2014

Failure diagnosis and prognosis in stochastic discrete-event and cyber-physical systems

Jun Chen
Iowa State University

Failure diagnosis and prognosis in stochastic discrete-event and cyber-physical systems

by

Jun Chen

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Electrical Engineering

Program of Study Committee:

Ratnesh Kumar, Major Professor

Samik Basu

Nicola Elia

Umesh Vaidya

Zhengyuan Zhu

Iowa State University

Ames, Iowa

2014

DEDICATION

I would like to dedicate this dissertation to my wife Peihan Zhong, my parents Zhennan Chen and Yuhua Jiang, my brother Qifeng Chen and my sister Lanming Chen.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincerest and deepest appreciation to my major professor, Dr. Ratnesh Kumar, for his guidance, patience and persistent support throughout my Ph.D. studies at Iowa State. This dissertation would not have been possible without his insights and advices. His rigorous scholarship and strong passion for exploring new areas will always inspire me.

I would also like to thank my committee members, Dr. Elia and Dr. Vaidya from Department of Electrical and Computer Engineering, Dr. Basu from Department of Computer Science and Dr. Zhu from Department of Statistics, for their valuable advices and assistance on this work.

Last but not the least, I would like to convey special thanks to my beloved wife, parents, brother and sister, and my friends for their understanding and support.

ABSTRACT

In this dissertation we study the problem of fault diagnosis in both discrete event systems and cyber physical systems. Discrete event systems (DESs) are event-driven systems with discrete states that evolve in response to abrupt occurrences of discrete changes (called events). The stochastic DESs are used to characterize the quantitative behavior of the system, by modeling the uncertainty on the occurrence of events as random variables with certain distribution. A stochastic DES is similar to the Markov chain models, with the difference being that, in stochastic DESs, the transition is labeled with the event while the event information is omitted in a Markov chain. Many physical systems, such as manufacturing systems, communication protocols, reactive software, telephone networks, traffic systems, robotics and digital hardware, can be modeled as DESs at a certain level of abstraction.

Fault diagnosis is to detect the occurrence of a fault so as to enable any fault tolerant actions. It is a crucial and challenging problem that has attracted considerable attentions in the literature of software engineering, automotive systems, power systems and nuclear engineering. In this dissertation, we propose the online detection schemes for stochastic DESs and also introduce the notions of missed detections (MDs) and false alarms (FAs), or equivalently, false-negatives and false-positives, for the schemes. The idea is that given any observation (of partially observed events), the detector recursively computes the conditional probability of the nonoccurrence of a fault and issues a "fault" decision if the probability of the nonoccurrence of a fault falls below an appropriately chosen threshold, and issues "no-decision" otherwise. We establish that *S-Diagnosability* is a necessary and sufficient condition for achieving any desired levels of MD and FA rates, where the notion of S-Diagnosability was proposed by Thorsley, *et al.* in 2005, requiring that given any tolerable ambiguity level ρ and error bound τ, there must exist a delay bound n such that for any fault trace, its extensions, longer than n and probability of ambiguity higher than ρ, occur with probability smaller than τ. Algorithms for

determining the detection scheme parameters of detection threshold and detection delay bound for the specified MD and FA rates requirement are also presented, based on the construction of an extended observer, which computes, for each observation sequence, the set of states reached in the system model, along with their probabilities and the number of post-fault transitions executed.

This dissertation also studies the fault diagnosis in cyber physical systems, where the dynamics of the physical systems over discrete sample instances are described by stochastic difference equations, and the nonfault behaviors are specified by linear-time temporal logic (LTL) formulas over sequences of requirement variables that are functions of inputs and states (just as the outputs). We first introduce the notion of an input-output stochastic hybrid automaton (I/O-SHA), and then show that it can be used to model the refinement of a given discrete-time stochastic system against its LTL specification so as to identify the system behaviors that satisfy the nonfault specification versus the ones that violate it in form of reachability of a fault location. For this we propose a refinement algorithm that refines the system model in form of discrete-time stochastic equations with respect to its specification model in form of a Büchi acceptor, and the resulting refinement can be modeled as an I/O-SHA. We further show that the fault detection problem then reduces to a state estimation problem for the I/O-SHA. The performance of the detection protocol is evaluated in terms of its FA and MD rates. We additionally propose the notion of S-Diagnosability for I/O-SHA, which can guarantee the existence of detectors that can achieve any desired FA and MD rates.

We further consider the fault prognosis problem, where the goal is to predict a fault prior to its occurrence, for stochastic DESs. We introduce *m-steps Stochastic-Prognosability*, or simply *S_m-Prognosability*, requiring for any tolerance level ρ and error bound τ, there exists a reaction bound $k \geq m$, such that the set of fault traces for which a fault cannot be predicted k steps in advance with tolerance level ρ, occurs with probability smaller than τ. Similar to the fault diagnosis problem, we formalize the notion of a prognoser that maps observations to decisions by comparing a suitable statistic with a threshold, and show that S_m-Prognosability is a necessary and sufficient condition for the existence of a prognoser with reaction bound at least m (i.e., prediction at least m-steps prior to the occurrence of a fault) that can achieve any specified FA and MD rate requirement. Moreover, we provide a polynomial algorithm for verifying S_m-Prognosability.

CHAPTER 1. INTRODUCTION

Given a set of system behaviors and a system specification identifying the set of system behaviors that are desirable, execution of a trace that violates the specification constitutes a fault, which could be classified as a permanent fault (as studied in [1]) or an intermittent fault (as studied in [2, 3, 4]), and needs to be detected accurately within proper delay bound to ensure timely activation of any fault tolerance action. The problem of fault detection is essential to perform any fault tolerance action and is an important and challenging problem in many disciplines such as software engineering [5], automotive systems [6], power systems [7], nuclear engineering [8], aerospace engineering [9] and digital circuits [10]. The problem of fault detection has been widely researched [11, 12, 13, 14, 15], and is recently studied in the setting of discrete event systems (DESs) [10, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30], distributed/decentralized systems [1, 21, 31, 32, 33, 34, 35, 36, 37, 38], stochastic systems [39, 40, 41, 42, 43, 44, 45, 46, 47] and systems with temporal logic fault specification [48, 49, 50].

In this dissertation, we first consider stochastic DESs with partial observability of events subjected to faults, modeled as execution of fault traces, or equivalently reachability of fault states, meaning that the fault to be detected is of permanent type. We explore the online detection algorithm, as well as its performance evaluation. Then we investigate the fault detection of certain cyber physical systems, whose physical dynamics over discrete sample instances are described by stochastic difference equations with their nonfault behaviors specified by linear-time temporal logic (LTL) formulas over sequences of requirement variables that are functions of inputs and states (just as the outputs). The problem of fault prognosis, where the goal is to predict a fault prior to its occurrence, is also studied.

1.1 Stochastic Discrete Event Systems

Discrete event systems (DESs) are event-driven systems with discrete states that evolve in response to abrupt occurrences of discrete changes (called events). Typical examples of the discrete events include completion of a transaction in a database system, the occurrence of a fault in a manufacture system and the transmission of a signal in a networked system, etc. Many physical systems, such as manufacturing systems, communication protocols, reactive software, telephone networks, traffic systems, robotics and digital hardware, can be modeled as DESs at a certain level of abstraction [51]. In contrast to the continuous systems in which the system state can take continuous value and changes continuously according to the evolution of time, the state of discrete event systems can only take discrete values, which changes in response to the occurrence of events, and in between event occurrences, the system remains in the current state [52]. The events can be the change of an integer value, the pushing of a button, or receiving a printing command for a printer, etc. The behavior of a DES is then consisting of all possible sequences of events the system can execute starting from its initial state.

