Online Prognosis of Stochastic Discrete Event Systems

Feng-Jiao Lian, Shao-Long Shu
Department of Control Science and Engineering, Tongji University, Shanghai, China
E-mail: wslfjwps@163.com, shushaolong@tongji.edu.cn

Abstract-In this paper, we investigate how to calculate the probability of fault occurrence in the future based on the observations online. In a stochastic discrete event system framework, we first calculate the probability of fault occurrence for pre-fault event sequences which do not include fault events. Our idea is to attach label to each state in the stochastic automaton and convert it into a Markov chain. Based on the Markov chain, we determine the probability of fault occurrence by calculating the sum of probabilities of some states in the Markov Chain. Second, for every pre-fault observable event sequence, we find all the possible pre-fault event sequences and the probability of their occurrences. Our idea is to update the current state estimate for the observable event sequence and then calculate the probability of every state based on the current state estimate which can be done with polynomial complexity. Combining these two results, the online prognosis problem is successfully solved.

Keywords-stochastic automaton; online prognosis; prognostic probability; state estimation; Markov chain

I. INTRODUCTION

In discrete event systems [1]-[3], fault prognosis has been extensively investigated as prognosability[4]-[8]. A discrete event system is said to be prognosable (or say predictive) if faults can be undoubtedly determined to occur in the future based on the observations of event sequences. Generally speaking, prognosability is rather strict, even more stronger than diagnosability, which requires faults can be diagnosed in finite observations after they occur [9].

In this paper, we do not require that the occurrence of faults can always be predicted with certainty, but to calculate the prognosable degree for all the observations. In order to do it, we adopt stochastic automata to model discrete event systems and an online prognoser is used to calculate the probability of fault occurrence in the future. Whenever the system has an output (an observable event), the prognoser will immediately update the probability of fault occurrence. It is an online prognosis problem of stochastic discrete event systems.

We solve the online prognosis problem as follows. We first calculate the probability of fault occurrence for pre-fault event sequences which do not include the fault events. Our idea is to attach label to each state in the stochastic automaton and convert it into a Markov chain. Based on the Markov chain, we determine the probability of fault occurrence by calculating the sum of probabilities of some states in the Markov Chain which has been successfully. We then, for every output (an observable event) of the system, find all the possible pre-fault event sequences and the probability of their occurrences. Our idea is to update the current state estimate for the current output and then calculate the probability of every state in the current state estimate which can be done with polynomial complexity. Combining these two results, the online prognosis problem is successfully solved.

Recently, some work has been done in a stochastic discrete event system framework. The papers [10]-[12] discuss fault diagnosis problem of stochastic discrete event systems. The differences are as follows: 1) they discuss fault diagnosis while our paper discusses fault prognosis. 2) They consider qualitative performance evaluation of stochastic discrete event systems while our paper considers quantitative performance evaluation of stochastic discrete event systems. In our previous work [13], detectability measure was investigated which calculates the sum of probabilities of all the detectable event sequences. Even though what it investigates is a quantitative performance evaluation problem, the differences are obviously as follows: 1) our previous work [13] discusses state estimation while this paper considers fault prognosis. 2) Calculating detectability measure requires that every detectable event sequence can surely determine the current state while in this paper, we calculate the predictable probability of faults for every event sequence. 3) The computational complexity of calculating detectability measure is exponential while the computational complexity of calculating prognosable probability is polynomial. The most related work is done by [14], where fault prognosis of stochastic discrete event systems is investigated. However it still investigates prognosability which is a qualitative performance evaluation indicator. Furthermore, it does not discuss how to calculate the prognosable probability online which is an open problem [14].

The rest of the paper is organized as follows. Section II introduces the stochastic automaton model and some necessary notions. Section III proposes an algorithm to calculate the prognostic probability of pre-fault event sequences. Section IV presents a method to compute the prognostic probability of a pre-fault observable events sequence. Combine these results; a procedure is proposed to solve the problem of online prognosis in Section V. Finally, we conclude the paper in section VI.
II. STOCHASTIC DISCRETE EVENT SYSTEMS

A stochastic discrete event system is modeled by a stochastic automaton
\[ \tilde{G} = (Q, \Sigma, \delta, p_{\tilde{G}}, q_0), \]
where \( Q \) is the finite state set; \( \Sigma \) is the finite event set; \( \delta: Q \times \Sigma \to Q \) is the state transition function which defines the dynamics of the stochastic automaton. The transition function \( \delta \) is extended to \( \delta: Q \times \Sigma \to Q \) in the usual way. The initial state is \( q_0 \). We use \( \Gamma(q) \) to denote all the events which can occur at state \( q \). For string \( s \in \sum^* \), we use \( |s| \) to denote its length. For a set like \( Q \), we use \( |Q| \) to denote its cardinality. We use \( N \) to denote the set of natural numbers. All the possible faults are modeled as a fault event \( f \) because we only want to predict the occurrence of faults.

With a slight abuse of notations, we use \((q_i, \sigma, q_j)\) to denote a transition which means that event \( \sigma \) drives the system from state \( q_i \) to state \( q_j \). The set of all the transitions are denoted as \( \mathcal{T} = \{(q_i, \sigma, q_j) \in Q \times \Sigma : \delta(q_i, \sigma) = q_j\} \). \( p_\mathcal{T} \) is the probability transition function of \( \tilde{G} \) defined as \( p_\mathcal{T}: \mathcal{T} \to (0,1) \). For a given transition \((q_i, \sigma, q_j)\), \( p_\mathcal{T}(q_i, \sigma, q_j) \) is the probability that event \( \sigma \) occurs and the system transitions to state \( q_j \) given that the system is in state \( q_i \). Note that if a transition is defined, then it will occur with a positive probability. For any undefined transition \((q_i, \sigma, q_j) \notin \mathcal{T} \), we define \( p_\mathcal{T}(q_i, \sigma, q_j) = 0 \) which means the transition never occurs.

