{N}_{\delta_2}^t \leq N \leq |\tilde{C}_t|$ ,

$$\frac{\sum_{i=N}^{|\tilde{C}_t|} \tilde{\alpha}_t(c_i)}{\sum_{c \in \tilde{C}_t} \tilde{\alpha}_t(c)} < \delta_2,$$

which intuitively means that the probability of having active count  $c_i$  where  $i > \overline{N}_{\delta_2}^t$  is negligible. We collect the set of counts that explains the distribution of counts approximately in the following manner: For  $x \in X$  and  $c_i \in \tilde{C}_t$ ,

$$\alpha_t^{\delta_2}(x, c_i) := \begin{cases} \tilde{\alpha}_t(x, c_i) & \text{if } i \leq \overline{N}_{\delta_2}^t \\ 0 & \text{o.w.} \end{cases}$$

With normalization, we get

$$\hat{\alpha}_t^{\delta_2}(x, c) := \frac{\alpha_t^{\delta_2}(x, c)}{\sum_{i=1}^{\overline{N}_{\delta_2}^t} \sum_{j=1}^{n_s} \alpha_t^{\delta_2}(x_j, c_i)}.$$

Equipped with the above procedure, we give the approximated recursion below.

$$\tilde{\alpha}_t(x_i, c) = \sum_{c'=0}^{\min(c, N_{\delta}^{y_{it}})} \sum_{j=1}^{n_s} \hat{\alpha}_{t-1}^{\delta_2}(x_j, c-c') \cdot \tilde{\lambda}(x_i, c' | x_j, y_{it}).$$

Note that the approximated set of active counts at time  $t$  is

$$\hat{C}_t = \{c : \hat{\alpha}_t(x_i, c) > 0, x_i \in X, c \in \mathbb{Z}_*\}$$

and  $|\hat{C}_t| \leq N^*$  for all  $t \geq 0$ , trivially.

Having an approximated active counter information state developed above, we can give an approximated MMSE counter as below:

$$E(N_t(f) | \{y_k\}_{k=1}^t) \approx \sum_{i=1}^{\overline{N}_{\delta_2}^t} c_i \sum_{j=1}^{n_s} \hat{\alpha}_t^{\delta_2}(x_j, c_i).$$

## V. APPLICATION

Consider a material flow system depicted in Fig. 2. Figure

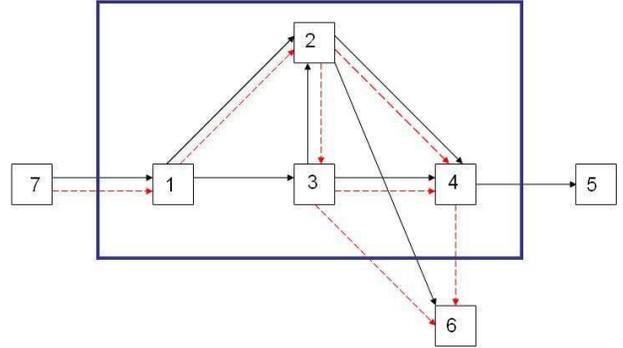


Fig. 2. Material Flow System

2 depicts a material handling 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 are buffered to be pushed in (to Station 1) for processing. 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.

### A. Stochastic Automata System Model

A set of automata is used to model the material flows among stations. For example, the automaton in Fig. 3 captures the material flow of 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 zero buffering capacity for all internal material processing stations.<sup>4</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  $(3, 1, 1)$ . Output stations 5 and 6 find the automata model  $O_5$  and  $O_6$  depicted in Fig. 4 and Fig. 5, 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. 6.

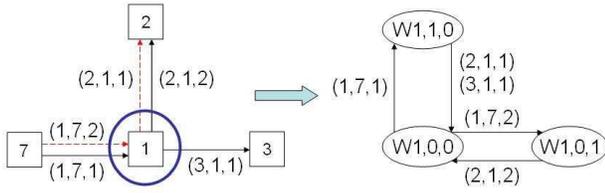


Fig. 3. Working Station Automaton Model:  $W_1$

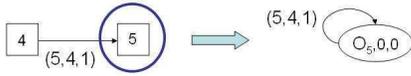


Fig. 4. Output Station Automaton Model:  $O_5$

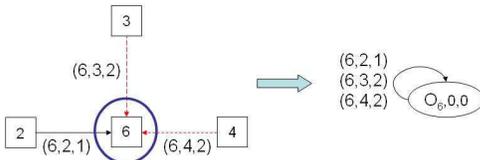


Fig. 5. Output Station Automaton Model:  $O_6$

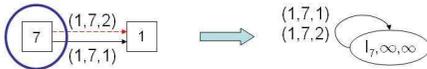


Fig. 6. Input Station Automaton Model:  $I_7$

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

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

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

<sup>4</sup>Modeling buffering capabilities may amount to introduce more states depending on the given material processing policies (FIFO, LIFO, etc.) and define appropriate transitions.

