Discrete Event Dynamic Systems (2023) 33:105–127
https://doi.org/10.1007/s10626-023-00374-y

Supervisory control to maximize mean time to failure in discrete event systems

Feng Lin[1] · **Caisheng Wang**[1] · **Masoud H. Nazari**[1] · **Wenyuan Li**[2]

Received: 2 February 2022 / Accepted: 1 April 2023 / Published online: 10 June 2023

Abstract
In this paper, we investigate the use of supervisory control to maximize mean time to failure in a discrete event system framework. A complex engineering system is modeled as a discrete event system. Some of the states of the system are essential to the functionality of the system and are called required states. Some other states represent failures in the system and are called failure states. The control objective is to maximize the mean time to failure (MTTF) while allowing the system to visit all required states. The control is achieved by a supervisor that disables some controllable events based on monitoring the observable events as in classical supervisory control. To design such a supervisor, the MTTF of a supervised system is calculated by converting a discrete event system into a Markov chain having the same MTTF. Based on MTTF, two algorithms are developed that together allow us to design an optimal supervisor. The theoretical results are applied to power systems by investigating the maintenance management of equipment such as transformers.

Keywords Discrete event systems · Time to failure · Supervisory control · Markov chain

1 Introduction

Discrete event systems are first introduced in the 1980s to model man-made systems with discrete states and discrete events. The dynamics of these systems cannot be modeled by differential or difference equations. Rather, their dynamics are described by occurrences of events that move the system from one discrete state to another. Several theories of discrete event systems have been developed. Among them, supervisory control theory is developed to control a discrete event system so that the controlled system is safe and live. Here safety means that the controlled system will never enter some illegal/unsafe states, while liveness means that the controlled system will eventually enter some final/marked states. Important concepts such as controllability Ramadge and Wonham (1987), observability Lin and Wonham

✉ Feng Lin
flin@wayne.edu

[1] Department of Electrical and Computer Engineering, Wayne State University, MI, Detroit 48202, USA

[2] School of Electrical Engineering, Chongqing University, 400044 Chongqing, China

(1988); Cieslak etal (1988), and co-observability Rudie and Wonham (1992) are introduced in supervisory control. A history of supervisory control can be found in Wonham and Cai (2019).

Fault diagnosis and diagnosability have also been investigated extensively using discrete event systems. Diagnosability of discrete event systems is first introduced in Sampath etal (1995); Lin (1994), where the goal is to diagnose failures in a system. A discrete event system is said to be diagnosable if any failure in the system can be diagnosed within a bounded number of observations of observable events. If a discrete event system is diagnosable, then a diagnoser can be designed to diagnose failures. Polynomial algorithms to check diagnosability are developed in Yoo and Lafortune (2002); Jiang etal (2001). A history of diagnosability of discrete event systems can be found in Lafortune etal (2018).

When we design and control a complex system, such as a power system, an airplane, or an automobile, we often want to know how long we can expect the system to run before some failures occur. In other words, we would like to know the mean time to failure (MTTF) of a system. The MTTF of a system depends on many factors, such as the MTTF of its components, how often the system is inspected and maintained, whether it operates in states that are prone to failures. In this paper, we investigate MTTF in the framework of discrete event systems.

To this end, we assume that some of the discrete states in a system are failure states. To increase MTTF, we can control the system by preventing it from entering some prone-to-failure states (states that have small MTTF) and by introducing maintenance. To design such a control rigorously, we use the supervisory control theory of discrete event systems. That is, control is implemented by a supervisor, which observes observable events and controls controllable events by disabling them if needed. The goal of a supervisor is to ensure that the set of possible trajectories/strings of events generated by the supervised system equals to a given specification language. This language specifies which prone-to-failure states shall be avoided and which maintenance shall be performed. It is proven in supervisory control theory that there exists a supervisor achieving the specification language if and only if the language is controllable and observable Ramadge and Wonham (1987); Lin and Wonham (1988). If the specification language is not controllable and observable, we can find its smallest superlanguage that is controllable and observable. Algorithms are available to do this formally and systematically, see Ramadge and Wonham (1987); Wonham etal (2018); Lin and Wonham (1988); Cassandras and Lafortune (2009), for example.

For a given supervisor, we would like to know the MTTF of the supervised system. This requires that we know the probability distributions of event lifetimes (the time it takes for the event to occur after it is allowed to occur), called lifetime distributions. In this paper, we assume that lifetime distributions are exponential with known means. To calculate MTTF, we use the existing results on first-passage time of continuous time Markov Chains Darling and Siegert (1953); Brown and Chaganty (1983); Yao (1985); Hunter (2018).

Clearly, discrete event systems and Markov chains are introduced for different purposes. While discrete event systems are used to model and control various systems to ensure safety and liveness, Markov chains are used to capture stochastic features of systems. However, for the purpose of calculating MTTF, a discrete event system can be converted into a Markov chain that has the same MTTF. Note that this particular Markov chain is for calculating MTTF only; the supervisory control is applied to the original discrete event system.

After developing a method to calculate MTTF for a given supervisor, we then pursue synthesis of an optimal supervisor as follows. We divide the states of the system to be controlled into three types: (1) failure states, (2) required states, that is, states that are essential to the function of the system, and (3) the remaining states. We start with the required states

and design a supervisor that reaches all required states. We check the remaining states to determine if they shall be removed or allowed in the controlled system. This is done in two steps. First, we define and calculate the prone-to-failure measures for the remaining states and order the remaining states according to this measure. Second, we add the remaining states to the supervised system one by one, starting from the least prone to failures state, to see if adding a state improves MTTF. If so, the state is added and the supervisor is re-designed. Otherwise, the state is not added. The computational complexity of synthesizing such an optimal supervisor is polynomial with respect to the number of states and hence can be used for large-scale systems.

We apply the results to power systems and investigate how to manage the maintenance of equipment such as transformers. Transformer failures have been investigated in power systems Jan etal (2015); Murugan and Ramasamy (2015); Zhong etal (2016); authorname (2018). Our approach can be used to calculate the MTTF of a transformer and to increase the MTTF by proper maintenance. For transformers considered in Zhong etal (2016), we show that, compared with the MTTF of the transformer without major and minor overhauls, its MTTF can be increased by 37% (from 11.1 years to 15.1 years) if major overhauls are performed and by 351% (from 11.1 years to 49.8 years) if both major and minor overhauls are performed.

The main contributions of the paper are as follows: (1) We introduce a formal definition of MTTF in a discrete event system. (2) We develop a method to calculate MTTF by converting a discrete event system into a Markov chain. (3) We propose to use a supervisor to maximize MTTF by removing prone-to-failure states and allowing (not removing) other states. (4) We develop a systematic approach to synthesize such a supervisor.

The paper is organized as follows. In Section 2, we briefly review discrete event systems and supervisory control. An automaton is used to model a discrete event system. Supervisor design procedure is outlined to achieve a specification language if the language is controllable and observable. In Section 3, random event lifetimes are introduced in discrete event systems. Operating rules of the resulting stochastic discrete event systems are defined. Based on these rules, times to failure are formally defined in Section 4. Section 5 investigates how to calculate MTTF. To this end, a stochastic discrete event system is first converted to a Markov chain. Results on the first-passage time of Markov Chain are then used to derive MTTF. In Section 6, we investigate how to use supervisory control to maximize MTTF. We start with a required language and calculate MTTF for this language. We then check if MTTF can be increased by removing or adding more states to the required language. The result is an optimal supervisor that maximizes MTTF. We apply the results to the maintenance management of transformers in power systems in Section 7.

2 Discrete event systems and supervisory control

In this section, we review results in discrete event systems and supervisory control. A discrete event system is modeled as an *automaton* (also called a finite state machine) denoted by

$$\mathcal{A} = (Y, \Sigma, \zeta, y_o), \tag{1}$$

where Y is the set of (discrete) states; Σ is the set of events, $\zeta : Y \times \Sigma \to Y$ is the (partial) transition function; and $y_o \in Y$ is the initial state. The transition function can be extended to $\zeta : Y \times \Sigma^* \to Y$, where Σ^* is the set of all strings over Σ, including the empty string

ε. With slight abuse of notation, the set of all possible transitions of $\mathcal{A}$ is also denoted by $\zeta = \{(y, \sigma, y') : \zeta(y, \sigma) = y'\}$.

$\mathcal{A}$ describes a plant (the system to be controlled). Its trajectory is described by a string of events $s \in \Sigma^*$. We use ! to express "is defined". When $\zeta(y_o, s)$ is defined, that is, $\zeta(y_o, s)!$, we say that s can be generated by $\mathcal{A}$. The set of all strings that can be generated by $\mathcal{A}$ is called *language generated* by $\mathcal{A}$, which is defined as

$$L(\mathcal{A}) = \{s \in \Sigma^* : \zeta(y_o, s)!\}$$

$L(\mathcal{A})$ describes the behavior of the uncontrolled system. A supervisor is used to control or supervise the system so that the strings generated by the closed-loop (or supervised) system are safe and admissible. This control requirement is described by a specification language $K \subseteq L(\mathcal{A})$.

The (prefix) closure of a language K is the set of all prefixes of all strings in the language. A language is closed if it equals its closure. By definition, $L(\mathcal{A})$ is closed.