The stochastic DESs are used to characterize the quantitative behavior of the system, by modeling the uncertainty on the occurrence of events as random variables with certain distribution. The formalism of probabilistic languages were introduced to describe the behavior of stochastic DESs in [53, 54]. In [53] the probabilities on transitions on *each* event from any state sum to one while in [54] the cumulative probability of state change over *all* states should be at most one. In this dissertation we use the model presented in [54], which is similar to the Markov chain models [55], with the difference being that, in stochastic DESs, the transition is labeled with the event while the event information is omitted in a Markov chain.

1.2 Diagnosis of Stochastic Discrete Event Systems

Logical Diagnosability (referred as Diagnosability in [16]) of DESs was introduced in [16], the idea being that a diagnosable DES must possess a delay bound n such that for any fault trace executed by the DES, its ambiguity with a nonfault trace should fully resolve within at most n

steps. Algorithms with polynomial complexity for verifying Logical Diagnosability were given in [17] and [20]. The notion has been extended to decentralized setting in [21], distributed setting in [23] and inference-based setting in [32]. The failure diagnosis for stochastic DESs was later studied in [41] which proposed *Stochastic (S)-Diagnosability* (referred as AA-diagnosability in [41]) requiring that given any tolerable ambiguity level ρ and error bound τ, there must exist a delay bound n such that for any fault trace, its extensions, longer than n and probability of ambiguity higher than ρ, occur with probability smaller than τ. Sufficient method for verifying S-Diagnosability was obtained in [41] that checks for certain structural properties of a diagnoser.

Since the initial work of [41], the following other works on diagnosis of stochastic DESs have appeared in literature. [22] studied the same problem, allowing the observations to be random. Reference [18] later showed that [41] is general enough to also capture any randomized observations, by way of suitably refining the plant model. Problems on counting the occurrences of intermittent/repetitive failure in stochastic DESs was researched in [4], extending the concepts first introduced in [2]. In [3] the authors proposed an approximated minimum mean square error counter for estimating the number of failure occurrence. The sensor selection problem to support diagnosability was introduced in [56] and was adopted for stochastic problems in [24] and [25] for counting the number of routing violations in material flow networks. The diagnosis problem is also investigated in stochastic Petri nets [39], [44]. Besides the diagnosis problem, the control problems for stochastic DESs have been examined in [54, 57, 58, 59, 60, 61].

The above cited works only study the *offline* verification of the S-Diagnosability property; a technique *online* fault detection hasn't yet been examined in literature. This dissertation investigates the online detection schemes for stochastic DESs and also introduces the notions of missed detections (MDs) and false alarms (FAs), or equivalently, false-negatives and false-positives, for the schemes. Due to the probabilistic nature of the problem, MDs and FAs are possible even for S-Diagnosable systems, and we establish that S-Diagnosability is a necessary and sufficient condition for achieving any desired levels of MD and FA rates.

We present a detection scheme, that can achieve the specified MD and FA rates, based on comparing a suitable detection statistic with a suitable detection threshold. We also algorithmically compute the corresponding detection delay bound. The idea is that given any observation (of partially observed events), the detector recursively computes the conditional probability of the nonoccurrence of a fault and issues a “fault” decision if the probability of the nonoccurrence of a fault falls below an appropriately chosen threshold, and issues “no-decision” otherwise. For systems that possesses S-Diagnosability property, there always exists a detection threshold and a delay bound so that this detector is able to achieve any desired level of MD and FA rates. Conversely, the existence of a detector for any desired performance requirement implies that the system possesses the S-Diagnosability property. Algorithms for determining the detection scheme parameters of detection threshold and detection delay bound for the specified MD and FA rates requirement are also presented, based on the construction of an extended observer, which computes, for each observation sequence, the set of states reached in the system model, along with their probabilities and the number of post-fault transitions executed. The algorithms are guaranteed to terminate and the upper bounds on the number of iterations prior to termination are reported as part of the correctness proof of the algorithms. Our detection strategy works for S-Diagnosable system as well as non-S-Diagnosable systems in the same manner. For S-Diagnosable systems it is possible to achieve arbitrary performance requirement for FA and MD rates, while for a non-S-Diagnosable system an arbitrary performance requirement is achievable only for the FA rate, whereas a lower bound exists for the achievable MD rate that is a function of the FA rate, and increases as FA rate requirement is made more stringent by decreasing it. A variant of the above mentioned algorithm is also presented to compute an upper bound for the minimum achievable MD rate for a non-S-Diagnosable system.

1.3 Diagnosis of Cyber Physical Systems

Stochastic hybrid system (SHS) is a system that involves both continuous and discrete stochastic dynamics. The problems of reachability, safety as well as control problem have been address in the literature [62, 63, 64, 65, 66, 67, 68, 69, 70]. For example, the probabilistic reachability problem on discrete time SHS is considered in [62, 63, 66] where the optimal Markov control policy is synthesized by dynamic programming, whereas the continuous time SHS is studied in [67]. The abstraction of a SHS is examined in [64, 65, 69], where the goal is to find a abstraction model which is computational ease and possesses the *exact* or *error-*

bounded behaviors with the original system. Another research topics on SHS is model checking for temporal property [68, 70].

In this dissertation we study fault diagnosis of cyber physical systems, where the physical dynamics over discrete sample instances are described by stochastic difference equations, and the nonfault behaviors are specified by linear-time temporal logic (LTL) formulas over sequences of requirement variables that are functions of inputs and states (just as the outputs). LTL formulas are widely used as correctness requirements (see for example [49, 71]) owing to the fact that they are easier to specify than automata models or formal languages, yet they are compact and expressive.

We introduce the notion of an input-output stochastic hybrid automaton (I/O-SHA), generalizing its logical counterpart presented in [72] by allowing randomness in invariants, guards, data updates, and output assignments. Then we show that I/O-SHA model can be used to model the refinement of a given discrete-time stochastic system against its LTL specification so as to identify the system behaviors that satisfy the nonfault specification versus the ones that violate it in form of reachability of a fault location. For this we propose a refinement algorithm that refines the system model in form of discrete-time stochastic equations with respect to its specification model in form of a Büchi acceptor, and the resulting refinement can be modeled as an I/O-SHA. We further show that the fault detection problem then reduces to a state estimation problem for the I/O-SHA, i.e., the probability of specification violation versus no violation can be estimated via a state estimation computation in the I/O-SHA model. This statistic, the probability of no-fault, is then used for issuing detection decisions. The performance of the detection protocol is evaluated in terms of its FA and MD rates. We additionally propose the notion of S-Diagnosability for I/O-SHA, which can guarantee the existence of detectors that can achieve any desired FA and MD rates.

1.4 Prognosis of Stochastic Discrete Event Systems

We further consider the fault prognosis problem, where the goal is to predict a fault prior to its occurrence, for stochastic DESs. The problem of predicting a fault prior to its occurrence is a well researched area (see for example [73, 74, 75, 76, 77, 78]). In [74] the notion of uniformly bounded prognosability of fault was formulated for logical discrete event systems (DESs), where each fault trace must possess a nonfault-prefix such that for all indistinguishable traces, a future fault is inevitable within a bounded delay that is uniform across all fault-traces. The notion was later extended to the decentralized setting in [75] and the requirement of the existence of a uniform bound was also removed. Reference [75] also established that the notion of prognosability is equivalent to the existence of a prognoser with no FA and no MD. The issue of prognosability under a general decentralized inferencing mechanism was proposed in [79], where a prognostic decision involved inferencing among a group of local prognosers over their local decisions and their ambiguity levels, and the notion of inference-prognosability and its verification was introduced to capture the necessity and sufficiency of inferencing based decentralized prognosis. The problem of distributed prognosability under bounded-delay communications among the local prognosers was studied in [80], where the notion of joint-prognosability and its verification was proposed.