The language generated by the stochastic automaton is defined as
\[ L(\tilde{G}) = \{s \in \sum^* : \delta(q_0, s)\}, \]
where \( \delta(q_0, s) \) means that \( \delta(q_0, s) \) is defined. If we remove probabilities from \( \tilde{G} \), the stochastic automaton is then converted into a non-stochastic automaton, denoted as \( G = (Q, \Sigma, \delta, q_0) \).

The non-stochastic automaton \( G \) has the same dynamics as the stochastic automaton \( \tilde{G} \) such that \( L(G) = L(\tilde{G}) \).

Generally speaking, some events are observable and the others are unobservable. The set of observable events is denoted by \( \Sigma_o \subseteq \Sigma \). \( \Sigma_{uo} = \Sigma - \Sigma_o \) is the set of unobservable events. Without loss of generality, we assume that the fault event \( f \) is unobservable, that is, \( f \in \Sigma_{uo} \). For any string \( s \in L(\tilde{G}) \), the observation is the natural projection \( M_{\tilde{G}}: \sum^* \to \sum_o \) defined recursively as
\[ M_{\tilde{G}}(\mathcal{E}) = \mathcal{E}; M_{\tilde{G}}(ss\sigma) = \begin{cases} M_{\tilde{G}}(s)s\sigma & \text{if } \sigma \in \sum_o, \\ M_{\tilde{G}}(s) & \text{if } \sigma \notin \sum_o. \end{cases} \]
where \( \mathcal{E} \) denotes the empty string. When considering the prediction problem of faults, we assume that the time interval between adjacent events in an event sequence is long enough to take action such as giving an alarm. Hence for an observable event \( t \), the inverse projection \( M_{\tilde{G}}^{-1}(t) \) of an observable string \( t \) is defined as:
\[ M_{\tilde{G}}^{-1}(\mathcal{E}) = \mathcal{E}; \]
\[ M_{\tilde{G}}^{-1}(t) = \{s_{ot} \in L(\tilde{G}) : M_{\tilde{G}}(s_{ot}) = t\}. \]
where \( s_{ot} \) is the string of which the last event is observable, that is, the last event is right the last event of observable string \( t \). We extend the projection and inverse projection to a language \( K \subseteq L(\tilde{G}) \) in the usual way as \( M_{\tilde{G}}(K) = \{M_{\tilde{G}}(s) : s \in K\} \) and \( M_{\tilde{G}}^{-1}(K) = \bigcup_{s \in K} M_{\tilde{G}}^{-1}(s) \).

Given a state \( q_i \in Q \), the probability of the occurrence of string \( s \) from it is denoted as
\[ \Pr_{\tilde{G}}(q_i, s) = 1; \]
\[ \Pr_{\tilde{G}}(q_i, \sigma) = p_{\tilde{G}}(q_i, \sigma, \delta(q_i, \sigma)); \]
\[ \Pr_{\tilde{G}}(q_i, s\sigma) = \Pr_{\tilde{G}}(q_i, s) \times p_{\tilde{G}}(\delta(q_i, s), \sigma, \delta(q_i, s)\sigma). \]
Note that the initial state \( q_0 \) is known. We assign a probability to each string in \( L(\tilde{G}) \) as follows:
\[ \Pr_{\tilde{G}}(\mathcal{E}) = \Pr_{\tilde{G}}(q_0, \mathcal{E}) = 1; \]
\[ \Pr_{\tilde{G}}(\sigma) = \Pr_{\tilde{G}}(q_0, \sigma, \delta(q_0, \sigma)); \]
\[ \Pr_{\tilde{G}}(s\sigma) = \Pr_{\tilde{G}}(q_0, s) \times p_{\tilde{G}}(\delta(q_0, s), \sigma, \delta(q_0, s)\sigma). \]
For any string \( s \notin L(\tilde{G}) \), we have \( \Pr_{\tilde{G}}(s) = 0 \).

In order to investigate the online prognosis problem, the following assumptions [13] for the stochastic automaton \( \tilde{G} \) are needed.

Assumption 1: The language \( L(\tilde{G}) \) generated by \( \tilde{G} \) is live. That is, for every state \( q_i \in Q \), at least one transition is defined and the sum of probabilities of all the transitions defined at this state equals to one or, equivalently,
\[ (\forall q_i \in Q) \sum_{\sigma\in\delta(q_i)} p_{\tilde{G}}(q_i, \sigma, \delta(q_i, \sigma)) = 1. \]

Assumption 2: The automaton \( \tilde{G} \) does not have any cycle of unobservable events or, equivalently,
\[ (\forall s \in L(\tilde{G})) (\exists n \in N) |s| \leq n. \]

III. PROGNOSTIC PROBABILITY OF PRE-FAULT EVENT SEQUENCES

Before considering the online prognosis problem, we first want to know the probability with which the faults will occur for a given pre-fault event sequence \( s \in L(\tilde{G}) \).

We say an event sequence \( s \) is pre-fault if the event sequence does not include event \( f \), denoted as \( f \notin s \). Else, \( f \in s \) and we say event sequence \( s \) is post-fault. We define the set of all the pre-fault event sequences \( L_{pre}(\tilde{G}) \) as
$L_{\text{pre}}(\tilde{G}) = \{ s \in L(\tilde{G}) : f \not\in s \}$,

$L_{\text{ms}}(\tilde{G})$ is prefix-closed, we define the set of all the post-fault event sequences $L_{\text{post}}(\tilde{G})$ as

$L_{\text{post}}(\tilde{G}) = \{ s \in L(\tilde{G}) : f \in s \}$.

Easily to know $L(\tilde{G}) = L_{\text{pre}}(\tilde{G}) \cup L_{\text{post}}(\tilde{G})$.

We continue to introduce some more notations. For event sequence $s \in L(\tilde{G})$, the postlanguage $L_i(\tilde{G})$ is the set of possible continuations of event sequence $s$, that is,

$L_i(\tilde{G}) = \{ s' \in \Sigma^* : ss' \in L(\tilde{G}) \}$.

Every event sequence in $L_i(\tilde{G})$ is pre-fault or post-fault. We then divide $L_i(\tilde{G})$ into two parts. One is the set of all the pre-fault event sequences, denoted as

$L_{i,\text{pre}}(\tilde{G}) = \{ s' \in L_i(\tilde{G}) : f \not\in s' \}$.