Not all events in $\mathcal{A}$ are controllable in the sense that their occurrences can be disabled. For example, failure events are not controllable because we cannot always prevent failures from occurring. Some events may not be observable in the sense that their occurrences cannot be observed. This is mainly because observing events requires sensors that may be too difficult or expensive to install. The set of controllable events is denoted by $\Sigma_c \subseteq \Sigma$. The set of uncontrollable events is denoted by $\Sigma_u = \Sigma - \Sigma_c$. The set of observable events is denoted by $\Sigma_o \subseteq \Sigma$. If a string $s \in L(\mathcal{A})$ occurs in $\mathcal{A}$, then the supervisor observes $P(s)$, where $P : \Sigma^* \to \Sigma_o^*$ is the *projection*, defined iteratively as follows. For $s \in \Sigma^*$ and $\sigma \in \Sigma$,

$$P(\varepsilon) = \varepsilon, \quad P(s\sigma) = \begin{cases} P(s)\sigma & \text{if } \sigma \in \Sigma_o \\ P(s) & \text{otherwise.} \end{cases} \tag{2}$$

The projection P can be extended from a string to a language Lin and Wonham (1988). $P(L(\mathcal{A}))$ represents the set of all possible observations.

A *supervisor* can now be defined formally as a mapping

$$\mathcal{S} : P(L(\mathcal{A})) \to 2^{\Sigma}. \tag{3}$$

Intuitively, after observing $t \in P(L(\mathcal{A}))$, supervisor $\mathcal{S}$ enables the events in $\mathcal{S}(t)$. Since uncontrollable events cannot be disabled, we require that $(\forall t \in P(L(\mathcal{A})))\Sigma_u \subseteq \mathcal{S}(t)$.

The supervised system is denoted by $\mathcal{S}/\mathcal{A}$. The language generated by $\mathcal{S}/\mathcal{A}$, denoted by $L(\mathcal{S}/\mathcal{A})$, is defined iteratively as follows:

$$\begin{aligned} &\varepsilon \in L(\mathcal{S}/\mathcal{A}), \\ &(\forall s \in L(\mathcal{S}/\mathcal{A}))(\forall \sigma \in \Sigma) s\sigma \in L(\mathcal{S}/\mathcal{A}) \Leftrightarrow s\sigma \in L(\mathcal{A}) \wedge \sigma \in \mathcal{S}(P(s)). \end{aligned} \tag{4}$$

In other words, if s occurs in $\mathcal{S}/\mathcal{A}$ ($s \in L(\mathcal{S}/\mathcal{A})$), then a new event σ can occur in $\mathcal{S}/\mathcal{A}$ ($s\sigma \in L(\mathcal{S}/\mathcal{A})$) if and only if σ can occur in $\mathcal{A}$ ($s\sigma \in L(\mathcal{A})$) and σ is enabled by $\mathcal{S}$ ($\sigma \in \mathcal{S}(P(s))$).

As mentioned above, the objective of a supervisor is to restrict the behavior of the system so that $L(\mathcal{S}/\mathcal{A}) = K$ for a given specification language $K \subseteq L(\mathcal{A})$. Since the supervisor cannot observe and control all events, this objective may or may not be achievable. To find a necessary and sufficient condition for achieving the objective, two important concepts are introduced.

(1) *Controllability* Ramadge and Wonham (1987): A language $K \subseteq L(\mathcal{A})$ is controllable if

$$K\Sigma_u \cap L(\mathcal{A}) \subseteq K. \tag{5}$$

(2) *Observability* Lin and Wonham (1988): A language $K \subseteq L(\mathcal{A})$ is observable if

$$(\forall s, s' \in K)(\forall \sigma \in \Sigma)P(s) = P(s') \wedge s\sigma \in K \wedge s'\sigma \in L(\mathcal{A}) \Rightarrow s'\sigma \in K. \quad (6)$$

It is proven in Lin and Wonham (1988) that there exists a supervisor $\mathcal{S}$ such that $L(\mathcal{S}/\mathcal{A}) = K$ if and only if K is controllable and observable.

If K is controllable and observable, then a state-based supervisor $\mathcal{S}$ satisfying $L(\mathcal{S}/\mathcal{A}) = K$ can be designed. To do so, we assume, without loss of generality, that K is generated by the following subautomaton of $\mathcal{A}$:

$$\mathcal{R} = (Y_{\mathcal{R}}, \Sigma, \zeta_{\mathcal{R}}, y_o), \quad (7)$$

where $Y_{\mathcal{R}} \subseteq Y$ and $\zeta_{\mathcal{R}} = \zeta|_{Y_{\mathcal{R}} \times \Sigma}$, that is, $\zeta_{\mathcal{R}}$ is the transition function restricted to $Y_{\mathcal{R}} \times \Sigma$. Denote subautomaton by $\mathcal{R} \sqsubseteq \mathcal{A}$. We have $K = L(\mathcal{R})$. A supervisor $\mathcal{S}$ with $L(\mathcal{S}/\mathcal{A}) = K$ can be designed in the following steps.

Step 1: Replace all unobservable transitions in $\mathcal{R}$ by ε transitions:

$$\mathcal{R}_\varepsilon = (Y_{\mathcal{R}}, \Sigma_o, \zeta_{\mathcal{R}_\varepsilon}, y_o),$$

where $\zeta_{\mathcal{R}_\varepsilon} = \{(y, \sigma, y') \in \zeta_{\mathcal{R}} : \sigma \in \Sigma_o\} \cup \{(y, \varepsilon, y') \in \zeta_{\mathcal{R}} : \sigma \notin \Sigma_o\}$.

Step 2: Convert $\mathcal{R}_\varepsilon$ from a nondeterministic automaton to a deterministic automaton, called observer Cassandras and Lafortune (2009); Wonham and Cai (2019):

$$\mathcal{R}_{obs} = (Z, \Sigma_o, \xi, z_o) = AC(2^{Y_{\mathcal{R}}}, \Sigma_o, \xi, UR(y_o)),$$

where $AC(.)$ denotes the accessible part; $UR(.)$ denotes the unobservable reach, which is defined, for $z \subseteq Y_{\mathcal{R}}$, as

$$UR(z) = \{y \in Y_{\mathcal{R}} : (\exists y' \in z)y \in \zeta_{\mathcal{R}_\varepsilon}(y', \varepsilon)\}.$$

The transition function ξ is defined, for $z \in Z$ and $\sigma \in \Sigma_o$, as

$$\xi(z, \sigma) = UR(\{y \in Y_{\mathcal{R}} : (\exists y' \in z)y \in \zeta_{\mathcal{R}_\varepsilon}(y', \sigma)\}).$$

It is well-known that $L(\mathcal{R}_{obs}) = P(L(\mathcal{A}))$ Cassandras and Lafortune (2009); Wonham and Cai (2019).

Step 3: Define state feedback $\phi : Z \to 2^{\Sigma}$ as

$$\phi(z) = \{\sigma \in \Sigma : (\forall y \in z)\zeta(y, \sigma)! \Rightarrow \zeta(y, \sigma) \in Y_{\mathcal{R}}\}.$$

In other words, an event σ is enabled at state z if it does not take the system out of $Y_{\mathcal{R}}$ at any state $y \in z$. We can then design a supervisor $\mathcal{S}$ as follows: For $t \in P(L(\mathcal{A})) = L(\mathcal{R}_{obs})$,

$$\mathcal{S}(t) = \phi(\xi(z_o, t)).$$

It can be shown that the above designed supervisor satisfies $L(\mathcal{S}/\mathcal{A}) = K$ if K is controllable and observable Lin and Wonham (1988); Cassandras and Lafortune (2009).

If K is not controllable and/or observable, then we can find the smallest superlanguage of K that is controllable and observable. This superlanugae is unique. It is called *the infimal controllable and observable superlanguage* of K and denoted by $K^{\downarrow}$. In other words, $K^{\downarrow}$ is controllable and observable, $K \subseteq K^{\downarrow}$, and $K^{\downarrow}$ is the smallest language satisfying these conditions. After finding $K^{\downarrow}$, we can design a supervisor such that $L(\mathcal{S}/\mathcal{A}) = K^{\downarrow}$ Lin and Wonham (1988); Cassandras and Lafortune (2009).

3 Stochastic discrete event systems

Many man-made systems can be modeled as discrete event systems. Such a discrete event system often has some failure states. We investigate two important questions in this paper. (1) How long can the system run before a failure occurs, that is, what is the MTTF of the system? (2) Can we control the system so that its MTTF is maximized?