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  $act(x) = \{\sigma \in \Sigma : \exists x' \in X \text{ s.t. } a(x, \sigma, x') > 0\}$ .

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

Suppose that motion sensors are installed at all internal working stations,  $W_1, W_2, W_3$ , and  $W_4$ . Motion sensors are only able to identify the first component of the executed event (i.e., that a material has been transferred to the sensor's station). Unidentified attributes of events are marked with \*. We consider the following two cases when modeling different qualities for motion sensors.

(Case 1) For  $i \in W := \{1, 2, 3, 4\}$ ,

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

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

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

(Case 2) For  $i \in W$ ,

$$b_2((i, j, k), (i', *, *)) = \begin{cases} 0.6; & \text{if } i' = i; (\checkmark_2) \\ 0.1, & \text{if } i' \in W \setminus \{i\}; (\dagger_2) \end{cases}$$

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

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

Above, the symbol \* is to indicate unidentified attributes of events;  $\checkmark_i$  represents the probability of detection for the Case  $i$ ;  $\dagger_i$  is to model the probabilities of misclassifications for the Case  $i$ ;  $\ddagger_i$  is for the probability of misdetection for the Case  $i$ . It is obvious to see that the motion sensors of Case 1 are more accurate than the ones of Case 2.

### C. Simulation and computational results

Suppose that we are interested in counting the occurrence of event  $(2, 1, 1)$ . The following approximation parameters are used:

$$\delta_1 = 0.001, \quad \delta_2 = 0.001, \quad \text{and } N_{\delta_2}^t = \infty.$$

With these parameters, we ran the simulations with the identical system run of length 3000 for the both cases.

For the sake of readability, Figs. 7 and 8 are marked with every  $100^{th}$  computational results. The points connected with the thick line is the approximated maximum active counts,  $\max(\tilde{C}_t)$ . The dotted line is to connect the approximated minimum active counts,  $\min(\tilde{C}_t)$  for every  $100^{th}$  computations. We used circles to mark the actual number of occurrences of event  $(2, 1, 1)$ . The approximated MMSE counts are marked with  $+$ . We note that the cone of uncertainty becomes larger as the length of system run increases for the both cases. However, one can observe that Case 2 shows the wider band of uncertainty compared to Case 1, in general. This is expected because Case 1 is equipped with the sensors of higher accuracy than those of Case 2.

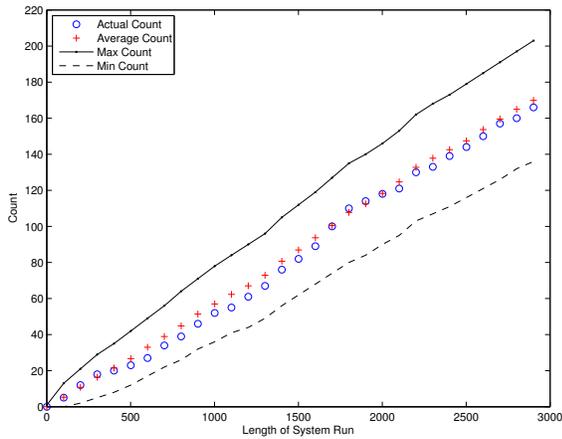


Fig. 7. Computational Results for Case 1

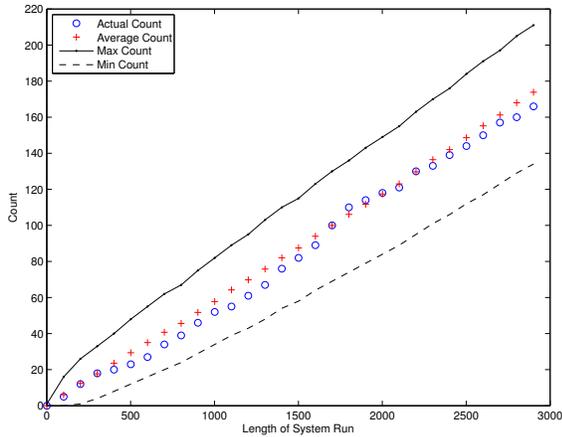


Fig. 8. Computational Results for Case 2

Figure 9 shows the progress of overcount/undercount of the approximated MMSE counts for the both cases. Overall, the approximated MMSE counter of the Case 1 shows less deviations from zero than that of Case 2.