The other is the set of all the post-fault event sequences, denoted as

$L_{i,\text{post}}(\tilde{G}) = \{ s' \in L_i(\tilde{G}) : f \in s' \}$.

Based on the length of strings, we divide the language $L_i(\tilde{G})$ into a serial of sublanguage $L_{i,0}(\tilde{G}), L_{i,1}(\tilde{G}), \ldots, L_{i,j}(\tilde{G}), \ldots, L_{i,j}(\tilde{G})$ consists of all the strings with the length $j$ in $L_i(\tilde{G})$, that is,

$L_{i,j}(\tilde{G}) = \{ s \in L_i(\tilde{G}) : |s| = j \}$.

Note that $L_{i,0}(\tilde{G}) = \{ e \}$. we further divide every sublanguage $L_{i,j}(\tilde{G})$ into two parts, the pre-fault part $L_{i,j,\text{pre}}(\tilde{G})$ defined as

$L_{i,j,\text{pre}}(\tilde{G}) = \{ s' \in L_{i,j}(\tilde{G}) : f \not\in s' \}$

and the post-fault part $L_{i,j,\text{post}}(\tilde{G})$ defined as

$L_{i,j,\text{post}}(\tilde{G}) = \{ s' \in L_{i,j}(\tilde{G}) : f \in s' \}$.

Here we want to know the probability with which the faults will occur for a given pre-fault event sequence $s \in L_{\text{pre}}(\tilde{G})$, that is, the sum of probabilities of all the sufficiently long post-fault event sequences in $L_{i,\text{post}}(\tilde{G})$. Since the given stochastic automaton is live, the sufficiently long post-fault event sequences should be those with the length going to infinity. Hence we have the following definition to describe the prognosis probability $P_i(s)$ as:

**Definition 1:** For a given pre-fault event sequence $s \in L_{\text{pre}}(\tilde{G})$, the probability with which the faults will occur, denoted as $P_i(s)$, is defined as

$P_i(s) = \lim_{n \to \infty} \sum_{s' \in L_{i,j,\text{post}}(\tilde{G})} P_{\tilde{G}}(\tilde{G}(q_0, s), s')$.

The following will discuss how to calculate $P_i(s)$. Our idea is to construct an augmented automaton $\tilde{G}_{\delta(q_0, s)}$ which defines the language $L_i(\tilde{G})$ and then translate calculating $P_i(s)$ into calculating the probability of some states in a Markov chain.

We define a label set $\Delta = \{ N, F \}$. The label $N$ is interpreted as the “normal” behavior of the system, while $F$ means at least a fault has occurred (or say the fault event $f$ has occurred). We add a label $f (N$ or $F$ ) to state $q \in Q$ as $(q, l)$. $(q, N)$ means the system comes to state $q$ with the occurrence of a pre-fault event sequence and $(q, F)$ means the system comes to state $q$ with the occurrence of a post-fault event sequence.

Then the augmented automaton $\tilde{G}_{\delta(q_0, s)}$ is constructed as

$\tilde{G}_{\delta(q_0, s)} = (Z, \Sigma, \zeta, P_{\tilde{G}(q_0, s), z_0})$

$= Ac(Q \times \Delta, \Sigma, \zeta, P_{\tilde{G}(q_0, s), z_0})$

where $Z$ is the set of states of $\tilde{G}_{\delta(q_0, s)}$ and a state $z \in Z$ is of the form $z = (q, l)(l \in \Delta)$. Note that state $z$ is a doubleton. We use $z(1)$ to denote the first element and $z(2)$ to denote the second element. That is, for a given state $z = (q, l)$, $z(1) = q$ and $z(2) = l$. The transition function $\zeta$ is defined, for state $(q, l) \in Q \times \Delta$ and event $\sigma \in \Sigma$, as

$\zeta((q, l), \sigma) = \begin{cases} (\delta(q, \sigma), l) & \text{if } \delta(q, \sigma)! \text{ and } \sigma \neq f \\ (\delta(q, \sigma), F) & \text{if } \delta(q, \sigma)! \text{ and } \sigma = f \\ \text{undefined} & \text{others} \end{cases}$.

The state transition probability is denoted as

$p_{\tilde{G}(q_0, s), \delta}(\tilde{G}(q_0, s), \sigma) = p_{\tilde{G}(q_0, s), \delta}(\tilde{G}(q_0, s), \sigma)$. Initially automaton $\tilde{G}_{\delta(q_0, s)}$ stays in state $\delta(q_0, s)$ and no faults occur. Hence the initial state $z_0 = (\delta(q_0, s), N)$. Since $\zeta((q, l), \sigma)$! if and only if $\delta(q, \sigma)!$, the language generated by $\tilde{G}_{\delta(q_0, s)}$ equals to $L_i(\tilde{G})$, that is, $L(\tilde{G}(q_0, s)) = L_i(\tilde{G})$. In $\tilde{G}_{\delta(q_0, s)}$, we divide the state set $Z$ into two parts: the pre-fault state set $Z_p$ and the post-fault state set $Z_F$. They are defined as

$Z_p = \{ z \in Z : z(2) = N \}$ and $Z_F = \{ z \in Z : z(2) = F \}$.

$\tilde{G}_{\delta(q_0, s)}$ has the following properties.

**Property 1:** For any string $s' \in L_i(\tilde{G})$, we can calculate the probability of its occurrence by automaton $\tilde{G}_{\delta(q_0, s)}$ as

$Pr_{\tilde{G}}(\delta(q_0, s), s') = Pr_{\tilde{G}(q_0, s), \delta}(s')$.