Intuitively, MTTF depends on lifetimes of events that are stochastic and paths to failure states described by the automaton $\mathcal{A}$. Formally, let us denote failure states by $Y_f \subseteq Y$. Y_f depends on the system under consideration and is given. To determine the corresponding failure states in the supervised system $\mathcal{S}/\mathcal{A}$, we note that $\mathcal{S}/\mathcal{A}$ is described by the following automaton.

$$\mathcal{G} = \mathcal{S}/\mathcal{A} = (Q, \Sigma, \delta, q_o) = AC(Y \times Z, \Sigma, \delta, (y_o, x_o)), \tag{8}$$

where the transition function δ is defined, for $q = (y, z) \in Q = AC(Y \times Z)$ and $\sigma \in \Sigma_o$, as

$$\delta(q, \sigma) = \begin{cases} (\zeta(y, \sigma), \xi(z, \sigma)) & \text{if } \zeta(y, \sigma)! \wedge \xi(z, \sigma)! \wedge \sigma \in \phi(z) \\ \text{undefined} & \text{otherwise} \end{cases}$$

and for $q = (y, z) \in Q$ and $\sigma \notin \Sigma_o$, as

$$\delta(q, \sigma) = \begin{cases} (\zeta(y, \sigma), z) & \text{if } \zeta(y, \sigma)! \wedge \sigma \in \phi(z) \\ \text{undefined} & \text{otherwise} \end{cases}$$

Note that we denote the states in the supervised system as $q \in Q$ to simplify the notations in the rest of the paper. A state $q = (y, z)$ is a failure state if $y \in Y_f$, that is,

$$Q_f = \{q \in Q : (\exists y \in Y_f) q = (y, z)\}.$$

Enumerate the states in Q as

$$Q = \{q_1, q_2, ..., q_n\}, \tag{9}$$

where n is the number of states in Q, denoted by $n = |Q|$. Without loss of generality, assume that there are $l = |Q_f|$ failure states and they are the last l states in the above enumeration, that is,

$$Q_f = \{q_{n-l+1}, q_{n-l+2}, ..., q_n\}. \tag{10}$$

Enumerate the event set as

$$\Sigma = \{\sigma_1, \sigma_2, ..., \sigma_m\}, \tag{11}$$

where m is the number of events: $m = |\Sigma|$. Suppose that the system is currently at state $q_i \in Q$. The set of all possible traces from q_i is given by the language

$$L(\mathcal{G}, q_i) = \{s : s \in \Sigma^* : \delta(q_i, s)!\}.$$

A string to failures, $s \in L(\mathcal{G}, q_i)$, is a string that ends at a failure state in Q_f and none of its prefixes visits any failure states in Q_f, that is,

$$s \in L(\mathcal{G}, q_i) \wedge \delta(q_i, s) \in Q_f \wedge (\forall s' < s)\delta(q_i, s') \notin Q_f,$$

where $s' < s$ means s' is a prefix of s and $s' \neq s$.

The set of all possible *strings to failures* is denoted by

$$L_f(\mathcal{G}, q_i) = \{s \in L(\mathcal{G}, q_i) : \delta(q_i, s) \in Q_f \wedge (\forall s' < s)\delta(q_i, s') \notin Q_f\}.$$

Note that $L_f(\mathcal{G}, q_i)$ is different than the marked language of $\mathcal{G}$ Wonham and Cai (2019); Cassandras and Lafortune (2009), because it requires the first visit to a failure state. Also, if the system is already in a failure state, $q_i \in Q_f$, then $L_f(\mathcal{G}, q_i) = \{\varepsilon\}$.

To investigate time to failure, we introduce event lifetimes and the associated clock structure as follows, which is same as described in Cassandras and Lafortune (2009). We say that an event $\sigma \in \Sigma$ is active after $s \in L(\mathcal{G})$ (or equivalently at state $q = \delta(q_o, s)$) if $s\sigma \in L(\mathcal{G})$ (or equivalently $\delta(q, \sigma)!$). If this is the case, σ becomes active when the last event in s occurs. We say that an event $\sigma \in \Sigma$ is activated if it becomes active. We say that an event $\sigma \in \Sigma$ is deactivated if it becomes inactive. The lifetime of an event is the time it takes for the event to occur after it is allowed to occur.

The *clock structure* is then denoted by

$$\Omega = \{(\omega_{\sigma_1}(i), \omega_{\sigma_2}(i), ..., \omega_{\sigma_m}(i)) : i = 1, 2, 3, ...\}, \tag{12}$$

where $\omega_{\sigma_k}(i)$ is the lifetime of the ith occurrence of event $\sigma_k \in \Sigma$.

For each event $\sigma \in \Sigma$, the lifetime $\omega_\sigma(i)$ is generated according to a given (stationary) probability distribution Ψ_σ, that is, for all $i = 1, 2, 3, ...$

$$Pr[\omega_\sigma(i) \leq x] = \Psi_\sigma(x). \tag{13}$$

In other words, $\Psi_\sigma(x)$ is the *cumulative distribution function* of the lifetime of σ. Denote the corresponding *probability density function* by $\psi_\sigma(x)$ and the *mean of the lifetime* of σ by μ_σ. Also, denote

$$\Psi = (\Psi_{\sigma_1}, \Psi_{\sigma_2}, ..., \Psi_{\sigma_m}). \tag{14}$$

We assume that the event lifetimes are independent of each other.

The *operating rules* of event occurrences in stochastic discrete event system $\mathcal{G}$ are as follows:

1. Each event is associated with a clock. When an event is first activated after its previous occurrence, its clock is initiated to be $\omega_\sigma(i), i = 1, 2, 3, ...$ The clock decreases at the rate of -1 (unitless), when it reaches 0, the event occurs. In other words, the clock describes the remaining lifetime of an event.
2. When an event occurs, the system may move to a new state. This causes some events to be activated and some other events to be deactivated.
3. When an event is deactivated, if its clock is not 0, it will be frozen until the event is re-activated.

Denote a realization of Ω based on Ψ by ω. For each realization ω, the stochastic discrete event system generates a string of events s. The string s depends on both ω and $\mathcal{G}$, as $\mathcal{G}$ specifies when events are activated and deactivated.

4 Time to failure

Consider a stochastic discrete event system defined by the triple

$$(\mathcal{G}, \Omega, \Psi). \tag{15}$$

Assume that the system is currently in state q_i and q_i is not a failure state, that is, $i \leq n - l$, (otherwise, the time to failure is 0). Denote the *time to failure* by

$$\pi_i = \text{the time to failure from state } q_i. \tag{16}$$

To find the mean time to failure, we consider the set of all possible strings to failures $L_f(\mathcal{G}, q_i)$ as defined in the previous section. For each string to failures $s \in L_f(\mathcal{G}, q_i)$, we denote

$$T(s) = \text{the time when the last event in } s \text{ occurs.}$$

The cumulative distribution function of time to failure is denoted by

$$H_i(x) = Pr[T(s) \leq x \mid s \in L_f(\mathcal{G}, q_i)].$$

The corresponding probability density function is denoted by $h_i(x)$.

Our objective is to find MTTF from state q_i, denoted by

$$\eta_i = E[\pi_i] = E[T(s) \mid s \in L_f(\mathcal{G}, q_i)] = \int_0^\infty x h_i(x) dx. \tag{17}$$

Before we demonstrate how to calculate η_i for general stochastic discrete event systems, let us first consider the following example.

Example 1 Consider the stochastic discrete event system shown in Fig. 1. The failure state is 3. There are three events: α, β, and γ. Assume that the system is at State 1 initially and the lifetime distributions are exponential, that is,

$$\Psi_\alpha(x) = 1 - e^{-x/\mu_\alpha}, \quad \Psi_\beta(x) = 1 - e^{-x/\mu_\beta}, \quad \Psi_\gamma(x) = 1 - e^{-x/\mu_\gamma},$$

where μ_α, μ_β, and μ_γ are mean lifetimes of α, β, and γ, respectively. As we will show in the next section, $\eta_1 = \mu_\alpha + \dfrac{\mu_\alpha^2}{\mu_\beta}$, $\eta_2 = 2\mu_\alpha + \dfrac{\mu_\alpha^2}{\mu_\beta}$.

For general stochastic discrete event systems with arbitrary lifetime distributions, it is not always possible to find analytic solutions for MTTF. To obtain an analytic solution, we must make some assumptions on the structures of stochastic discrete event systems and lifetime distributions, as to be investigated in the next section.

5 Calculation of mean time to failure

In order to calculate the MTTF analytically for a general stochastic discrete event system, we make the following assumptions. (1) The discrete event system $\mathcal{G}$ is strongly connected, that is, $(\forall q, q' \in Q)(\exists s \in \Sigma^*)\delta(q, s) = q'$. (2) The event lifetime distributions are independent and exponential, that is, $(\forall \sigma \in \Sigma)\Psi_\sigma(x) = 1 - e^{-x/\mu_\sigma}$. The first assumption is made without significant loss of generality, because if $\mathcal{G}$ is not strongly connected, we can investigate its strongly connected subsystems separately.

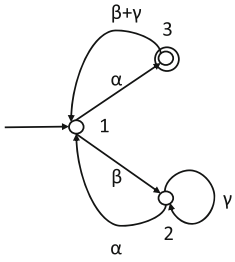

Fig. 1 Stochastic discrete event system $\mathcal{G}$ in Example 1. → denotes the initial state

Under the above assumptions, we show that the problem of finding MTTF in stochastic discrete event systems can be translated into the problem of finding first passage time in finite, irreducible, continuous time Markov chains, which has been investigated in the literature Darling and Siegert (1953); Brown and Chaganty (1983); Yao (1985).