Markers (o for Case 1 and x for Case 2) in Fig. 10

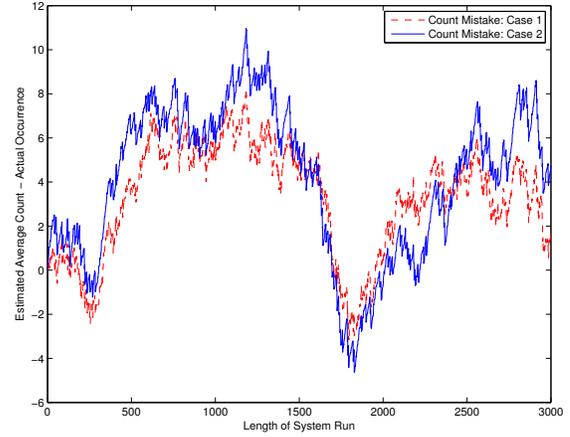


Fig. 9. Count mistakes

show the probability distributions of count for the both cases after the system executes the  $3000^{th}$  event. Solid lines are normal distribution counter parts with mean and variance are computed from the probability distributions of count. Both cases show near-perfect fit. Observing this, we conjecture the

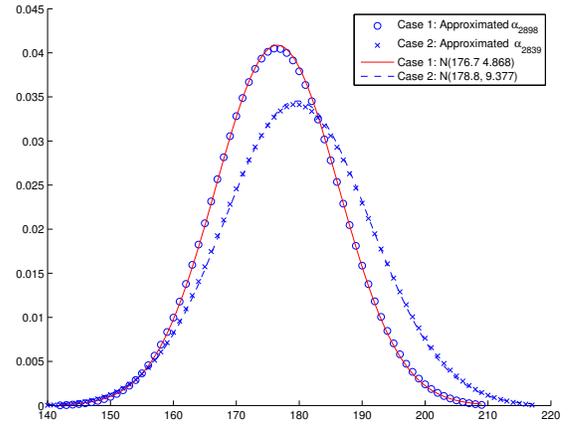


Fig. 10. Convergence to normal distribution

following.

(Conjecture) Let us denote random variable  $S_t$  where  $P(S_t = p) = \tilde{\alpha}_t(p)$  for  $t \in \mathbb{Z}$  and  $p \in \mathbb{Z}_*$ . If the stochastic automaton  $A$  is strongly connected, there exists a constant  $var \in \mathbb{R}$  such that

$$\frac{S_t - E(S_t)}{\sqrt{t}} \rightsquigarrow N(0, var) \quad (\text{in distribution})$$

where  $var \in \mathbb{R}$  is the variance of the normal distribution.

We note that  $var$  is a function of locations and qualities of sensors; higher qualities and better locations would entail a lower  $var$  value. This raises the issue of sensor selection, which is under current investigation.

## VI. FINAL REMARKS

This paper reports a methodology of stochastic counting of special events under stochastic automata with unreliable observation information. To facilitate computational feasibility, we developed an approximated stochastic counting scheme. Also notable is the conjecture on the convergence of count distribution. If this conjecture is positively verified, one can use the converged variance as a measure for the quality of the given observational network. Also interesting would be a synthesis problem such as selecting sensors to minimize the converged variance.

## ACKNOWLEDGEMENT

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

## REFERENCES

- [1] E. Athanassopoulou, 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. Laforune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.
- [3] O. Contant, S. Laforune, 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] L. R. Rabiner. A tutorial on hidden Markov models and selected applications inspeech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [8] M. Sampath, R. Sengupta, S. Laforune, K. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete event systems. *IEEE Trans. on Automat. Contr.*, 40(9):1555–1575, September 1995.
- [9] D. Thorsley and D. Teneketzis. Diagnosability of stochastic discrete-event systems. *IEEE Trans. Autom. Control*, 50(4):476– 492, 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 International Conference on Networking, Sensing and Control*, 2005.