**Proof:** Let us denote string $s'$ as $s' = \sigma_1, \sigma_2, \ldots, \sigma_m$. By the definition of $p_{\tilde{G}(q_0, s), \delta}$, we have
\[ \Pr_{\tilde{G}^{(q_0,s)}} (s') = \Pr_{\tilde{G}^{(q_0,s)}} (\sigma_1, \sigma_2, \ldots, \sigma_m) \]
\[ = p_{\tilde{G}^{(q_0,s)}} (z_0, 0, 1, \ldots, \sigma_1) \times p_{\tilde{G}^{(q_0,s)}} (\zeta (z_0, 0, \sigma_1), \sigma_2, \zeta (z_0, \sigma_1, \sigma_2)) \times \cdots \times p_{\tilde{G}^{(q_0,s)}} (\zeta (z_0, \sigma_1, \ldots, \sigma_{m-1}), \sigma_m, \zeta (z_0, \sigma_1, \ldots, \sigma_{m})) \]
\[ = p_{\tilde{G}} (z_0 (1, 0, 1, \ldots, \sigma_1)) \times p_{\tilde{G}} (\zeta (z_0, \sigma_1), \sigma_2, \zeta (z_0, \sigma_1, \sigma_2), (1)) \times \cdots \times p_{\tilde{G}} (\zeta (z_0, \sigma_1, \ldots, \sigma_{m-1}), \sigma_m, \zeta (z_0, \sigma_1, \ldots, \sigma_{m}), (1)) \]
\[ = p_{\tilde{G}} (\delta (q_0, s), \sigma_1, \delta (q_0, s, \sigma_1)) \times p_{\tilde{G}} (\delta (q_0, s, \sigma_1), \sigma_2, \delta (q_0, s, \sigma_1, \sigma_2)) \times \cdots \times p_{\tilde{G}} (\delta (q_0, s, \sigma_1, \ldots, \sigma_{m-1}), \sigma_m, \delta (q_0, s, \sigma_1, \ldots, \sigma_{m})) \]
\[ = \Pr_{\tilde{G}} (\delta (q_0, s), s, \sigma_1, \ldots, \sigma_{m}) \]
\[ = \Pr_{\tilde{G}} (\delta (q_0, s), s'). \]

Property 2: Any pre-fault string \( s' \in L_{s, pre} (\tilde{G}) \) drives automaton \( \tilde{G}^{(q_0,s)} \) into the pre-fault state set \( Z_N \), and any post-fault string \( s' \in L_{s, post} (\tilde{G}) \) drives automaton \( \tilde{G}^{(q_0,s)} \) into the post-fault state set \( Z_F \). In other words, for any string \( s' \in L_{\tilde{G}} \), \( s' \in L_{s, pre} (\tilde{G}) \) \( \Leftrightarrow \zeta (z_0, s') \in Z_N \wedge s' \in L_{s, post} (\tilde{G}) \) \( \Leftrightarrow \zeta (z_0, s') \in Z_F \).

Proof: We first prove \( s' \in L_{s, pre} (\tilde{G}) \Leftrightarrow \zeta (z_0, s') \in Z_N \) by induction on the length of string \( s' \in L_{\tilde{G}} \).

Base: initially \( |s'| = 0 \) and \( L_{s, pre} (\tilde{G}) = \{ e \} \subseteq L_{s, post} (\tilde{G}) \), we have \( \zeta (z_0, e) = z_0 \in Z_N \).

Induction hypothesis: Assume that for all strings \( s' \in L_{\tilde{G}} \) with \( |s'| \leq k \), \( s' \in L_{s, pre} (\tilde{G}) \Leftrightarrow \zeta (z_0, s') \in Z_N \).

Induction step: we show that, for any \( s' \sigma \in L_{s, pre} (\tilde{G}) \), \( s' \sigma \in L_{s, pre} (\tilde{G}) \Leftrightarrow \zeta (z_0, s' \sigma) \in Z_N \) as follows,
\( s' \sigma \in L_{s, pre} (\tilde{G}) \Leftrightarrow \zeta (z_0, s' \sigma) = \zeta (\zeta (z_0, s'), \sigma) \wedge f \notin s' \sigma \)
\( \Leftrightarrow \zeta (z_0, s' \sigma) = \zeta (\zeta (z_0, s'), \sigma) \wedge f \notin s' \wedge f \neq \sigma \)
\( \Leftrightarrow \zeta (z_0, s' \sigma) = \zeta (\delta (z_0 (1)), s', N), \sigma) \wedge f \neq \sigma \)
\( \Leftrightarrow \zeta (z_0, s' \sigma) = \delta (z_0 (1), s', N) \)
\( \zeta (z_0, s' \sigma) \in Z_N \)

We then prove \( s' \in L_{s, post} (\tilde{G}) \Leftrightarrow \zeta (z_0, s') \in Z_F \) as follows.
\( s' \in L_{s, post} (\tilde{G}) \Leftrightarrow s' \notin L_{s, pre} (\tilde{G}) \)
\( \Leftrightarrow \text{By the fact } L_s(\tilde{G}) = L_{s, pre} (\tilde{G}) \cup L_{s, post} (\tilde{G}) \)
\( \Leftrightarrow \zeta (z_0, s') \notin Z_N \)
\( \text{By the statement } s' \sigma \in L_{s, pre} (\tilde{G}) \Leftrightarrow \zeta (z_0, s', \sigma) \in Z_N \)
\( \Leftrightarrow \zeta (z_0, s') \in Z_F \)

We denote the prognosis probability of automaton \( \tilde{G}^{(q_0,s)} \) as \( P_f(\tilde{G}^{(q_0,s)}) \) and defined it as:
\[ P_f(\tilde{G}^{(q_0,s)}) = \lim_{i \to \infty} \bigcup_{l=1}^{\infty} \bigcup_{s' \in L_{s, post} (\tilde{G}^{(q_0,s)})} \Pr_{\tilde{G}^{(q_0,s)}} (s') \]
Then we have the following theorem to calculate \( P_f(s) \).