We introduce the notion of m-*steps Stochastic-Prognosability*, or simply S_m-*Prognosability*, requiring for any tolerance level ρ and error bound τ, there exists a reaction bound $k \geq m$, such that the set of fault traces for which a fault cannot be predicted k steps in advance with tolerance level ρ, occurs with probability smaller than τ. Similar to the fault diagnosis problem, we formalize the notion of a prognoser that maps observations to decisions by comparing a suitable statistic with a threshold, and show that S_m-Prognosability is a necessary and sufficient condition for the existence of a prognoser with reaction bound at least m (i.e., prediction at least m-steps prior to the occurrence of a fault) that can achieve any specified FA and MD rate requirement. In this sense S_m-Prognosability can be viewed as a generalization of the logical prognosability, since it provides a basis for the existence and synthesis of a prognoser that can achieve a user-specified level of FA and MD. In contrast, the logical version is rather rigid, offering no further options for systems that fail to be logically prognosable, even when there may exist a prognoser that can achieve a satisfying performance as measured in terms of FA and MD rates. Further, we also provide a polynomial algorithm for verifying S_m-Prognosability.

1.5 Organization of Dissertation

The rest of this dissertation is organized as follows.

Chapter 2 contains the notations and preliminaries which are necessary for this dissertation, including language, Markov chain, the definition of S-Diagnosability, as well as linear-time temporal logic (LTL).

In Chapter 3, we present a fault detection scheme for stochastic DES, which recursively computes the conditional probability of the nonoccurrence of a fault and issues a "fault" decision if the probability of the nonoccurrence of a fault falls below an appropriately chosen threshold, and issues "no-decision" otherwise. For systems that possesses S-Diagnosability property, there always exists a detection threshold and a delay bound so that this detector is able to achieve any desired level of MD and FA rates. Conversely, the existence of a detector for any desired performance requirement implies that the system possesses the S-Diagnosability property. Algorithms for determining the detection scheme parameters of detection threshold and detection delay bound for the specified MD and FA rates requirement are also presented, based on the construction of an extended observer, which computes, for each observation sequence, the set of states reached in the system model, along with their probabilities and the number of post-fault transitions executed. We also explore the non-S-Diagnosable system, for which an arbitrary performance requirement is achievable only for the FA rate, whereas a lower bound exists for the achievable MD rate that is a function of the FA rate, and increases as FA rate requirement is made more stringent by decreasing it. A variant of the above mentioned algorithm is also presented to compute an upper bound for the minimum achievable MD rate for a non-S-Diagnosable system.

In Chapter 4, we study fault diagnosis of cyber physical systems, where the physical dynamics over discrete sample instances are described by stochastic difference equations and the nonfault behaviors are specified by linear-time temporal logic (LTL) formulas over sequences of requirement variables that are functions of inputs and states (just as the outputs). Firstly, we propose the notion of input-output stochastic hybrid automaton (I/O-SHA), extending the logical input-output hybrid automaton (I/O-HA) introduced in [72], by allowing randomness in invariants, guards, data updates, and output assignments. Secondly, we present a method to refine a given discrete-time stochastic system against a *deterministic* (LTL) specification (one that can be accepted by a deterministic Büchi automaton), where the refinement is an I/O-SHA with the property that the violation of the LTL specification can be captured as a reachability property, and the probability of specification violation versus no violation can be estimated via a state estimation computation in the I/O-SHA model. Thirdly, we provide a procedure to recursively compute the probability of fault versus no-fault (specification violation versus no-violation), which is used as a statistic for issuing detection decisions. Finally, we propose the notion of S-Diagnosability for I/O-SHA, which can guarantee the existence of detectors that can achieve any desired FA and MD rates.

In Chapter 5, the problem of fault prognosis, where the goal is to predict a fault prior to its occurrence, is investigated. We propose the notion of S_m-Prognosability which requires that a fault should be statistically predicted at least m steps in advance with large probability. We show that S_m-Prognosability is necessary and sufficient for the existence of a m-prognoser satisfying arbitrary FA and MD rates requirement. Polynomial verification algorithm for S_m-Prognosability is also presented. Practical examples on "crowd" protocol and HVAC system are provided to illustrate the work in this chapter.

In Chapter 6, we summarize the work and conclude with the discussions of future work.

CHAPTER 2. NOTATIONS AND PRELIMINARIES

This chapter contains the notations and preliminaries which are necessary for this dissertation, including language, Markov chain, the definition of S-Diagnosability, as well as linear-time temporal logic (LTL). A more thorough introduction can be found in [51, 54, 55, 81, 82, 83].

2.1 Language, Automaton and Markov Chain

For an event set Σ, define $\overline{\Sigma} := \Sigma \cup \{\epsilon\}$, where ϵ denotes "no-event". The set of all finite length event sequences over Σ, including ϵ, is denoted as Σ^*. A *trace* is a member of Σ^* and a *language* is a subset of Σ^*. We use $s \leq t$ to denote that $s \in \Sigma^*$ is a prefix of $t \in \Sigma^*$, $pr(s)$ to denote the set of all prefixes of s, and $|s|$ to denote the length of s or the number of events in s. For $\sim \in \{<, \leq, >, \geq, =\}$ and $n \in \mathbb{N}$, where $\mathbb{N}$ denotes the set of all nonnegative integers, define $\Sigma^{\sim n} := \{s \in \Sigma^* : |s| \sim n\}$ and denote $\Sigma^{=n}$ as Σ^n for simplicity. For $L \subseteq \Sigma^*$, its prefix-closure is defined as $pr(L) := \bigcup_{s \in L} pr(s)$, and L is said to be prefix-closed (or simply closed) if $pr(L) = L$. Given two languages L_1 and L_2, their *concatenation* is defined as $L_1L_2 := \{st : s \in L_1, t \in L_2\}$, the set of traces in L_1 *after* L_2 is defined as $L_1 \backslash L_2 := \{t \in \Sigma^* : \exists s \in L_2, st \in L_1\}$, and the set of traces in L_1 *quotient* L_2 is defined as $L_1/L_2 := \{s \in pr(L_1) : \exists t \in L_2, st \in L_1\}$.

A stochastic DES can be modeled by a *stochastic automaton* $G = (X, \Sigma, \alpha, x_0)$, where X is the set of states, Σ is the set of events, $x_0 \in X$ is the initial state, and $\alpha : X \times \Sigma \times X \to [0, 1]$ is the transition probability function [54] satisfying $\forall x \in X, \sum_{\sigma \in \Sigma} \sum_{x' \in X} \alpha(x, \sigma, x') = 1$, i.e., there is no "termination" at any of the states. (Note there is no loss of generality in assuming no termination, since otherwise, one can augment the model with a newly introduced "termination-state", and transitions from each state to the termination state on a newly introduced "termination-event" that is unobservable and whose occurrence probability equals the probability of termination of the said state.) G is non-stochastic if $\alpha : X \times \Sigma \times X \to \{0,1\}$, and a non-stochastic DES is deterministic if $\forall x \in X, \sigma \in \Sigma, \sum_{x' \in X} \alpha(x, \sigma, x') \in \{0,1\}$, i.e., each state has at most one transition on each event. The transition probability function α can be generalized to $\alpha : X \times \Sigma^* \times X$ in a natural way: $\forall x_i, x_j \in X, s \in \Sigma^*, \sigma \in \Sigma, \alpha(x_i, s\sigma, x_j) = \sum_{x_k \in X} \alpha(x_i, s, x_k)\alpha(x_k, \sigma, x_j)$, and $\alpha(x_i, \epsilon, x_j) = 1$ if $x_i = x_j$ and 0 otherwise. Define a *transition* in G as a triple $(x_i, \sigma, x_j) \in X \times \Sigma \times X$ where $\alpha(x_i, \sigma, x_j) > 0$ and define the language generated by G as $L(G) := \{s \in \Sigma^* : \exists x \in X, \alpha(x_0, s, x) > 0\}$.