We first convert a stochastic discrete event system $(\mathcal{G}, \Omega, \Psi)$ into a continuous time Markov chain as follows:

Step 1: Remove all self loops in $\mathcal{G}$, that is,

$$\mathcal{G}' = (Q, \Sigma, \delta', Q_f).$$

In $\mathcal{G}'$, δ' is obtained as follows: Remove from the set of all transitions in $\mathcal{G}$

$$\delta = \{(q, \sigma, q') : q, q' \in Q \wedge \sigma \in \Sigma \wedge \delta(q, \sigma) = q'\}$$

all the self loops to obtain

$$\delta' = \delta - \{(q, \sigma, q) : q \in Q \wedge \sigma \in \Sigma \wedge \delta(q, \sigma) = q\}.$$

Step 2: Identify all parallel transitions (PT) in $\mathcal{G}$, that is, for all $q, q' \in Q, q \neq q'$,

$$PT(q, q') = \{(q, \sigma, q') \in \delta' : \sigma \in \Sigma\}.$$

Step 3: The state space of the continuous time Markov chain is the same as that of $\mathcal{G}$, that is,

$$Q = \{q_1, q_2, ..., q_n\}.$$

Step 4: Define the transition rate matrix (also known as an intensity matrix or infinitesimal generator matrix) as

$$\Lambda = [\lambda_{ij}], \tag{18}$$

where

$$\begin{aligned} \lambda_{ij} &= \sum_{(q_i, \sigma, q_j) \in PT(q_i, q_j)} \frac{1}{\mu_\sigma}, \quad i, j = 1, 2, ..., n \wedge i \neq j \\ \lambda_{ii} &= -\sum_{i \neq j} \lambda_{ij}, \quad i = 1, 2, ..., n. \end{aligned} \tag{19}$$

For $q_i \in Q$, denote the time that the system stays in q_i, called sojourn time, by ρ_i. For the continuous time Markov Chain (Q, Λ), the mean sojourn time is given by

$$E[\rho_i] = -\frac{1}{\lambda_{ii}} = \frac{1}{\sum_{i \neq j} \lambda_{ij}}. \tag{20}$$

Theorem 1 *For the stochastic discrete event system $(\mathcal{G}, \Omega, \Psi)$ and the continuous time Markov chain (Q, Λ) defined above, the sojourn time ρ_i in any state $q_i \in Q$ is the same for the stochastic discrete event system and the continuous time Markov chain.*

Proof Since self loops do not change the state of the system, removing self loops does not change the sojourn time at any state.

Because event lifetimes are independent and exponential, parallel transitions, say α and β, can be combined, the lifetime of the combined transition is also exponential, with a mean lifetime of $\frac{\mu_\alpha \mu_\beta}{\mu_\alpha + \mu_\beta}$.

Since the exponential distribution is memoryless, re-activating the clock in Operating Rule 4 is same as starting a new clock.

Therefore, the stochastic discrete event system $(\mathcal{G}, \Omega, \Psi)$ and the continuous time Markov chain (Q, Λ) has the same sojourn time ρ_i for any state $q_i \in Q$.

□

Note that although the sojourn times are the same for the discrete event system and the continuous time Markov chain, the conversion from the discrete event system $(\mathcal{G}, \Omega, \Psi)$ to the continuous time Markov chain (Q, Λ) removes some important information on the discrete event system. For example, self-loops and parallel transitions are important in modeling and control of discrete event systems Cassandras and Lafortune (2009); Wonham etal (2018). In other words, the conversion is for calculating MTTF only. The discrete event system $(\mathcal{G}, \Omega, \Psi)$ with controllable and observable events is needed for modeling and control.

The problem of finding the first passage time of a finite, irreducible (that is, strongly connected), continuous time Markov chain has been investigated in the literature Darling and Siegert (1953); Brown and Chaganty (1983); Yao (1985). We review and summarize the results as follows.

For $q_i \notin Q_f$, that is, $i = 1, 2, ..., n-l$, denote the Laplace transform of $h_i(x)$ by $LP_i(s)$. In other words,

$$LP_i(s) = \int_0^\infty e^{-xs} h_i(x) dx = E[e^{-\pi_i s}]. \tag{21}$$

The system will stay in q_i for ρ_i time and then move to state q_j with probability $-\frac{\lambda_{ij}}{\lambda_{ii}} = \frac{\lambda_{ij}}{\sum_{i \neq j} \lambda_{ij}}$. Hence,

$$E[e^{-\pi_i s}] = -\sum_{j=1, j\neq i}^{n} \frac{\lambda_{ij}}{\lambda_{ii}} E[e^{-(\rho_i + \pi_j)s}] = -\sum_{j=1, j\neq i}^{n} \frac{\lambda_{ij}}{\lambda_{ii}} E[e^{-\rho_i s}] E[e^{-\pi_j s}].$$

Since for $j = 1, ..., n-l$, $E[e^{-\pi_j s}] = LP_j(s)$, and for $j = n-l+1, ..., n$, $E[e^{-\pi_j s}] = 1$,

$$LP_i(s) = -\sum_{j=1, j\neq i}^{n-l} \frac{\lambda_{ij}}{\lambda_{ii}} E[e^{-\rho_i s}] LP_j(s) - \sum_{j=n-l+1}^{n} \frac{\lambda_{ij}}{\lambda_{ii}} E[e^{-\rho_i s}]. \tag{22}$$

Taking derivative with respect to s on both sides, we have

$$\begin{aligned} \frac{dLP_i(s)}{ds} = &-\sum_{j=1, j\neq i}^{n-l} \frac{\lambda_{ij}}{\lambda_{ii}} E[-\rho_i e^{-\rho_i s}] LP_j(s) - \sum_{j=1, j\neq i}^{n-l} \frac{\lambda_{ij}}{\lambda_{ii}} E[e^{-\rho_i s}] \frac{dLP_j(s)}{ds} \\ &- \sum_{j=n-l+1}^{n} \frac{\lambda_{ij}}{\lambda_{ii}} E[-\rho_i e^{-\rho_i s}]. \end{aligned} \tag{23}$$

Since

$$\lim_{s\to 0}\frac{dLP_i(s)}{ds}=\lim_{s\to 0}\int_0^\infty(-x)e^{-xs}h_i(x)dx=-\int_0^\infty xh_i(x)dx=-E[\pi_i]=-\eta_i$$
$$\lim_{s\to 0}LP_j(s)=\lim_{s\to 0}\int_0^\infty e^{-xs}h_i(x)dx=\int_0^\infty h_i(x)dx=1$$
$$\lim_{s\to 0}E[-\rho_i e^{-\rho_i s}]=E[-\rho_i]=\frac{1}{\lambda_{ii}}$$
$$\lim_{s\to 0}E[e^{-\rho_i s}]=1,$$

we have, by letting $s\to 0$,

$$\eta_i=\sum_{j=1,j\neq i}^{n-l}\frac{\lambda_{ij}}{\lambda_{ii}}\frac{1}{\lambda_{ii}}+\sum_{j=1,j\neq i}^{n-l}\frac{\lambda_{ij}}{\lambda_{ii}}(-\eta_j)+\sum_{j=n-l+1}^{n}\frac{\lambda_{ij}}{\lambda_{ii}}\frac{1}{\lambda_{ii}},$$

which is equivalent to

$$\lambda_{ii}\eta_i=\sum_{j=1,j\neq i}^{n-l}\frac{\lambda_{ij}}{\lambda_{ii}}+\sum_{j=1,j\neq i}^{n-l}\lambda_{ij}(-\eta_j)+\sum_{j=n-l+1}^{n}\frac{\lambda_{ij}}{\lambda_{ii}}.$$

Hence,

$$\lambda_{ii}\eta_i+\sum_{j=1,j\neq i}^{n-l}\lambda_{ij}\eta_j=\sum_{j=1,j\neq i}^{n}\frac{\lambda_{ij}}{\lambda_{ii}}=-1.$$

Write the above equations in the matrix form

$$\mathbf{A}\begin{bmatrix}\eta_1\\ \eta_2\\ \cdots\\ \eta_{n-l}\end{bmatrix}=-\begin{bmatrix}1\\ 1\\ \cdots\\ 1\end{bmatrix},\tag{24}$$

where

$$\mathbf{A}=\begin{bmatrix}\lambda_{11} & \lambda_{12} & \cdots & \lambda_{1(n-l)}\\ \lambda_{21} & \lambda_{22} & \cdots & \lambda_{2(n-l)}\\ \cdots & & \cdots & \\ \lambda_{(n-l)1} & \lambda_{(n-l)2} & \cdots & \lambda_{(n-l)(n-l)}\end{bmatrix}.\tag{25}$$

Note that $\mathbf{A}$ is the principal submatrix of Λ with dimension $(n-l)\times(n-l)$. The following theorem presents the results on MTTF.

Theorem 2 *Consider the stochastic discrete event system $(\mathcal{G},\Omega,\Psi)$. MTTF is given by*

$$\begin{bmatrix}E[\pi_1]\\ E[\pi_2]\\ \cdots\\ E[\pi_{n-1}]\end{bmatrix}=\begin{bmatrix}\eta_1\\ \eta_2\\ \cdots\\ \eta_{n-l}\end{bmatrix}=-\boldsymbol{A}^{-1}\begin{bmatrix}1\\ 1\\ \cdots\\ 1\end{bmatrix}.\tag{26}$$

Proof By Theorem 1 and the derivation above.

Using similar approaches, we can prove the following theorem that can be used to calculate the standard deviation of time to failure.