**Theorem 1:** The prognosis probability of pre-fault event sequence \( s \) equals to the prognosis probability of automaton \( \tilde{G}^{(q_0,s)} \) as
\[ P_f(s) = P_f(\tilde{G}^{(q_0,s)}) \]

Proof:
\[ P_f(s) = \lim_{i \to \infty} \bigcup_{s' \in L_{s, post} (\tilde{G}^{(q_0,s)})} \Pr_{\tilde{G}^{(q_0,s)}} (s') \text{ (By Definition 1)} \]
\[ = \lim_{i \to \infty} \bigcup_{s' \in L_{s, post} (\tilde{G}^{(q_0,s)})} \Pr_{\tilde{G}^{(q_0,s)}} (s') \text{ (By Property 1)} \]
\[ = P_f(\tilde{G}^{(q_0,s)}) \]

We divide language \( L(\tilde{G}^{(q_0,s)}) \) into two parts: One is the set of strings which drive automaton \( \tilde{G}^{(q_0,s)} \) into “good” states in \( Z_N \) as
\[ L_{Z_N}(\tilde{G}^{(q_0,s)}) = \{ s' \in L(\tilde{G}^{(q_0,s)}), \zeta (z_0, s') \in Z_N \} \]
The other is the set of strings which drive automaton \( \tilde{G}^{(q_0,s)} \) into “bad” states in \( Z_F \) as
\[ L_{Z_F}(\tilde{G}^{(q_0,s)}) = \{ s' \in L(\tilde{G}^{(q_0,s)}), \zeta (z_0, s') \in Z_F \} \]

We further divide \( L_{Z_\sigma}(\tilde{G}^{(q_0,s)}) \) into a serial of subsets based on the length of strings
\[ L_{Z_{\sigma}}(\tilde{G}^{(q_0,s)}) = \{ s' \in L_{\tilde{G}^{(q_0,s)}}, |s'| = i \} \text{ } i = 0, 1, 2, \ldots \]
and divide \( L_{Z_F}(\tilde{G}^{(q_0,s)}) \) into a serial of subsets as
\[ L_{Z_{\sigma}}(\tilde{G}^{(q_0,s)}) = \{ s' \in L_{\tilde{G}^{(q_0,s)}}, |s'| = i \} \text{ } i = 0, 1, 2, \ldots \]

Then we have
**Theorem 2:** The prognosis probability of automaton \( \tilde{G}^{(q_0,s)} \) is the sum of probabilities of those infinite event sequences goes to “bad” states. That is,
\( P_f(G_{\delta(q_0,s)}) = \lim_{i \to \infty} \sum_{s \in L_{z,F}} (G_{\delta(q_0,s)}) \Pr_{G_{\delta(q_0,s)}}(s') \)

**Proof:** From the definition of \( P_f(G_{\delta(q_0,s)}) \), we have

\[
P_f(G_{\delta(q_0,s)}) = \lim_{i \to \infty} \sum_{s \in L_{z,F}} (G_{\delta(q_0,s)}) \Pr_{G_{\delta(q_0,s)}}(s') = \lim_{i \to \infty} \sum_{s \in L_{z,F}} (G_{\delta(q_0,s)}) \Pr_{G_{\delta(q_0,s)}}(s')
\]

(By Property 2)

\[
P_f(G_{\delta(q_0,s)}) = \lim_{i \to \infty} \sum_{s \in L_{z,F}} (G_{\delta(q_0,s)}) \Pr_{G_{\delta(q_0,s)}}(s').
\]

A corollary then follows.

**Corollary 1:** The prognosis probability of pre-fault string \( s \) can be determined as

\[
P_f(s) = \lim_{i \to \infty} \sum_{s \in L_{z,F}} (G_{\delta(q_0,s)}) \Pr_{G_{\delta(q_0,s)}}(s').
\]

We can view the stochastic automaton \( G_{\delta(q_0,s)} \) as a Markov chain if we remove the event labels associated with the transitions. The Markov chain is denoted as

\[ M_{G_{\delta(q_0,s)}} = (Z, p_M, \pi_0) \]

where \( Z \) is the state set of \( G_{\delta(q_0,s)} \) and \( \pi_0 \) is the initial state probability distribution vector of \( G_{\delta(q_0,s)} \) as

\[
\pi_0 = [\pi_0(z_0), \pi_0(z_1), \pi_0(z_2), \ldots] = [1,0,0,\ldots,0].
\]

For states \( z_i, z_j \in Z \), \( p_M(z_i, z_j) \) is the state transition probability from \( z_i \) to \( z_j \) and defined as

\[
p_M(z_i, z_j) = \sum_{\sigma \in \Sigma} p_{G_{\delta(q_0,s)}}(z_i, \sigma, z_j).
\]

With a slight abuse of notations, we use \( p_{z_i} \) to denote the state transition probability matrix. Its \((i,j)\)th entry is the state transition probability \( p_M(z_i, z_j) \) from state \( z_i \) to state \( z_j \).

For the Markov chain \( M_{G_{\delta(q_0,s)}} \), the state probability distribution vector at step \( n \) is denoted as \( \pi_n \). The vector \( \pi_n \) can be calculated recursively as \( \pi_n = \pi_{n-1} \times p_M \) with the initial state probability distribution vector \( \pi_0 \). We use \( \pi_i(z_j) \) to denote the probability of state \( z_j \) in the \( i \)th step.

As discussed in the paper [13], the Markov chain satisfies:

**Theorem 3:** The probability of language \( L_{z,F}(G_{\delta(q_0,s)}) \) in \( G_{\delta(q_0,s)} \) equals to the sum of probabilities of states in \( Z_F \) of \( \pi_n \). In other words,

\[
\bigcup_{s \in L_{z,F}} (G_{\delta(q_0,s)}) \Pr_{G_{\delta(q_0,s)}}(s') = \bigcup_{s \in Z_F} \pi_n(s).
\]

With the results in Theorem 3, we have the following theorem to calculate \( P_f(s) \).