The initialization of a stochastic automaton can also be modeled as an initial state distribution π_0 over the state space X instead of an initial state x_0, where π_0 is a row vector whose elements are nonnegative and sum to one. In this case, the *generating probability* of an event trace $s \in L(G)$ is given by $\alpha_G(s) := \sum_{x_j \in X} \pi_0(x_j) \sum_{x \in X} \alpha(x_j, s, x)$. Two automata, defined over the same event set, are said to be *p-equivalent* if for every event trace, the generating probability in two automata are equal [84]. A polynomial time algorithm for checking whether or not two automata are p-equivalent is presented in [84], which also returns a minimal length event trace that serves as a counterexample (has different generating probabilities in the two automata) in case the two automata are not p-equivalent.

To represent the limited sensing capabilities of a diagnoser/prognoser, we introduce an event observation mask, $M : \overline{\Sigma} \to \overline{\Delta}$, where Δ is the set of observed symbols and $M(\epsilon) = \epsilon$. An event σ is *unobservable* if $M(\sigma) = \epsilon$. The set of unobservable events is denoted as Σ_{uo}, and so the set of observable events is given by $\Sigma - \Sigma_{uo}$. Note this mask function is more general than a natural projection in that it allows *unobservable* events (with mask value ϵ) as well as *partially observable* events (with mask value non-ϵ but identical to the mask value of another event). For example in a material handling system, it may be possible to sense the arrival of a part but not its type, and so all arrivals at a certain sensor would be indistinguishable from each other, yet not fully unobserved. The observation mask can be generalized to $M : \Sigma^* \to \Delta^*$ in a natural way: $\forall s \in \Sigma^*, \sigma \in \overline{\Sigma}, L \subseteq \Sigma^*$, $M(\epsilon) = \epsilon$, $M(s\sigma) = M(s)M(\sigma)$ and $M(L) = \{M(s) : s \in L\}$.

Example 1. Fig. 2.1 is an example of a stochastic automaton G. The set of states is $X = \{0, 1, 2, 3\}$ with initial state $x_0 = 0$, event set $\Sigma = \{a, b, c, f\}$. A state is depicted as a node, whereas a transition is depicted as an edge between its origin and termination states, with its

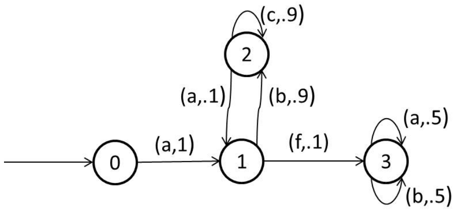


Figure 2.1 Stochastic automaton G for Example 1.

event name and probability value labeled on the edge. The observation mask M is such that $M(f) = \epsilon$ and otherwise $M(\sigma) = \sigma$. □

Given a stochastic DES $G = (X, \Sigma, \alpha, x_0)$, its embedded Markov chain is obtained by abstracting out the event information associated with the transitions, i.e., the embedded Markov chain is given by (X, Ω, x_0), where Ω is a size-$|X| \times |X|$ square matrix with ijth entry given by $\Omega_{ij} = \sum_{\sigma\in\Sigma} \alpha(x_i, \sigma, x_j)$. (Note the Markov chain contains *at most one* transition between a pair of states *in each direction* and *does not carry* an event label.) The following is a useful property of a finite state Markov chain, [55].

Property 1. Let X be the state space of a finite state Markov chain and $X = X_R \cup X_T$, where X_R and X_T denote the set of recurrent and transient states, respectively. Let $x \in X$ be an arbitrary state of the chain and t be any transition sequence starting from x. Then

$$(\forall\tau > 0)(\exists n \in \mathbb{N})Pr(t : \exists x' \in X_T, \alpha(x, t, x') > 0, |t| \geq n) < \tau,$$

which means that as the number of transitions increases, the probability of the Markov chain being in a transient state approaches zero.

For a stochastic automaton $G = (X, \Sigma, \alpha, x_0)$, a *component* $C = (X_C, \alpha_C)$ of G is a "subgraph" of G, i.e., $X_C \subseteq X$ and $\forall x, x' \in X_C$ and $\sigma \in \Sigma$, $\alpha_C(x, \sigma, x') = \alpha(x, \sigma, x')$, whenever the latter is defined. C is said to be a *strongly connected component* (SCC) or *irreducible* if $\forall x, x' \in X_C$, $\exists s \in \Sigma^*$ such that $\alpha_C(x, s, x') > 0$. A SCC C is said to be *closed* if for each $x \in X_C$, $\sum_{\sigma\in\Sigma}\sum_{x'\in X_C} \alpha_C(x, \sigma, x') = 1$. The states which belong to a closed SCC are *recurrent states* and the remaining states (that do not belong to any closed SCC) are *transient states*. A closed or recurrent SCC with finitely many states possesses a unique *stationary state* *distribution* after reaching which the state distribution remains unchanged. A state is *periodic* with *period* $k \geq 2$, if any return to this state must occur in multiples of k steps. A state is *aperiodic* if it is not periodic. A SCC is aperiodic if it contains an aperiodic state (and in which case all its states are also aperiodic) [55].

A component with dual transition distribution is denoted as $C = (X_C, \{\alpha_C^1, \alpha_C^2\})$, where transitions are associated with a pair of transition distribution functions α_C^1 and α_C^2. A component C with dual distribution is a *bi-SCC* if both $C_1 = (X_C, \alpha_C^1)$ and $C_2 = (X_C, \alpha_C^2)$ are strongly connected. A bi-SCC C is a *bi-closed* SCC if both $C_1 = (X_C, \alpha_C^1)$ and $C_2 = (X_C, \alpha_C^2)$ are closed. For a bi-closed SCC C with event labels Σ, we can construct two embedded stochastic automata $A_C^1 = (X_C, \Sigma, \alpha_C^1, \pi_C^1)$ and $A_C^2 = (X_C, \Sigma, \alpha_C^2, \pi_C^2)$, where π_C^1 and π_C^2 are the stationary state distributions of A_C^1 and A_C^2 respectively. A bi-closed SCC C is said to be *p-equivalent* if its embedded automata A_C^1 and A_C^2 are p-equivalent.

For a stochastic automaton $G = (X, \Sigma, \alpha, x_0)$ with generated language $L(G) = L$, let $K \subseteq L$ be a nonempty closed sublanguage representing a nonfault specification for G, i.e., $L - K$ consists of behaviors that execute some fault. Then the task of diagnosis is to detect the execution of any trace in $L - K$ after its execution, within certain delay bound and with sufficient confidence. Let $K \subseteq L$ be generated by a *deterministic* automaton $R = (Q, \Sigma, \beta, q)$ such that $L(R) = K$ (from now on we interchangably use K and R to refer to the "nonfault specification"). Then the refinement of the plant with respect to the specification, denoted as G^R, can be used to capture the fault traces in the form of the reachability of a fault state carrying the label F in G^R, which is given by $G^R := (Y, \Sigma, \gamma, (x_0, q_0))$, where $Y = X \times \overline{Q}$ and $\overline{Q} = Q \cup \{F\}$, and $\forall (x, \overline{q}), (x', \overline{q}') \in X \times \overline{Q}, \sigma \in \Sigma, \gamma((x, \overline{q}), \sigma, (x', \overline{q}')) = \alpha(x, \sigma, x')$ if the following holds:

$$(\overline{q}, \overline{q}' \in Q \wedge \beta(\overline{q}, \sigma, \overline{q}') > 0) \vee (\overline{q} = \overline{q}' = F) \vee \left(\overline{q}' = F \wedge \sum_{q \in Q} \beta(\overline{q}, \sigma, q) = 0 \right),$$

and otherwise $\gamma((x, \overline{q}), \sigma, (x', \overline{q}')) = 0$. Then it can be seen that the refined plant G^R has the following properties: (1) $L(G^R) = L(G) = L$, (2) any fault trace $s \in L - K$ transitions the refinement G^R to a fault state (a state containing F as its second coordinate), and (3) the occurrence probability of each trace in G^R is the same as that in G, i.e., $\sum_{x \in X} \alpha(x_0, s, x) =$

$\sum_{(x,\overline{q})\in X\times\overline{Q}}\gamma((x_0,q_0),s,(x,\overline{q}))$.