Theorem 3 *Consider the stochastic discrete event system* $(\mathcal{G}, \Omega, \Psi)$. *The standard deviation of time to failure is given by, for* $i = 1, 2, ..., n - l$,

$$\upsilon_i = \sqrt{Var(\pi_i)} = \sqrt{E[\pi_i^2] - (E[\pi_i])^2}, \tag{27}$$

where

$$\begin{bmatrix} E[\pi_1^2] \\ E[\pi_2^2] \\ \cdots \\ E[\pi_{n-1}^2] \end{bmatrix} = 2\boldsymbol{A}^{-2} \begin{bmatrix} 1 \\ 1 \\ \cdots \\ 1 \end{bmatrix}. \tag{28}$$

Proof By Theorem 1 and taking the second derivative of $LP_i(s)$. See Appendix A for details.□

Example 2 Consider the stochastic discrete event system of Example 1 shown in Fig. 1. We first convert the stochastic discrete event system into a continuous time Markov chain as follows.

Step 1: Remove all self loops in $\mathcal{G}$.

Step 2: Identify all parallel transitions (PT) in $\mathcal{G}$,

$$\begin{array}{ll} PT(1,2) = \{(1,\beta,2)\} & PT(1,3) = \{(1,\alpha,3)\} \\ PT(2,1) = \{(2,\alpha,1)\} & PT(2,3) = \emptyset \\ PT(3,1) = \{(3,\beta,1),(3,\gamma,1)\} & PT(3,2) = \emptyset \end{array}$$

Step 3: The state space of the continuous time Markov chain is the same as that of $\mathcal{G}$, that is,

$$Q = \{q_1, q_2, q_3\}.$$

Step 4: Define the transition rate matrix as

$$\Lambda = \begin{bmatrix} -\frac{1}{\mu_\alpha} - \frac{1}{\mu_\beta} & \frac{1}{\mu_\beta} & \frac{1}{\mu_\alpha} \\ \frac{1}{\mu_\alpha} & -\frac{1}{\mu_\alpha} & 0 \\ \frac{1}{\mu_\beta} + \frac{1}{\mu_\gamma} & 0 & -\frac{1}{\mu_\beta} - \frac{1}{\mu_\gamma} \end{bmatrix}.$$

Since there is only one failure state, $l = 1$ and $n - l = 2$. Hence

$$\mathbf{A} = \begin{bmatrix} -\frac{1}{\mu_\alpha} - \frac{1}{\mu_\beta} & \frac{1}{\mu_\beta} \\ \frac{1}{\mu_\alpha} & -\frac{1}{\mu_\alpha} \end{bmatrix}.$$

MTTF is then given by

$$\begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} E[\pi_1] \\ E[\pi_2] \end{bmatrix} = -\mathbf{A}^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \mu_\alpha + \frac{\mu_\alpha^2}{\mu_\beta} \\ 2\mu_\alpha + \frac{\mu_\alpha^2}{\mu_\beta} \end{bmatrix}.$$

To calculate the standard deviation of time to failure, we have

$$\begin{bmatrix} E[\pi_1^2] \\ E[\pi_2^2] \end{bmatrix} = 2\mathbf{A}^{-2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2\mu_\alpha^2 + 6\frac{\mu_\alpha^3}{\mu_\beta} + 2\frac{\mu_\alpha^4}{\mu_\beta^2} \\ 6\mu_\alpha^2 + 8\frac{\mu_\alpha^3}{\mu_\beta} + 2\frac{\mu_\alpha^4}{\mu_\beta^2} \end{bmatrix}.$$

Hence,

$$\begin{bmatrix} \upsilon_1 \\ \upsilon_2 \end{bmatrix} = \begin{bmatrix} \sqrt{E[\pi_i^2] - (E[\pi_i])^2} \\ \sqrt{E[\pi_i^2] - (E[\pi_i])^2} \end{bmatrix} = \begin{bmatrix} \mu_\alpha \sqrt{1 + 4\frac{\mu_\alpha}{\mu_\beta} + (\frac{\mu_\alpha}{\mu_\beta})^2} \\ \mu_\alpha \sqrt{2 + 4\frac{\mu_\alpha}{\mu_\beta} + (\frac{\mu_\alpha}{\mu_\beta})^2} \end{bmatrix}.$$

In particular, if $\mu_\beta \to \infty$,

$$\begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} \to \begin{bmatrix} \mu_\alpha \\ 2\mu_\alpha \end{bmatrix}, \begin{bmatrix} \upsilon_1 \\ \upsilon_2 \end{bmatrix} \to \begin{bmatrix} \mu_\alpha \\ \sqrt{2}\mu_\alpha \end{bmatrix}.$$

If $\mu_\alpha \to \infty$,

$$\begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} \to \begin{bmatrix} \infty \\ \infty \end{bmatrix}, \begin{bmatrix} \upsilon_1 \\ \upsilon_2 \end{bmatrix} \to \begin{bmatrix} \infty \\ \infty \end{bmatrix}.$$

If $\mu_\beta = \mu_\alpha$,

$$\begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} 2\mu_\alpha \\ 3\mu_\alpha \end{bmatrix}, \begin{bmatrix} \upsilon_1 \\ \upsilon_2 \end{bmatrix} = \begin{bmatrix} \sqrt{6}\mu_\alpha \\ \sqrt{7}\mu_\alpha \end{bmatrix}.$$

6 Supervisory control to maximize mean time to failure

In this section, we use supervisory control to maximize MTTF. The idea is that while some states are necessary to perform required tasks, the remaining states are optional. For those optional states, we determine one by one whether it shall be removed or added by supervisory control, depending on whether it increases or decreases MTTF.

Formally, let us denote the set of required states by $Y_r \subseteq Y$. Required states are states necessary for the system to perform its tasks and hence any supervisor must allow the system to visit these states Lin and Wonham (1988); Cassandras and Lafortune (2009). The subautomaton with the required states is denoted by

$$\mathcal{A}_r = (Y_r, \Sigma, \zeta_r, y_o),$$

where $\zeta_r = \zeta|_{Y_r \times \Sigma}$. The language generated by $\mathcal{A}_r$, $K_r = L(\mathcal{A}_r)$, is called the required language. The required states Y_r are fixed and determined by the problem to be solved.

To maximize MTTF, we design a supervisor that allows states in $Y - Y_r$ or not, depending on whether allowing them will increase MTTF or not. We check the states in $Y - Y_r$ one by one in a given order $Y - Y_r = \{y_1, y_2, ..., y_k\}$ to see if it shall be allowed or not using Algorithm 1 below.

Since the optimal supervisor $\mathcal{S}_k$ obtained by Algorithm 1 depends on the order $Y - Y_r = \{y_1, y_2, ..., y_k\}$, theoretically, we need to consider all possible orders, which is a big job. However, we can avoid this exhaustive search if we first order the states in $Y - Y_r$ by measuring how a state $y \in Y - Y_r$ is prone to failure using Algorithm 2.

Since the optimal supervisor $\mathcal{S}_k$ obtained by Algorithm 1 depends on the order $Y - Y_r = \{y_1, y_2, ..., y_k\}$, theoretically, we need to consider all possible orders, which is a big job. However, we can avoid this exhaustive search if we first order the states in $Y - Y_r$ by measuring how a state $y \in Y - Y_r$ is prone to failure using Algorithm 2.

Since $\theta(y_i)$ is the MTTF from y_i in $\mathcal{A}(y)$, if $\theta(y_i) > \theta(y_j)$, then it takes more time for the system to fail from y_i than from y_j, that is, y_i is less prone to failure than y_j. Hence, $\theta(y_i)$ can be used as a measure on how y_i is prone to failure. After obtaining $\Theta = \{\theta(y) : y \in Y - Y_r\}$ using Algorithm 2, we order the states in $Y - Y_r$ as $Y - Y_r = \{y_1^o, y_2^o, ..., y_k^o\}$ such that

Algorithm 1 Find optimal supervisor for $Y - Y_r = \{y_1, y_2, ..., y_k\}$.

Input: $\mathcal{A} = (Y, \Sigma, \zeta, y_o)$, Y_f, $Y - Y_r = \{y_1, y_2, ..., y_k\}$
Output: Optimal supervisor $\mathcal{S}^o$
1: $Y_0 = Y_r$;
2: $\mathcal{A}_0 = (Y_0, \Sigma, \zeta_0, y_o)$, where $\zeta_0 = \zeta|_{Y_0 \times \Sigma}$;
3: for $i = 1, 2, ...k$ do begin
4: $Y_i = Y_{i-1} \cup \{y_i\}$;
5: $\mathcal{A}_i = (Y_i, \Sigma, \zeta_i, y_o)$, where $\zeta_i = \zeta|_{Y_i \times \Sigma}$;
6: $K_i = L(\mathcal{A}_i)$;
7: calculate $K_i^{\downarrow}$;
8: design a supervisor $\mathcal{S}_i$ such that $L(\mathcal{S}_i/\mathcal{A}) = K_i^{\downarrow}$;
9: calculate MTTF $\eta_{o,i}$ of $\mathcal{S}_i/\mathcal{A}$ from the initial state q_o;
10: if $\eta_{o,i} < \eta_{o,i-1}$, then
11: $Y_i = Y_{i-1}$;
12: $\mathcal{S}_i = \mathcal{S}_{i-1}$;
13: end (for loop)
14: Optimal supervisor $\mathcal{S}^o = \mathcal{S}_k$;
15: End.

Algorithm 2 Find prone to failure measures.