**Theorem 4:** The prognosis probability of pre-fault event sequence \( s \) can be determined by Markov chain \( M_{G_{\delta(q_0,s)}} \) as

\[
P_f(s) = \lim_{i \to \infty} \sum_{s \in L_{z,F}} (G_{\delta(q_0,s)}) \Pr_{G_{\delta(q_0,s)}}(s') = \lim_{i \to \infty} \sum_{s \in L_{z,F}} (G_{\delta(q_0,s)}) \Pr_{G_{\delta(q_0,s)}}(s').
\]

(By Theorem 3)

**IV. PROGNOSTIC PROBABILITY OF PRE-FAULT OBSERVABLE EVENT SEQUENCES**

Generally speaking, some events are observable and the other events are unobservable in practical systems. Hence what we see is not the occurred event sequence, but its observation (an observable event sequence). When we see the observable event sequence, we want to know the probability with which the fault event will occur.

For the given system \( G \), the set of observable event sequences, denoted as \( M(L(G)) \), is

\[
L_s(G) = M(L(G)) = \{s \in L(G) : s \in s\}.
\]

For observe observable event sequence \( t \in L_o(G) \), the possible occurred event sequences are all in \( M^{-1}_o(t) \). When we discuss prognosis probability, the fault event should not have occurred. In other words, every event sequence in \( M^{-1}_o(t) \) should be pre-fault. We say an observable event sequence \( t \in M(L(G)) \) is pre-fault if the entire event sequences in \( M^{-1}_o(t) \) are pre-fault. Then we get a subset which includes all the pre-fault observable event sequences as

\[
L_o,pre(G) = \{t \in L_o(G) : M^{-1}_o(t) \subseteq L_o,pre(G)\}.
\]

The prognosis probability \( P_f(t) \) should be the sum of the prognosis probabilities of these event sequences in \( M^{-1}_o(t) \). Considering the occurrence probabilities of different event sequences in \( M^{-1}_o(t) \) are different, the prognosis probability for any pre-fault observable event sequence \( t \in L_o,pre(G) \) should be defined as

**Definition 2:** For pre-fault observable event sequence \( t \in L_o,pre(G) \), the prognosis probability with which the faults will occur, denoted as \( P_f(t) \), is defined as

\[
P_f(t) = \frac{\bigcup_{s \in M^{-1}_o(t)} \Pr_{G}(s) \times P_f(s)}{\bigcup_{s \in M^{-1}_o(t)} \Pr_{G}(s)}.
\]

Note that \( \Pr_{G}(s) \) is the occurrence probability of string \( s \) and \( P_f(s) \) is the prognosis probability of faults with the occurrence of string \( s \).

The following will discuss how to calculate it. When we observe the occurrence of observable event sequence \( t \), the current state probability distribution is denoted with a vector \( \phi(t) \) as:
\[ \phi(t) = [\phi(t)(q_0), \phi(t)(q_1), \ldots, \phi(t)(q_d)] \]

where \( \phi(t)(q_i) \) is the probability with which the system stays in state \( q_i \) after we observe event sequence \( t \). It is calculated as

\[ \phi(t)(q_i) = \frac{\sum_{s \in M^2(t) \cap \delta(q_i,s)} p_{G}(s)}{\sum_{s \in M^2(t)} p_{G}(s)} \]

Then we have:

**Theorem 5:** The prognosis probability of pre-fault observable event sequence \( t \) can be determined by the current state probability distribution \( \phi(t) \) and the prognosis probability of automaton \( \widetilde{G}_q \) as

\[ P_f(t) = \frac{\sum_{q_i \in Q} \phi(t)(q_i) P_f(q_i)}{\sum_{q_i \in Q} \phi(t)(q_i)} \]

where \( P_f(G_q) \) is the prognosis probability with which the faults will occur given that the system stays in state \( q_i \).

**Proof:** By the definition of \( P_f(t) \), we have

\[ P_f(t) = \frac{\sum_{q_i \in Q} \phi(t)(q_i) P_f(q_i)}{\sum_{q_i \in Q} \phi(t)(q_i)} \]

(B) for every state \( q_i \), \( \phi(t)(q_i) \)

Calculating \( P_f(G_q) \) has been solved in the above section.

The following will consider how to calculate \( \phi(t) \) for observable string \( t \). Let \( L_c(\widetilde{G}, q_i, q_j) \) represents the set of all the strings which drive automaton \( \widetilde{G} \) from state \( q_i \) to state \( q_j \) when we observe event \( \sigma \in \Sigma_c \). \( L_c(\widetilde{G}, q_i, q_j) \) is defined as

\[ L_c(\widetilde{G}, q_i, q_j) = \{ s \in \Sigma_c : s \in M^{-1}(\sigma) \cap \delta(q_i,s) = q_j \} \]

We denote the state transition probability matrix as \( MTX_{\sigma} \). Its \( (i,j) \)th entry, which means the transition probability from state \( q_i \) to state \( q_j \) when we see observable event \( \sigma \), is defined as:

\[ MTX_{\sigma}(i,j) = \sum_{q_i \in Q} \phi(t)(q_i) P_f(q_i) \]

For every observable event \( \sigma \in \Sigma_c \), we obtain its state probability transition matrix \( MTX_{\sigma} \) by calculating every entry \( MTX_{\sigma}(i,j) \). We then calculate \( \phi(t) \) recursively as follows.

**Theorem 6:** For observable event sequence \( t = \sigma \sigma_2 \ldots \sigma_{m-1} \in L_c(\widetilde{G}) \), the current state probability distribution vector \( \phi(\sigma \sigma_2 \ldots \sigma_{m-1}) \) is calculated recursively as

\[ \phi(\sigma \sigma_2 \ldots \sigma_{m-1}) = NORM(\phi(\sigma \sigma_2 \ldots \sigma_{m-1}) \times MTX_{\sigma_{m-1}}) \]

where \( NORM \) is the normalization operation.

For the interest of space, we transfer the proof of Theorem 6 into VII. Appendix.

V. ONLINE PROGNOSIS

Now we discuss the online prognosis problem. The problem is to calculate the prognosis probability for observed event sequences online. Assume the observation sequence is \( x(1)x(2) \cdots x(i-1) \) (Note that it is an observation event sequence), for a new observation \( x(i) \) (an observable event), we want to update the prognosis probability \( P_f(i) \) online such that

\[ P_f(i) = P_f(x(1)x(2) \cdots x(i-1)x(i)) \]

The following will consider how to update the prognosis probability \( P_f(i) \) for every output \( x(i) \) online. We first should ensure the fault has not occurred. In other words, the observed event sequence \( x(1)x(2) \cdots x(i) \) is pre-fault. If it is not pre-fault, calculating the prognosis probability \( P_f(i) \) becomes trivial. Let us first consider how to check whether \( x(1)x(2) \cdots x(i) \) is pre-fault or not when we observe \( x(i) \).