For $y_i, y_j \in Y$ and $\delta \in \Delta$, define the set of traces originating at y_i, terminating at y_j and executing a sequence of unobservable events followed by a single observable event with observation δ as $L_{G^R}(y_i,\delta,y_j) := \{s \in \Sigma^* : s = u\sigma, M(u) = \epsilon, M(\sigma) = \delta, \gamma(y_i,s,y_j) > 0\}$. Define $\alpha(L_{G^R}(y_i,\delta,y_j)) := \sum_{s\in L_{G^R}(y_i,\delta,y_j)}\gamma(y_i,s,y_j)$ as the occurrence probability of traces in $L_{G^R}(y_i,\delta,y_j)$ and denote it as $\mu_{i,\delta,j}$ for short. Also define $\lambda_{ij} = \sum_{\sigma\in\Sigma_{uo}}\gamma(y_i,\sigma,y_j)$ as the probability of transitioning from y_i to y_j while executing a single unobservable event. Then it can be seen that $\mu_{i,\delta,j} = \sum_k \lambda_{ik}\mu_{k,\delta,j} + \sum_{\sigma\in\Sigma:M(\sigma)=\delta}\gamma(y_i,\sigma,y_j)$, where the first term on the right hand side (RHS) involves transitioning in at least two steps via some intermediate state, whereas the second RHS term involves transitioning directly in exactly one step. Thus for each $\delta \in \Delta$, given the values $\{\lambda_{ij} | i,j \in Y\}$ and $\{\sum_{\sigma\in\Sigma:M(\sigma)=\delta}\gamma(y_i,\sigma,y_j) | i,j \in Y\}$, all the probabilities $\{\mu_{i,\delta,j} | i,j \in Y\}$ can be found by solving the following matrix equation (see for example [85] for a similar matrix equation):

$$\boldsymbol{\mu}(\delta) \;=\; \boldsymbol{\lambda}\boldsymbol{\mu}(\delta) + \boldsymbol{\gamma}(\delta), \qquad (2.1)$$

where $\boldsymbol{\mu}(\delta)$, $\boldsymbol{\lambda}$ and $\boldsymbol{\gamma}(\delta)$ are all $|Y| \times |Y|$ square matrices whose ijth elements are given by $\mu_{i,\delta,j}$, λ_{ij} and $\sum_{\sigma\in\Sigma:M(\sigma)=\delta}\gamma(y_i,\sigma,y_j)$, respectively. The complexity of finding $\mu_{i,\delta,j}$ by solving equation (2.1) is $O(|Y|^3)$.

Example 2. For system presented in Fig. 2.1, the deterministic nonfault specification R is given in Fig. 2.2. Then the refined plant G^R is shown in Fig. 2.3. Let the state space of G^R be $Y = \{y_1 = (0,0), y_2 = (1,1), y_3 = (2,2), y_4 = (3,F)\}$. By solving matrix equations (2.1), we get

$$\boldsymbol{\mu}(a) \;=\; \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & .05 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & .5 \end{bmatrix}$$

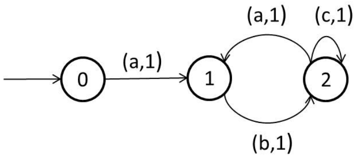


Figure 2.2 Nonfault specification R for Example 2.

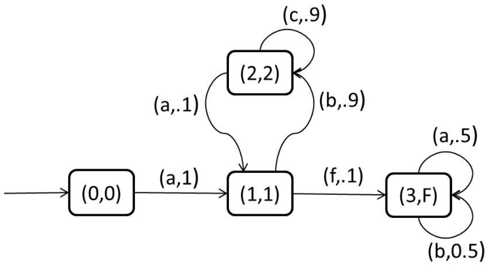


Figure 2.3 The refined plant for Example 2.

$$\boldsymbol{\mu}(b) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & .9 & .05 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & .5 \end{bmatrix}$$

$$\boldsymbol{\mu}(c) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & .9 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

□

2.2 Stochastic Diagnosability of DESs

Given a stochastic DES $G = (X, \Sigma, \alpha, x_0)$ and its deterministic nonfault specification R, with $L = L(G)$ and $K = L(R)$, $L - K \subseteq L$ is the set of all fault traces. The objective of the diagnosability problem is to determine, under what conditions the occurrence of a fault trace $s \in L - K$ can be detected within an uniformly bounded delay. The definition of S-

Diagnosability requires that given any tolerance level ρ and error bound τ, there must exist a delay bound n such that for any fault trace $s \in L - K$, its extensions, longer than n and probability of ambiguity higher than ρ, occur with probability smaller than τ.

Definition 1. Given a stochastic DES $G = (X, \Sigma, \alpha, x_0)$, deterministic nonfault specification $R = (Q, \Sigma, \beta, q_0)$ with generated languages $L = L(G)$ and $K = L(R)$, (L, K) is said to be *Stochastically Diagnosable*, or simply *S-Diagnosable*, if

$$(\forall \tau > 0 \wedge \forall \rho > 0)(\exists n \in \mathbb{N})(\forall s \in L - K) Pr(t : t \in L \backslash s, |t| \geq n, Pr_{amb}(st) > \rho) < \tau, \tag{2.2}$$

where $Pr_{amb} : L - K \to [0, 1]$ is a map that assigns to each fault trace $s \in L - K$, the probability of s being ambiguous, which is the conditional probability of all nonfault indistinguishable traces conditioned by the fact that ambiguity can only arise from the system traces that produce the same observation as s, and is given by:

$$Pr_{amb}(s) = Pr(u \in K | M(u) = M(s)) = \frac{Pr(u \in K : M(u) = M(s))}{Pr(u \in L : M(u) = M(s))} \tag{2.3}$$

Note in the definition of $Pr_{amb}(s)$, "|" denotes the conditioning operation.

Remark 1. The definition of S-Diagnosability introduced above can be seen to be the same as AA-diagnosability (see Definition 2 below taken from [41]).

Definition 2 ([41]). A live, prefix-closed language L is AA-diagnosable with respect to an observation mask M and a set of transition probability p if

$$(\forall \tau > 0 \wedge \forall \alpha < 1)(\exists n \in \mathbb{N})(\forall s \in L - K) Pr(t : t \in L \backslash s, |t| \geq n, D_\alpha(st) = 0) < \tau,$$

where the D_α function is defined as

$$D_\alpha(st) = \begin{cases} 1 & \text{if } Pr(u \in L - K | M(u) = M(st)) > \alpha \\ 0 & \text{otherwise.} \end{cases}$$

It is trivial to show that for a given s and its extension t, $D_\alpha(st) = 0$ if and only if the set of ambiguous nonfault traces occur with high probability, as can be seen: $D_\alpha(st) = 0 \Leftrightarrow Pr(u \in L - K | M(u) = M(st)) \leq \alpha \Leftrightarrow Pr(u \in K | M(u) = M(st)) \geq 1 - \alpha \Leftrightarrow Pr_{amb}(st) \geq 1 - \alpha =: \rho$, and therefore the Definition 1 and 2 presented above are equivalent. □

The next definition introduces a stronger version, called SS-Diagnosability (referred as A-diagnosability in [41]), by setting $\rho = 0$ in Definition 1.