Input: $\mathcal{A} = (Y, \Sigma, \zeta, y_o)$, Y_f, Y_r
Output: Prone to failure measures Θ
1: for $y \in Y - Y_r$ do begin
2: identify all direct paths from y to Y_f (paths without loops);
3: remove all states and transitions in $Y - Y_f$, except those in the direct paths identified above. Denote the resulting automaton as $\mathcal{A}(y)$;
4: calculate MTTF from y using automaton $\mathcal{A}(y)$. Denote the result as $\theta(y)$;
5: end (for loop)
6: Prone to failure measures $\Theta = \{\theta(y) : y \in Y - Y_r\}$;
7: End.

$\theta(y_i^o) \geq \theta(y_{i+1}^o)$, $i = 1, 2, ..., k-1$. We then use Algorithm 1 with $Y - Y_r = \{y_1^o, y_2^o, ..., y_k^o\}$ to obtain the optimal supervisor $\mathcal{S}^o$ that maximizes MTTF as shown in the following theorem.

Theorem 4 *Order the elements in $Y - Y_r$ as $Y - Y_r = \{y_1^o, y_2^o, ..., y_k^o\}$ such that $\theta(y_i^o) \geq \theta(y_{i+1}^o)$, $i = 1, 2, ..., k-1$ based on the outputs of Algorithm 2. Let $Y - Y_r$ be the input to Algorithm 1 and the optimal supervisor outputted by Algorithm 1 be $\mathcal{S}^o$. For a state-based supervisor $\mathcal{S}$, denote the states visited by $\mathcal{S}/\mathcal{A}$ from the initial state q_o by $Y_{\mathcal{S}}$ and MTTF of $\mathcal{S}/\mathcal{A}$ from the initial state q_o by $MTTF(\mathcal{S}/\mathcal{A})$. Assume that, for $y_i^o \in Y_{\mathcal{S}} - Y_{\mathcal{S}^o}$, $MTTF(\mathcal{S}/\mathcal{A}) > MTTF(\mathcal{S}^o/\mathcal{A})$ implies adding y_i^o to Y_r increases MTTF. Then the optimal supervisor $\mathcal{S}^o$ maximizes MTTF, that is,*

$$(\forall \mathcal{S})(L(\mathcal{S}/\mathcal{A}) \supseteq K_r \Rightarrow MTTF(\mathcal{S}/\mathcal{A}) \leq MTTF(\mathcal{S}^o/\mathcal{A})).$$

Proof We prove Theorem 4 by contradiction. If

$$(\forall \mathcal{S})(L(\mathcal{S}/\mathcal{A}) \supseteq K_r \Rightarrow MTTF(\mathcal{S}/\mathcal{A}) \leq MTTF(\mathcal{S}^o/\mathcal{A})),$$

is not true, then

$$\begin{aligned} & \neg(\forall \mathcal{S})(L(\mathcal{S}/\mathcal{A}) \supseteq K_r \Rightarrow MTTF(\mathcal{S}/\mathcal{A}) \leq MTTF(\mathcal{S}^o/\mathcal{A})) \\ \Leftrightarrow \quad & (\exists \mathcal{S})(L(\mathcal{S}/\mathcal{A}) \supseteq K_r \wedge MTTF(\mathcal{S}/\mathcal{A}) > MTTF(\mathcal{S}^o/\mathcal{A})) \\ \Leftrightarrow \quad & (\exists \mathcal{S})(L(\mathcal{S}/\mathcal{A}) \supseteq K_r^{\downarrow} \wedge MTTF(\mathcal{S}/\mathcal{A}) > MTTF(\mathcal{S}^o/\mathcal{A})) \\ & \text{(by the definition of } K_r^{\downarrow}) \end{aligned}$$

Since $L(\mathcal{S}/\mathcal{A}) \supseteq K_r$, we have $Y_{\mathcal{S}} \supseteq Y_r$. From Algorithm 1, we have $Y_{\mathcal{S}}^o \supseteq Y_r$. Because $MTTF(\mathcal{S}/\mathcal{A}) > MTTF\ (\mathcal{S}^o/\mathcal{A})$, we have $Y_{\mathcal{S}} \neq Y_{\mathcal{S}}^o$.

Let y_i^o be the first element in $\{y_1^o, y_2^o, ..., y_k^o\}$ such that $y_i^o \in Y_{\mathcal{S}} - Y_{\mathcal{S}^o}$. Since $MTTF(\mathcal{S}/\mathcal{A}) > MTTF(\mathcal{S}^o/\mathcal{A})$, adding y_i^o to Y_r increases MTTF. However, by Algorithm 1, $y_i^o \notin Y_{\mathcal{S}^o}$ implies adding y_i^o to Y_r decreases MTTF, a contradiction. □

The computational complexity of synthesizing the optimal supervisor $\mathcal{S}_k^o$ can be analyzed as follows. (1) The computational complexity of finding MTTF is of the order $O(|Q|^2)$. (2) The computational complexity of finding $K^{\downarrow}$ and design a supervisor is of the order $O(|Y|)$ for full observation (all event are observable) and $O(2^{|Y|})$ for partial observation (not all event are observable). (3) The computational complexity of Algorithm 1 is of the order $O(|Y|(|Y| + |Q|^2))$ for full observation and $O(|Y|(2^{|Y|} + |Q|^2)$ for partial observation. (4) The computational complexity of Algorithm 2 is of the order $O(|Y|^3)$.

7 Applications to management of transformers in power systems

In this section, we apply the theoretical results of the previous sections to power system failure analysis. We calculate and maximize MTTF of equipment in power systems. Our approach can be used for any equipment in a cyber-physical power system (and indeed in any industrial system). We use the model and data of a 220 kV transformer discussed in Zhong etal (2016) as an example mainly because the mean lifetimes of its events are readily available from Zhong etal (2016).

The transformer is modeled as a discrete event system $\mathcal{A} = (Y, \Sigma, \zeta, y_o)$ shown in Fig. 2. The definitions of states in $Y = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ are given in Table 1. The events are $\Sigma = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \beta_1, \beta_2, \beta_3, \beta_4, \gamma_1, \gamma_2, \gamma_3, \gamma_4\}$, where α_i are uncontrollable events representing the natural progression of the transformer; β_i are controllable events representing the actions of minor overhaul, major overhaul, repair, and replacement, respectively; and γ_i are uncontrollable events representing the completion of minor overhaul, major

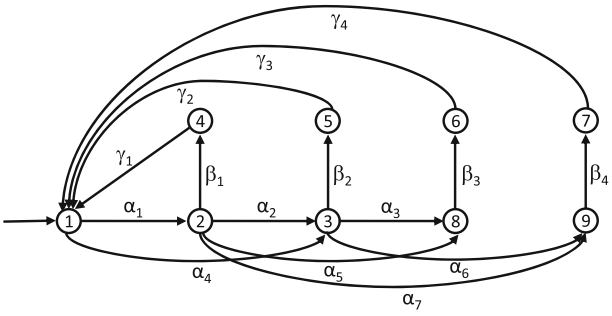

Fig. 2 A discrete event system model of a transformer

Table 1 Definitions of states of the transformer

State	Definition
1	Normal
2	Needing minor overhaul
3	Needing major overhaul
4	Minor overhaul
5	Major overhaul
6	Repair
7	Replacement
8	Repairable failure
9	Aging failure

overhaul, repair, and replacement, respectively. We assume that all events are observable, that is, $\Sigma_o = \Sigma$.

The mean lifetimes of events μ_σ and their reciprocals $1/\mu_\sigma$ for $\sigma \in \Sigma$ are given in Table 2 (see Zhong etal (2016)).

Let the required state be $Y_r = \{1, 2, 3, 8, 9\}$ and the failure states be $Y_f = \{8, 9\}$. Hence, $\mathcal{A}_r = (Y_r, \Sigma, \zeta_r, y_o)$ is shown in Fig. 3.

Since $\Sigma_c = \{\beta_1, \beta_2, \beta_3, \beta_4\}$ and $\Sigma_o = \Sigma$, it can be shown Lin and Wonham (1988) that $K_r = L(\mathcal{A}_r)$ is controllable and observable. Hence, a supervisor $\mathcal{S}_r$ exists such that $L(\mathcal{S}_r/\mathcal{A}_r) = K_r$. In fact, the supervised system $\mathcal{G}_r = \mathcal{S}_r/\mathcal{A}_r$ is isomorphic to $\mathcal{A}_r$, that is, there is a one-to-one mapping between states and transitions of $\mathcal{G}_r$ and $\mathcal{A}_r$.

Table 2 Mean lifetimes of events and their reciprocals

Event σ	Mean lifetime μ_σ (year)	$1/\mu_\sigma$ (times/year)
α_1	1.22	0.82
α_2	3.23	0.31
α_3	19.61	0.051
α_4	2.44	0.41
α_5	20.83	0.048
α_6	21.23	0.0471
α_7	21.23	0.0471
β_1	0.083	12.04
β_2	0.49	2.0
β_3	0.0219	45.62
β_4	0.0323	30.41
γ_1	0.0285	35.1
γ_2	0.0555	18.02
γ_3	0.0823	12.15
γ_4	0.1242	8.05

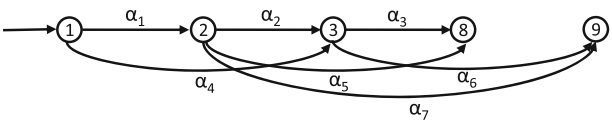

Fig. 3 $\mathcal{A}_r = (Y_r, \Sigma, \zeta_r, y_o)$ of the transformer with $\eta_1 = 11.0570$

Recalling from the previous sections, we can calculate MTTF in $\mathcal{A}_r$. For $Y_r = \{1, 2, 3, 8, 9\}$, using the mean lifetimes of events in Table 2, we have

$$\Lambda = \begin{bmatrix} -1.23 & 0.82 & 0.41 & 0 & 0 \\ 0 & -0.4051 & 0.31 & 0.048 & 0.0471 \\ 0 & 0 & -0.0981 & 0.051 & 0.0471 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Hence

$$\mathbf{A} = \begin{bmatrix} -1.23 & 0.82 & 0.41 \\ 0 & -0.4051 & 0.31 \\ 0 & 0 & -0.0981 \end{bmatrix}.$$

$$\begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} = -\mathbf{A}^{-1} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 11.0570 \\ 10.2692 \\ 10.1937 \end{bmatrix}.$$

In particular, MTTF from the initial state is $\eta_1 = 11.0570$.