We use the approach in Section 3 to construct an augmented stochastic automaton \( \widetilde{G}_q \)

\[ \widetilde{G}_q = (Z, \Sigma, \zeta, p_{\widetilde{G}_q}, (q_0, N)) \]

Stochastic automaton \( \widetilde{G}_q \) has the following properties:

1) \( L(\widetilde{G}_q) = L(\widetilde{G}) \);
2) \( L_c(\widetilde{G}_q) = L_c(\widetilde{G}) \);
3) \( (\forall s \in L_{\text{pre}}(\widetilde{G})) \zeta((q_0, N), s) = (\delta(q_0, s), N) \);
4) \( (\forall s \in L_{\text{pos}}(\widetilde{G})) \zeta((q_0, N), s) = (\delta(q_0, s), F) \).

**Proof:** Forwardly got it.

We then define the current state estimate \( SE(t) \) for observable event sequence \( t \) in \( \widetilde{G}_q \) as:

\[ SE(t) = \{ z : (\exists s \in M_{\widetilde{G}_q}^{-1}(t)) z = \zeta((q_0, N), s) \} \]

Note that \( M_{\widetilde{G}_q}^{-1}(t) = M_{\widetilde{G}}^{-1}(t) \).

With the current state estimate \( SE(t) \), we can check whether \( t \) is pre-fault or not by the following theorem.

**Theorem 7:** The observed event sequence \( x(1)x(2) \cdots x(i) \) is pre-fault if and only if all the states in the state estimate \( SE(x(1)x(2) \cdots x(i)) \) are “good” states. That is,
Proof: Indeed, we have
\[
x(1)x(2)\cdots x(i) \in L_{o,pre}(\tilde{G})
\]
\[
\iff (\forall z \in SE(x(1)x(2)\cdots x(i)))z(2) = N.
\]
(By the definition of \(L_{o,pre}(\tilde{G})\))

\[
\iff (\forall s \in M^{-1}(x(1)x(2)\cdots x(i)))s \in L_{o,pre}(\tilde{G})
\]
(By the third and fourth statements of Property 3)

(\(x(1)x(2)\cdots x(i))\) is the state estimate)

For a state estimate \(y \subseteq Z\), we denote its unobservable reach as:

\[
UR(y) = \{z : (\exists s' \in y)(\exists s \in \sum^*_o)z = \zeta(z', s)\}.
\]

With the occurrence of observable event \(\sigma\), we denote its observable reach as:

\[
OR(y, \sigma) = \{z : (\exists s' \in y)z = \zeta(z', \sigma)\}.
\]

Then we can recursively calculate the current state estimate \(SE(x(1)x(2)\cdots x(i))\) as shown in the following theorem.

**Theorem 8:** For the observed event sequence \(x(1)x(2)\cdots x(i)\), its state estimate can be calculated recursively as:

\[
SE(x(1)x(2)\cdots x(i)) = OR(UR(SE(x(1)x(2)\cdots x(i-1))), x(i)).
\]

**Proof:** By the definition of \(M^{-1}(\cdot)\), we know, for any string \(s \in M^{-1}(x(1)x(2)\cdots x(i))\), it can be re-write as

\[
s = s's''x(i)
\]

where, \(s' \in M^{-1}(x(1)x(2)\cdots x(i-1))\) and \(s'' \in \sum^*_o\).

With the fact, let us prove the theorem as follows.

The prediction probability for the \(i\)th output \(x(i)\) is denoted as:

\[
\frac{SE(x(1)x(2)\cdots x(i))}{\sum^*_o}
\]

\[
\{z : (\exists s \in M^{-1}(x(1)x(2)\cdots x(i)))s = (\delta(q_o, s), N)\}
\]

\[
By the definition of \(SE(\cdot)\)
\]

\[
\{z : (\exists s' \in M^{-1}(x(1)x(2)\cdots x(i-1)))(\exists s'' \in \sum^*_o)z = \zeta((q_o, N), s''', x(i))\}
\]

\[
By the fact of \(s = s's''x(i)\)
\]

\[
\{z : (\exists s' \in M^{-1}(x(1)x(2)\cdots x(i-1)))(\exists s'' \in \sum^*_o)z = \zeta((q_o, N), s', s''x(i))\}
\]

\[
By the definition of \(SE(\cdot)\)
\]

\[
\{z : (\exists s' \in M^{-1}(x(1)x(2)\cdots x(i-1)))(\exists s'' \in \sum^*_o)z = \zeta((q_o, N), s'', x(i))\}
\]

\[
By the definition of \(OR(\cdot)\)
\]

Now we propose an online prognoser to calculate the prognosis probability for every observed event sequence. We use \(SE(i), \phi(i)\) to denote the current state estimate and the current state probability distribution vector when the online prognoser observes the \(i\)th output \(x(i)\). The prognoser runs as follows:

Initially, there are no observations. We calculate the current state estimate \(SE(0)\), the current state probability distribution vector \(\phi(0)\) and the current prognosis probability \(P_f(0)\) as:

\[
SE(0) = (q_o, N);
\]

\[
\phi(0) = [1, 0, \cdots, 0];
\]

\[
P_f(0) = P_f(\tilde{G}_o).
\]

Whenever one new output \(x(i)\) is observed, the online prognoser updates the current state estimate and the current state probability distribution vector as:

\[
SE(i) = OR(UR(SE(i-1)), x(i)));
\]

\[
\phi(i) = NORM(\phi(i-1) \times MTX_{x(i)}).
\]

With the current state estimate \(SE(i)\), we update the current prognosis probability as:

\[
P_f(i) = \begin{cases} \bigcup_{q_o \in \phi(i)} \phi(i)(q_o)P_f(\tilde{G}_o) & \text{if } (\forall z \in SE(i))z(2) = N \\ \text{None} & \text{otherwise} \end{cases}
\]

The following theorem shows the prognoser is correct. In other words, the prognosis probability \(P_f(i)\) is always what we want.