Definition 3. Given a stochastic DES $G = (X, \Sigma, \alpha, x_0)$, deterministic nonfault specification $R = (Q, \Sigma, \beta, q_0)$ with generated languages $L = L(G)$ and $K = L(R)$, (L, K) is said to be *Strongly Stochastically Diagnosable*, or simply *SS-Diagnosable*, if

$$(\forall \tau > 0)(\exists n \in \mathbb{N})(\forall s \in L - K) Pr(t : t \in L \backslash s, |t| \geq n, Pr_{amb}(st) > 0) < \tau.$$

Remark 2. The definition of SS-Diagnosability introduced above can be seen to be the same as A-diagnosability proposed in [41] as demonstrated next. Consider the definition of [41].

Definition 4 ([41]). A live, prefix-closed language L is A-diagnosable with respect to an observation mask M and a set of transition probability p if

$$(\forall \tau > 0)(\exists n \in \mathbb{N})(\forall s \in L - K) Pr(t : t \in L \backslash s, |t| \geq n, D(st) = 0) < \tau,$$

where the D function is defined as

$$D(st) = \begin{cases} 1 & \text{if } M(u) = M(st) \Rightarrow u \in L - K \\ 0 & \text{otherwise} \end{cases}.$$

It is trivial to show that in Definition 4, $D(st) = 0$ if and only if $Pr_{amb}(st) > 0$ and therefore the Definition 3 and 4 presented above are equivalent. □

Example 3. Consider the system G and nonfault specification R in Fig. 2.4. The set of states is $X = \{0, 1, 2, 3\}$ with initial state $x_0 = 0$, event set $\Sigma = \{a, b, c, \sigma_{uo}, \sigma_f\}$. In this example, σ_{uo} and σ_f can not be detected by any sensor, whereas the observability of events a, b, c can vary depending on the configuration of the sensors used. Some examples we will use below are the following three projection masks for which observable events have identity masks, and so only the unobservable events are mentioned (note while all of three mask functions considered above are natural projections, our framework allows more general non-projection masks):

- Observation mask M_1: $\Sigma_{uo} = \{c, \sigma_f, \sigma_{uo}\}$;
- Observation mask M_2: $\Sigma_{uo} = \{b, \sigma_f, \sigma_{uo}\}$;

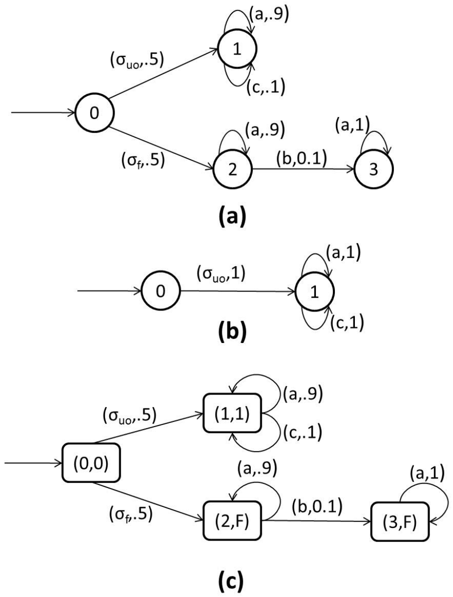


Figure 2.4 System, specification, refinement for Example 3: (a) Stochastic automaton G. (b) Deterministic nonfault specification R. (c) Refinement G^R.

- Observation mask M_3: $\Sigma_{uo} = \{b, c, \sigma_f, \sigma_{uo}\}$.

According to Definition 1 and 3, if the observation mask is M_1, then the system is SS-Diagnosable: for a fault trace $s \in \sigma_f a^* b a^*$, $Pr_{amb}(st) = 0$ for all $t \in L\backslash s$ since b can be observed after the execution of a fault and no b is possible after a nonfault trace in $\sigma_{uo}(a+c)^*$; whereas for a fault trace $s \in \sigma_f a^*$, $Pr_{amb}(st) > 0$ if and only if no b is executed, i.e., $t = a^n$, whose probability approaches zero as n grows arbitrarily large.

If the observation mask is M_2 instead of M_1, i.e., b is unobservable while c is observable, then the system is not SS-Diagnosable, since for every fault trace $s \in \sigma_f a^* \cup \sigma_f a^* b a^*$, there is a nonfault trace $s' \in \sigma_{uo} a^*$ that has the same observation as s, namely a sequence of a's, and so $Pr(t : t \in L\backslash s, |t| \geq n, Pr_{amb}(st) > 0) = 1$ for all $n \in \mathbb{N}$. However, since for any fault trace s and its extension t, with $n := |st|$, $Pr_{amb}(st) = 0.9^n/1$, which decreases as n increases. So we can always choose an $n \in \mathbb{N}$ such that $0.9^n < \rho$, i.e., $Pr_{amb}(st) < \rho$ for all t that is longer than n. Thus the system under the observation mask M_2 is S-Diagnosable (even though not strongly).

In the case of the observation mask M_3, i.e., both b and c being unobservable, the system is not S-Diagnosable, since for any fault trace s and its extension $t \in L\backslash s$, we have $Pr_{amb}(st) = 0.5$, which is a constant. □

2.3 Linear-time Temporal Logic

Later in this dissertation, we study the fault diagnosis in cyber physical systems, where the physical system is subject to disturbance and noise, as modeled by stochastic difference equations:

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, v_k) \\ r_k &= g(x_k, u_k) \\ y_k &= h(x_k, u_k, w_k). \end{aligned}$$

where u, x, r, y, v, w represent, respectively, the input, state, requirement (unobserved), output (observed), disturbance and noise variables, and k is the time index. Note the requirement variable, being user-defined, is independent of disturbance or noise. The properties of the nonfault system behaviors are described by using a LTL formula over the requirement variables, which may not be directly observed and hence must be estimated from the observations of inputs and outputs. In the following we present a brief description of LTL; a more thorough introduction can be found in [81, 82, 83].

Let $M_d = (L_d, \delta, AP, label)$ be a state transition graph, where L_d is the set of states, $\delta : L_d \rightarrow 2^{L_d}$ is a total transition relation, i.e., $\forall l \in L_d, \delta(l) \neq \emptyset$, AP is a finite set of atomic proposition symbols, and $label : L_d \rightarrow 2^{AP}$ is a function that labels each state with the set of atomic propositions true at that state. A sequence of states $\pi = (l_0(\pi), l_1(\pi), \dots)$ is a *state trace* in M_d if $l_{i+1}(\pi) \in \delta(l_i(\pi))$ for every $i \in \{0, 1, \dots\}$. $\pi^k = (l_k(\pi), l_{k+1}(\pi), \dots)$, where $k \in \mathbb{N}$, is used to denote the suffix of π starting from index k. A *proposition trace* over an atomic proposition set AP is defined as a sequence of set of atomic propositions, $\pi_p = (label_0, label_1, \dots)$ such that $label_i \subseteq AP, \forall i \in \{0, 1, \dots\}$. A proposition trace $\pi_p = (label_0, label_1, \dots)$ over AP is said to be *contained* in M_d if there exists a state trace $\pi = (l_0, l_1, \dots)$ in M_d such that $label_i = label(l_i), \forall i \in \{0, 1, \dots\}$, in which case π_p is said to be associated with π.

LTL temporal logic is a formalism for describing properties of sequences of states. Such properties are expressed using *temporal operators* of the temporal logic which include:

- X ("next time"): it requires that a property hold in the next state of the state trace;
- U ("until"): it is used to combine two properties. The combined property holds if there is a state on the state trace where the second property holds, and at every preceding state on the trace, the first property holds;
- F ("eventually" or "in the future"): it requires that a property will hold at some future state on the state trace;
- G ("always" or "globally"): it requires that a property holds at every state on the trace; and
- B ("before"): it also combines two properties. It requires that if there is a state on the state trace where the second property holds, then there exists a preceding state on the trace where the first property holds.