Next we investigate if MTTF can be improved by adding more states to $\mathcal{A}_r$. Using Algorithm 2, we construct $\mathcal{A}(y)$ for $y = 4, 5, 6, 7$. $\mathcal{A}(4)$ is shown in Fig. 4 $\mathcal{A}(5)$, $\mathcal{A}(6)$, and $\mathcal{A}(7)$ are similar.

The measures of prone to failure of state y, $\theta(y)$, are then calculated as follows.

$$\theta(4) = 11.0855, \quad \theta(5) = 11.1125$$
$$\theta(6) = 11.1393, \quad \theta(7) = 11.1812.$$

Since $\theta(7) > \theta(6) > \theta(5) > \theta(4)$, let us add state $y = 7, 6, 5, 4$ to $\mathcal{A}_r$ one by one in that order. Adding $y = 7$ results in $\mathcal{A}_7$, which is shown in Fig. 5. It can be shown that $K_7 = L(\mathcal{A}_7)$ is controllable and observable. Hence, $K_7^{\downarrow} = K_7$ and a supervisor $\mathcal{S}_7$ exists such that $L(\mathcal{S}_7/\mathcal{A}_7) = K_7$. The supervised system $\mathcal{G}_7 = \mathcal{S}_7/\mathcal{A}_7$ is isomorphic to $\mathcal{A}_7$.

We calculate MTTF in $\mathcal{A}_7$ in a way similar to those in $\mathcal{A}_r$ and obtain $\eta_1 = 11.0570$. Hence, adding $y = 7$ does not improve MTTF. Therefore, we do not add $y = 7$. By a similar calculation, adding $y = 6$ does not improve MTTF, and we do not add $y = 6$.

Adding $y = 5$ results in $\mathcal{A}_5$, which is shown in Fig. 6. $K_5 = L(\mathcal{A}_5)$ is controllable and observable. Hence, $K_5^{\downarrow} = K_5$. We calculate MTTF in $\mathcal{A}_5$ and obtain $\eta_1 = 15.1092$. Clearly, performing a major overhaul improves MTTF, as MTTF is increased by 37%.

Adding $y = 4$ results in $\mathcal{A}_4$, which is shown in Fig. 7. $K_4 = L(\mathcal{A}_4)$ is also controllable and observable. Hence, $K_4^{\downarrow} = K_4$. We calculate MTTF in $\mathcal{A}_5$ and obtain $\eta_1 = 49.8456$. Clearly, performing a minor overhaul in addition to a major overhaul significantly improves MTTF, as MTTF is increased by 351%.

The supervised system under optimal control is given in Fig. 7

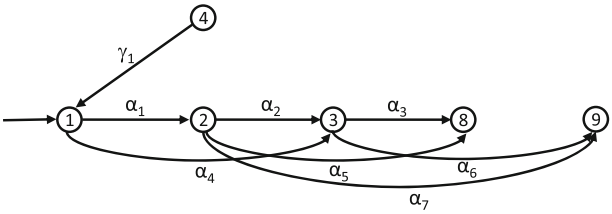

Fig. 4 $\mathcal{A}(4)$ of the transformer

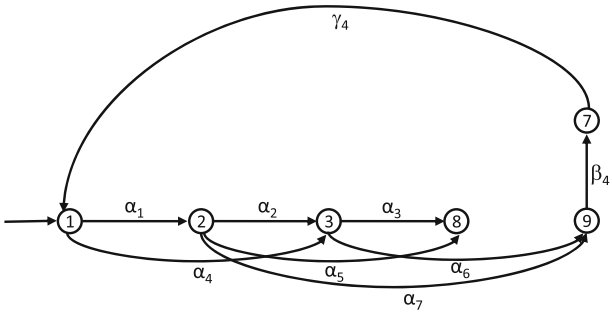

Fig. 5 $\mathcal{A}_7$ of the transformer with $\eta_1 = 11.0570$

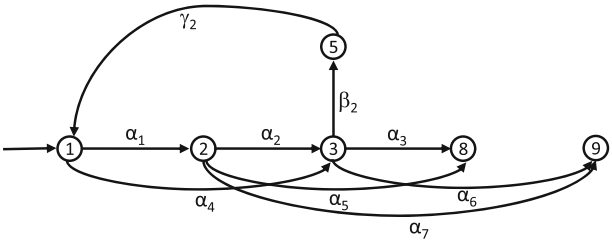

Fig. 6 $\mathcal{A}_5$ of the transformer with $\eta_1 = 15.1092$

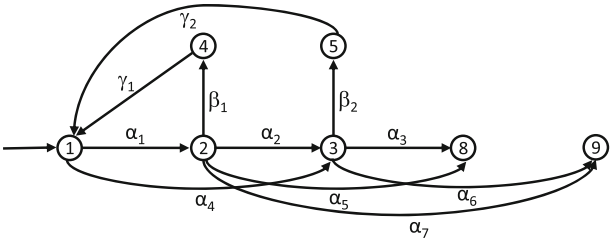

Fig. 7 $\mathcal{A}_4$ of the transformer with $\eta_1 = 49.8456$

8 Conclusion

The results in this paper are novel because MTTF has not been investigated in discrete event systems before. To the best of our knowledge, this paper is the first to provide a formal definition of MTTF in a discrete event system. The method proposed to calculate MTTF is also new and easy to use. It involves the calculation of the inverses of intensity matrices. Using a supervisor to maximize MTTF has not been attempted before. It allows the system to avoid prone-to-failure states while performing tasks that can prolong MTTF. Such a supervisor can be designed systematically using the proposed method. The benefits of the approach are illustrated by the application to the operation and maintenance of transformers in power systems. In the future work, we plan to apply the developed theory to more engineering systems. Since the computational complexity of calculating MTTF is polynomial with respect to the number of states, we expect that our method can be applied to complex systems.

Acknowledgements We would like to thank Dr. Robert Brandt for pointing us to the results on Markov chains and other inspiring discussions

Compliance with ethical standards

Competing Interest The authors have no competing interests to declare that are relevant to the content of this article.

Appendix: Proof of Theorem 3

Start from the derivative of $LP_i(s)$ in Eq. (23), let us take the second derivative of $LP_i(s)$:

$$\frac{d^2LP_i(s)}{ds^2} = -\sum_{j\neq i, j=1}^{n-l} \frac{\lambda_{ij}}{\lambda_{ii}} E[(-\rho_i)^2 e^{-\rho_i s}] LP_j(s) - 2\sum_{j\neq i, j=1}^{n-l} \frac{\lambda_{ij}}{\lambda_{ii}} E[-\rho_i e^{-\rho_i s}] \frac{dLP_i(s)}{ds}$$
$$- \sum_{j\neq i, j=1}^{n-l} \frac{\lambda_{ij}}{\lambda_{ii}} E[e^{-\rho_i s}] \frac{d^2LP_j(s)}{ds^2} - \sum_{j=n-l+1}^{n} \frac{\lambda_{ij}}{\lambda_{ii}} E[(-\rho_i)^2 e^{-\rho_i s}].$$

Since

$$\lim_{s\to 0} \frac{d^2LP_i(s)}{ds^2} = \lim_{s\to 0} \int_0^\infty (-x)^2 e^{-xs} h_i(x)dx = \int_0^\infty x^2 h_i(x)dx = E[(\pi_i)^2]$$
$$\lim_{s\to 0} \frac{dLP_i(s)}{ds} = \lim_{s\to 0} \int_0^\infty (-x) e^{-xs} h_i(x)dx = -\int_0^\infty x h_i(x)dx = -E[\pi_i] = -\eta_i$$
$$\lim_{s\to 0} LP_j(s) = \lim_{s\to 0} \int_0^\infty e^{-xs} h_i(x)dx = \int_0^\infty h_i(x)dx = 1$$
$$\lim_{s\to 0} E[(-\rho_i)^2 e^{-\rho_i s}] = E[(\rho_i)^2] = 2(\frac{1}{\lambda_{ii}})^2$$
$$\lim_{s\to 0} E[-\rho_i e^{-\rho_i s}] = \frac{1}{\lambda_{ii}}$$
$$\lim_{s\to 0} E[e^{-\rho_i s}] = 1,$$

we have, by letting $s \to 0$,

$$E[(\pi_i)^2] = -\sum_{j\neq i, j=1}^{n-l} \frac{\lambda_{ij}}{\lambda_{ii}} 2(\frac{1}{\lambda_{ii}})^2 - 2\sum_{j\neq i, j=1}^{n-l} \frac{\lambda_{ij}}{\lambda_{ii}} \frac{1}{\lambda_{ii}}(-\eta_i) - \sum_{j\neq i, j=1}^{n-l} \frac{\lambda_{ij}}{\lambda_{ii}} E[(\pi_j)^2] - \sum_{j=n-l+1}^{n} \frac{\lambda_{ij}}{\lambda_{ii}} 2(\frac{1}{\lambda_{ii}})^2.$$