**Theorem 9:** The updated prognosis probability \(P_f(i)\) by the online prognoser is always correct such that...
\[ P_j(i) = \begin{cases} 
  P_j(x(1)x(2)x(3)\cdots x(i)) & \text{if } x(1)x(2)x(3)\cdots x(i) \\
  None & \text{otherwise} 
\end{cases} \]

\[ x(i) \in L_{\epsilon, \text{pre}}(\hat{G}) \]

**Proof:** 
We first prove \((\forall i \in N)SE(i) = SE(x(1)x(2)x(3)\cdots x(i))\) by induction on \(i\).

**Base:**
Initially \(i = 0\). We have \(SE(i) = SE(0) = (q_0, N) = SE(\epsilon)\).

**Induction hypothesis:** Assume that for all \(i \leq k\), we have \(SE(i) = SE(x(1)x(2)x(3)\cdots x(i))\).

**Induction step:**
We show that \(SE(k + 1) = SE(x(1)x(2)x(3)\cdots x(k)x(k + 1))\) as follows,

\[ SE(k + 1) = OR(UR(SE(k)), x(k + 1)) \]

\[ = OR(UR(SE(x(1)x(2)x(3)\cdots x(k))), x(k + 1)) \]

(by induction hypothesis)

\[ = SE(x(1)x(2)x(3)\cdots x(k)x(k + 1)) \]

(by Theorem 8)

We then prove \((\forall i \in N)\phi(i) = \phi(0) = [1, 0, 0, \cdots] = \phi(\epsilon)\).

**Induction hypothesis:** Assume that for all \(i \leq k\), we have \(\phi(i) = \phi(x(1)x(2)x(3)\cdots x(i))\).

**Induction step:**
We show that \(\phi(k + 1) = \phi(x(1)x(2)x(3)\cdots x(k)x(k + 1))\) as follows,

\[ \phi(k + 1) = \text{NORM}(\phi(k) \times \text{MTX}_{x(k+1)}) \]

\[ = \text{NORM}(\phi(x(1)x(2)x(3)\cdots x(k))) \times \text{MTX}_{x(k+1)} \]

(by induction hypothesis)

\[ = \phi(x(1)x(2)x(3)\cdots x(k)x(k + 1)) \]

(by Theorem 6)

With the two results, let us now prove the theorem as

\[ x(1)x(2)x(3)\cdots x(i) \in L_{\epsilon, \text{pre}}(\hat{G}) \]

\[ \Leftrightarrow (\forall z \in SE((x(1)x(2)x(3)\cdots x(i)))z(2) = N \]

(by Theorem 7)

\[ \Leftrightarrow (\forall z \in SE(i))z(2) = N \]

(by the fact of \(SE(i) = SE(x(1)x(2)x(3)\cdots x(i))\))

\[ \Leftrightarrow P_j(i) = \bigcup_{q_j, \phi} \phi(i) \phi(q_j) P_j(\hat{G}_q) \]

(by the update of \(P_j(i)\))

\[ \Leftrightarrow P_j(i) = \bigcup_{q_j, \phi} \phi(i) \phi(q_j) P_j(\hat{G}_q) \]

(by the fact of \(\phi(i) = \phi(x(1)x(2)x(3)\cdots x(i))\))

\[ \Leftrightarrow P_j(i) = P_j(x(1)x(2)x(3)\cdots x(i)) \]

(by Theorem 5)

**VI. CONCLUSIONS**

This paper investigates how to calculate the prognosis probability of stochastic discrete event systems. The main contributions are summarized as follows: 1) We define the prognosis probability for pre-fault event sequences and find effective algorithms to calculate it. 2) We further define the prognosis probability for pre-fault observable event sequences and find effective algorithms to calculate it. 3) By updating the current state estimate, the current state probability distribution vector, and the current prognosis probability with an online prognoser, the online prognosis problem is successfully solved.

**REFERENCES**


Proof of Theorem 6

We re-write string $t = t' \sigma_m$ ($t' = \sigma_1 \sigma_2 \cdots \sigma_{m-1}$) and enumerate all the states in $Q$ as $q_0, q_1, \ldots, q_n$. We then have: $\phi(\sigma_1 \sigma_2 \cdots \sigma_m) = \text{NORM}\left(\phi(\sigma_1 \sigma_2 \cdots \sigma_{m-1}) \times \text{MTX}_{\sigma_m}\right)$. The detailed proof is supplied as follows.

\[
\phi(t' \sigma_m) = \phi(t' \sigma_m) = \left(\frac{\sum_{s \in M_G(t' \sigma_m)} \text{Pr}_G(s)}{\sum_{s \in M_G(t' \sigma_m)} \text{Pr}_G(s)}, \ldots, \frac{\sum_{s \in M_G(t' \sigma_m)} \text{Pr}_G(s)}{\sum_{s \in M_G(t' \sigma_m)} \text{Pr}_G(s)}\right)
\]

\[
= \text{NORM}\left(\sum_{s \in M_G(t' \sigma_m)} \text{Pr}_G(s) \times \phi(q_0, s') \cdots \sum_{s \in M_G(t' \sigma_m)} \text{Pr}_G(s) \times \phi(q_n, s')\right)
\]

\[
= \text{NORM}\left(\left\{\sum_{s \in M_G(t' \sigma_m)} \text{Pr}_G(s) \times \phi(q_0, s') \cdots \sum_{s \in M_G(t' \sigma_m)} \text{Pr}_G(s) \times \phi(q_n, s')\right\}\right)
\]

\[
= \text{NORM}\left(\phi(t') \times \text{MTX}_{\sigma_m}\right)
\]

\[
= \text{NORM}\left(\phi(\sigma_1 \sigma_2 \cdots \sigma_m) \times \text{MTX}_{\sigma_m}\right)
\]