We have the following relations among the above operators, where ϕ denotes a temporal logic formula:

- $F\phi \equiv trueU\phi$,
- $G\phi \equiv \neg F \neg \phi$, and
- $\phi Bg \equiv \neg(\neg \phi U g)$.

So we can use X and U to express all the other temporal operators. LTL formulas are generated by the following rules:

P1) if $p \in AP$, then p is a LTL formula;

P2) if ϕ_1 and ϕ_2 are LTL formulas, then so are $\neg \phi_1$ and $\phi_1 \wedge \phi_2$;

P3) if ϕ_1 and ϕ_2 are LTL formulas, then so are $X\phi_1$ and $\phi_1 U \phi_2$.

The semantics of LTL can be defined with respect to the *infinite* state traces in a state transition graph $M_d = (L_d, \delta, AP, label)$. For a LTL formula ϕ, we use the notation $< M_d, \pi > \models f$ (resp., $< M_d, \pi > \not\models f$) to denote that f holds (resp., does not hold) along the infinite state trace π in M_d. The relation $\models$ is defined inductively as follows:

1. $\forall p \in AP, \pi \models p$ if and only if $p \in label(l_0(\pi))$.

2. $\pi \models \neg\phi$ if and only if $\pi \not\models \phi$.

3. $\pi \models \phi_1 \wedge \phi_2$ if and only if $\pi \models \phi_1$ and $\pi \models \phi_2$.

4. $\pi \models X\phi$ if and only if $\pi^1 \models \phi$.

5. $\pi \models \phi_1 U \phi_2$ if and only if there exists a k such that $\pi^k \models \phi_2$ and for all $j \leq k-1, \pi^j \models \phi_1$.

The semantics of LTL formulas can also be expressed over infinite length proposition traces without referring to any specific state transition graph. This is done by replacing the first condition shown previously with

$$\forall p \in AP, \pi_p = (label_0, label_1, \dots) \models p \Leftrightarrow p \in label_0,$$

where π_p is an infinite proposition trace over AP, i.e., $label_i \subseteq AP$ for all $i \geq 0$. While the semantics of LTL are defined over infinite traces, it can also be extended to finite traces: A finite trace $(l_0, \dots, l_n)$ satisfies a LTL formula ϕ if and only if the infinite trace $(l_0, \dots, l_n, l_n, \dots)$ satisfies ϕ [82].

Given a LTL formula ϕ, denote S_ϕ as the set of all infinitely long proposition traces over AP satisfying ϕ. Then we can obtain a generalized nondeterministic Büchi automaton T_ϕ ([81]) that accepts S_ϕ. To construct T_ϕ, we first put ϕ into *negation normal form*, in which negation is only applied at the atomic level. Then we rewrite each subformula of the form Fg as $TrueUg$. Let $|\phi|$ be the number of subformulas of the form $\lambda U \mu$. Then the generalized nondeterministic Büchi automaton has $|\phi|$ sets of accepting states and is of the form:

$$T_\phi = (L_\phi, 2^{AP}, \delta_\phi, l_0^\phi, \mathcal{L}_\phi)$$

where

- L_ϕ is the set of states;
- $\delta_\phi : L_\phi \times 2^{AP} \to L_\phi$ is the transition relation;
- l_0^ϕ is the initial state, and
- $\mathcal{L}_\phi \subseteq 2^{L_\phi}$ is the generalized Büchi acceptance condition, such that for each subformula of the form $\lambda U \mu$ in ϕ, there exists a $\mathcal{L} \in \mathcal{L}_\phi$ which is used to capture the fulfillment of $\lambda U \mu$.

When $|\mathcal{L}_\phi| = 1$, then the generalized Büchi automaton reduces to a standard one. An infinite length proposition trace $\pi_p = (label_1, label_2, \dots)$ over AP is accepted by T_ϕ if and only if there exists an infinite length state trace $\pi = (l_0^\phi, l_1, \dots)$ in T_ϕ such that $l_i \in \delta_f(l_{i-1}, label_i)$ for all $i \geq 1$, and π visits each set of locations in $\mathcal{L}_\phi$ infinitely often. Then the set of all infinite length proposition traces accepted by T_ϕ, called its ω-*language*, equals S_ϕ.

While every LTL formula can be characterized as the ω-language accepted by a nondeterministic Büchi automaton, only certain fragments of LTL can be modeled as the ω-language accepted by a *deterministic* Büchi automaton. In this dissertation we only consider *prediagnosable* LTL formulas (see Definition 5 in Chapter 4) that can be accepted by deterministic Büchi automata.

CHAPTER 3. FAILURE DIAGNOSIS OF STOCHASTIC DES

In this chapter, we present a detector for online fault detection of stochastic DESs, and show that the S-Diagnosability property is a necessary and sufficient condition of the existence of the aforementioned detector. Algorithms for computing detector parameters for given specified performance requirements are also presented for both S-Diagnosable and non-S-Diagnosable, while in the latter case the termination of the proposed algorithm is not guaranteed.

3.1 Online Detector and its Existence Condition

3.1.1 Computation of Likelihood of No-fault

When the system executes a trace $s \in L$, an observation $o = M(s)$ is received by a fault detector. In order to issue a "fault" decision versus no-decision for the observation $o = M(s)$, we propose the detector compute the likelihood of no-fault, and issue a "fault" decision if this likelihood of no-fault is small (i.e., below a suitable threshold), and otherwise issue no-decision. In this subsection we present how this likelihood can be recursively computed. With a slight abuse of notation, we denote the no-fault likelihood function $P_N : M(L) \to [0,1]$ and define it to be the conditional probability of nonoccurrence of a fault following any observation $o \in M(L)$:

$$P_N(o) \quad := \quad Pr(u \in K | M(u) = o) = \frac{Pr(u \in K : M(u) = o)}{Pr(u \in L : M(u) = o)}.$$

Note that $P_N(o)$ is the probability of nonfault traces conditioned by the fact that ambiguity can only arise from the system traces that produce the observation o. In order to recursively compute P_N we proceed as follows. For a given refined plant G^R whose state space is partitioned into nonfault states versus fault states, we define a nonfault indication binary column vector $I_{nf} \in \{0,1\}^{|Y|\times 1}$, where an entry of 1 indicates a nonfault state. Also define state distribution vector $\boldsymbol{\pi} : M(L) \to [0,1]^{1\times|Y|}$, i.e., for each $o \in M(L)$, $\boldsymbol{\pi}(o)$ is the state distribution of G^R

following the observation o. Then $\boldsymbol{\pi}(\cdot)$ is recursively computed as follows: $\boldsymbol{\pi}(\epsilon) = [1, 0, \dots, 0]$, and for any $o \in M(L), \delta \in \Delta$,

$$\boldsymbol{\pi}(o\delta) \quad = \quad \frac{\boldsymbol{\pi}(o)\boldsymbol{\mu}(\delta)}{||\boldsymbol{\pi}(o)\boldsymbol{\mu}(\delta)||},$$

where $\boldsymbol{\mu}(\delta)$ is computed by solving matrix equations (2.1), and $\| \cdot \|$ is simply the sum of all vector elements. Then for an observation o, $P_N(o)$ is simply given by

$$P_N(o) \quad = \quad \boldsymbol{\pi}(o)I_{nf},$$

where note that $\boldsymbol{\pi}(o)$ and hence also $P_N(o)$ are recursively computed.