In other words,

$$\begin{aligned}
\sum_{j=1}^{n-l} \lambda_{ij} E\left[(\pi_j)^2\right] &= \lambda_{ii} E\left[(\pi_i)^2\right] + \sum_{j\neq i, j=1}^{n-l} \lambda_{ij} E\left[(\pi_j)^2\right] \\
&= -2\sum_{j\neq i, j=1}^{n-l} \frac{\lambda_{ij}}{\lambda_{ii}^2} - 2\sum_{j\neq i, j=1}^{n-l} \frac{\lambda_{ij}}{\lambda_{ii}}(-\eta_i) - 2\sum_{j=n-l+1}^{n} \frac{\lambda_{ij}}{\lambda_{ii}^2} \\
&= -2\sum_{j\neq i, j=1}^{n} \frac{\lambda_{ij}}{\lambda_{ii}^2} + 2\sum_{j\neq i, j=1}^{n-l} \frac{\lambda_{ij}}{\lambda_{ii}} \eta_i \\
&\quad \text{(by Eq. (19))} \\
&= 2\frac{1}{\lambda_{ii}} + 2\sum_{j\neq i, j=1}^{n-l} \frac{\lambda_{ij}}{\lambda_{ii}} \eta_i \\
&= 2\frac{1}{\lambda_{ii}} - 2\eta_i + 2\sum_{j=1}^{n-l} \frac{\lambda_{ij}}{\lambda_{ii}} \eta_i.
\end{aligned}$$

Since

$$\mathbf{A}\begin{bmatrix} \eta_1 \\ \eta_2 \\ \dots \\ \eta_{n-l} \end{bmatrix} = -\begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \end{bmatrix},$$

we have

$$2\sum_{j=1}^{n-l} \frac{\lambda_{ij}}{\lambda_{ii}} \eta_i = 2\frac{1}{\lambda_{ii}} \sum_{j=1}^{n-l} \lambda_{ij} \eta_i = -2\frac{1}{\lambda_{ii}}.$$

Hence,

$$\sum_{j=1}^{n-l} \lambda_{ij} E[(\pi_j)^2] = 2\frac{1}{\lambda_{ii}} - 2\eta_i + 2\sum_{j=1}^{n-l} \frac{\lambda_{ij}}{\lambda_{ii}} \eta_i = -2\eta_i.$$

In the matrix form

$$\mathbf{A}\begin{bmatrix} E[\pi_1^2] \\ E[\pi_2^2] \\ \dots \\ E[\pi_{n-1}^2] \end{bmatrix} = -2\begin{bmatrix} \eta_1 \\ \eta_2 \\ \dots \\ \eta_{n-l} \end{bmatrix} = 2\mathbf{A}^{-1}\begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \end{bmatrix}.$$

Therefore,

$$\begin{bmatrix} E[\pi_1^2] \\ E[\pi_2^2] \\ \dots \\ E[\pi_{n-1}^2] \end{bmatrix} = 2\mathbf{A}^{-2} \begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \end{bmatrix}.$$

References

Ramadge PJ, Wonham WM (1987) Supervisory control of a class of discrete event processes. SIAM J Control Optim 25(1):206–230

Lin F, Wonham WM (1988) On observability of discrete-event systems. Inform Sci 44(3):173–198

Cieslak R, Desclaux C, Fawaz AS, Varaiya P (1988) Supervisory control of discrete-event processes with partial observations. IEEE Trans Autom Control 33(3):249–260

Rudie K, Wonham WM (1992) Think globally, act locally: decentralized supervisory control. IEEE Trans Autom Control 37(11):1692–1708

Wonham WM, Cai K (2019) Supervisory control of discrete-event systems. Springer

Sampath M, Sengupta R, Lafortune S, Sinnamohideen K, Teneketzis D (1995) Diagnosability of discrete-event systems. IEEE Trans Autom Control 40(9):1555–1575

Lin F (1994) Diagnosability of discrete event systems and its applications. Discret Event Dyn Syst 4(2):197–212

Yoo T-S, Lafortune S (2002) Polynomial-time verification of diagnosability of partially observed discrete-event systems. IEEE Trans Autom Control 47(9):1491–1495

Jiang S, Huang Z, Chandra V, Kumar R (2001) A polynomial algorithm for testing diagnosability of discrete-event systems. IEEE Trans Autom Control 46(8):1318–1321

Lafortune S, Lin F, Hadjicostis CN (2018) On the history of diagnosability and opacity in discrete event systems. Annu Rev Control 45:257–266

Wonham W, Cai K, Rudie K (2018) Supervisory control of discrete-event systems: a brief history. Annu Rev Control 45:250–256

Cassandras CG Lafortune S (2009) Introduction to discrete event systems. Springer Sci Bus Media

Darling DA, Siegert A (1953) The first passage problem for a continuous markov process. Ann Math Stat 24(4):624–639

Brown M Chaganty NR (1983) On the first passage time distribution for a class of markov chains. Annals Probab 1000–1008

Yao DD (1985) First-passage-time moments of markov processes. J Appl Probab 939–945

Hunter JJ (2018) The computation of the mean first passage times for markov chains. Linear Algebra Appl 549:100–122

Jan ST Afzal R Khan AZ (2015) Transformer failures, causes & impact. In: International conference data mining, civil and mechanical engineering pp 49–52

Murugan R, Ramasamy R (2015) Failure analysis of power transformer for effective maintenance planning in electric utilities. Eng Fail Anal 55:182–192

Zhong J, Li W, Wang C, Yu J, Xu R (2016) Determining optimal inspection intervals in maintenance considering equipment aging failures. IEEE Trans Power Syst 32(2):1474–1482

Christina A, Salam M, Rahman Q, Wen F, Ang S, Voon W (2018) Causes of transformer failures and diagnostic methods-a review. Renew Sust Energ Rev 82:1442–1456

Feng Lin received his B.Eng. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 1982, and the M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Toronto, Toronto, ON, Canada, in 1984 and 1988, respectively. He was a Post-Doctoral Fellow with Harvard University, Cambridge, MA, USA, from 1987 to 1988. Since 1988, he has been with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI, USA, where he is currently a professor. His current research interests include discrete event systems, hybrid systems, robust control, and their applications in alternative energy, biomedical systems, and automotive control. He authored a book entitled "Robust Control Design: An Optimal Control Approach" and coauthored a paper that received a George Axelby outstanding paper award from the IEEE Control Systems Society. He was an associate editor of IEEE Transactions on Automatic Control.

Caisheng Wang received the BS and MS degrees from Chongqing University, China, in 1994 and 1997, respectively, and the Ph.D. degree from Montana State University, Bozeman, MT, in 2006, all in electrical engineering. From August 1997 to May 2002, he worked as an electrical engineer and then a vice department chair in Zhejiang Electric Power Test & Research Institute, Hangzhou, China. Since August 2006, he has joined Wayne State University, where he is currently a Professor at the Department of Electrical and Computer Engineering. His current research interests include modeling and control of power systems and electric vehicles, power electronics, energy storage devices, distributed generation and Microgrids, alternative/hybrid energy power generation systems, and fault diagnosis and on-line monitoring of electric apparatus. He was an Associate Editor of IEEE Transactions on Smart Grid.

Masoud H. Nazari received the MSc degree in Electrical Engineering (Energy Systems) from Sharif University of Technology in 2005, the MSc degree in Engineering & Public Policy from Carnegie Mellon University, Pittsburgh, PA, USA, in 2010 and dual Ph.D. degree in Electrical & Computer Engineering, and Engineering & Public Policy from Carnegie Mellon University in 2012. He was a Postdoctoral Fellow from 2013 to 2015 in the Electrical & Computer Engineering Department at the Georgia Institute of Technology. He was an Assistant Professor of Electrical Engineering with California State University, Long Beach, USA. He is currently an Assistant Professor of Electrical & Computer Engineering with Wayne State University, Detroit, MI, USA. He is the Chair of IEEE Smart Buildings-Loads-Customer Systems (SBLC) Architecture Subcommittee, Editorial Board of IEEE Technology Conferences, and Global Learning Faculty Fellow at WSU. He has been the Primary Investigator of several research projects by National Science Foundation, California Energy Commission, and DTE Energy. Dr. Nazari is the IEEE Senior Member in the Power and Energy Society (PES) and was the recipient of the Best Paper Award in the 2017 North American Power Symposium.

Wenyuan Li received his Ph.D. degree in electrical engineering from Chongqing University, China, in 1987, where he is currently a professor. He worked with BC Hydro in Canada between 1991 and 2015. His interests include smart grids, power system operation, planning, optimization and reliability assessment. He is a Life Fellow of the IEEE, a Fellow of the Canadian Academy of Engineering and a Foreign Member of the Chinese Academy of Engineering. He received several IEEE Power and Energy Society (PES) awards including the IEEE PES Roy Billinton Power System Reliability Award in 2011 and IEEE Canada Electric Power Medal in 2014.