Example 4. In the system of Fig. 2.3, the indication vector is given as

$$I_{nf} \quad = \quad [1, 1, 1, 0]^T,$$

and the state distribution vector is initialized as:

$$\boldsymbol{\pi}(\epsilon) \quad = \quad [1, 0, 0, 0].$$

If $o = aba$, then $P_N(o) = 0.783$; if $o = ababc$, then $P_N(o) = 1$; if $o = abaa$, then $P_N(o) = 0$. □

3.1.2 Online Detection Scheme

For issuing online detection decision, we propose a detector, $D : M(L) \to \{F, \epsilon\}$ that for each observation in $M(L)$ issues either a "fault (F)" decision or "no-decision (ϵ)" by comparing the likelihood of no-fault to a suitable threshold, as follows:

$$\forall o \in M(L), [D(o) = F] \Leftrightarrow [\exists \overline{o} \leq o : P_N(\overline{o}) \leq \rho_D], \tag{3.1}$$

where ρ_D is the detection threshold, appropriately chosen to meet the desired FA rate requirement. Note by definition, if a detection decision is F, then it remains F for all future observations, i.e., the detector "does not change its mind", which is expected for the case of permanent faults.

Remark 3. For given detector parameters, the detection scheme (3.1) requires solving (2.1) offline for each $\delta \in \Delta$, and computing online the likelihood of no-fault upon the arrival of a new observation. The former has the complexity of $O(|\Delta| \times |X|^3 \times |\overline{Q}|^3 + |\Sigma| \times |X|^2 \times |\overline{Q}|^2) \leq O(|\Sigma| \times |X|^3 \times |\overline{Q}|^3)$, whereas the latter requires an $O(|X|^2 \times |\overline{Q}|^2)$ complexity. Since (2.1) can be solved offline before the initialization of the online monitoring, the online detection has a quadratic complexity. □

Note a *false alarm* occurs if the detector D issues F while the refined plant is in a nonfault state; and dually a *missed detection* occurs if the detector D fails to issue a F decision within an appropriate delay bound n_D after the occurrence of a fault. In other words, letting P_D^{md} and P_D^{fa} denote the MD and FA rates respectively of a detector D, then

$$P_D^{md} := Pr(st \in L - K : s \in L - K, |t| \geq n_D, P_N(M(st)) > \rho_D), \tag{3.2}$$

$$P_D^{fa} := Pr(s \in K : P_N(M(s)) \leq \rho_D). \tag{3.3}$$

Example 5. For the refined plant of Fig. 2.3 which is S-Diagnosable, suppose we set the threshold $\rho_D = 0.8$. Then any nonfault trace in $a(bc^+a)^*ba \subset K$ will be false-alarmed ($P_N(ababa) = 0.783 < \rho_D$), and thus,

$$P_D^{fa}|_{\rho_D=0.8} = Pr(u \in a(bc^+a)^*ba) = 47.37\%.$$

On the other hand if we set $\rho_D = 0.5$, then any nonfault trace in $a(bc^+a)^*baba \subset K$ will be false-alarmed ($P_N(ababa) = 0.488 < \rho_D$), and thus,

$$P_D^{fa}|_{\rho_D=0.5} = Pr(u \in a(bc^+a)^*baba) = 4.26\%.$$

Now supposing that 4.26% FA rate is acceptable, we fix the detection threshold ρ_D to 0.5. If the detection delay bound is set to be $n_D = 3$, then any fault trace $s \in a(bc^+a)^*fbab \in L - K$ will be miss-detected and thus the MD rate is given by:

$$P_D^{md}|_{\rho_D=0.5,n_D=3} = 6.58\%.$$

On the other hand if the detection delay bound is set to be $n_D = 4$, then any fault trace $s \in L - K$ could be detected, i.e.,

$$P_D^{md}|_{\rho_D=0.5,n_D=4} = 0.$$

□

3.1.3 Existence Condition

We begin by establishing in the following theorem, a property of non-p-equivalent irreducible automata, that for any ambiguity level ρ and tolerance level τ, there must exist a bound n such that the set of traces, of the first automaton, that are longer than the bound and are ambiguous with the traces of the second automaton with ambiguity level higher than ρ, occur with probability lower than τ. Note that s_1 (resp. s_2) denotes a trace generated in A_1 (resp. A_2). The proof is given in Appendix A.

Theorem 1. Given two irreducible finite-state automata A_1 and A_2, where their initial state distribution is the same as their stationary state distribution, if A_1 and A_2 are not p-equivalent, then

$$(\forall \tau > 0 \wedge \forall \rho > 0)(\exists n \in \mathbb{N}) Pr(s_1 : |s_1| > n, Pr(s_2 | M(s_1) = M(s_2)) > \rho) < \tau.$$

Following we present a new characterization of S-Diagnosability which states that the S-Diagnosability is lost if and only if there exists an indistinguishable pair of fault and nonfault traces such that all future observations have identical probability of being fault versus nonfault. The correctness proof is given in the Appendix A.

Theorem 2. (L, K) is not S-Diagnosable if and only if:

$$\begin{aligned}&(\exists s \in L - K, s' \in K \text{ s.t. } M(s) = M(s'))(\forall o \in \Delta^*)\\ &\qquad Pr(t : t \in L \backslash s, M(t) = o) = Pr(t : t \in K \backslash s', M(t) = o).\end{aligned} \tag{3.4}$$

Remark 4. While the definition of S-Diagnosability applies to the set of fault traces $L - K$, Theorem 2 is symmetric with respect to fault and nonfault traces, and thus suggests that notion of diagnosability can also be defined for nonfault traces: $s \in K$ is not diagnosable if and only if there exists $s' \in L - K \cap M^{-1}M(s)$ such that for all future observations $o \in \Delta^*$, $Pr(M^{-1}(o) \cap K \backslash s) = Pr(M^{-1}(o) \cap L \backslash s')$. We denote the set of all non-diagnosable nonfault traces as $K^{nd} \subseteq K$. Clearly, for a S-Diagnosable system, $K^{nd} = \emptyset$. □

Now we are ready to show the main result of this section, which provides insight into the significance of the S-Diagnosability property for the purpose of online fault detection, by showing its necessity and sufficiency for the existence of an online detector that can achieve any desired levels of MD and FA rates.

Theorem 3. (L, K) is S-Diagnosable if and only if for any FA rate requirement $\phi > 0$ and MD rate requirement $\tau > 0$, there exist a detection threshold $\rho_D > 0$ and a delay bound n_D such that $P_D^{fa} \leq \phi$ and $P_D^{md} \leq \tau$.

Proof. (Sufficiency) For a S-Diagnosable system (L, K), we need to show the existence of ρ_D and n_D for achieving given ϕ and τ.

For finding ρ_D, first we partition the set of nonfault traces into three sub-languages, i.e., $K = K_1 \cup K_2 \cup K_3$, where K_1 is the set possessing a fault extension ($K_1 = K \cap pr(L - K)$), K_2 is the set with no fault extension and is non-diagnosable ($K_2 = K^{nd}$), and $K_3 = K - K_1 - K_2$ is the set with no fault extension and is diagnosable. Note if (L, K) is diagnosable, then $K_2 = K^{nd} = \emptyset$.

For the nonfault traces in $K_1 = K \cap pr(L - K)$ that possess a fault extension, nonfaulty-ness is a transient property, and so for any $\phi_1 > 0$ there exists $m_1 \in \mathbb{N}$ such that the traces in K_1 that are longer than m_1 occur with probability smaller than ϕ_1. Denote $\rho_1 = \min_{s \in K_1, |s| \leq m_1} P_N(s)$. Since for a nonfault trace s, $P_N(s) > 0$, and since the traces of length smaller than m_1 are finite, $\rho_1 > 0$. By choosing $\rho_D < \rho_1$ we can ensure that the detector issues a decision for only the traces in K_1 that are longer than m_1. (For shorter traces, P_N value will be larger than $\rho_1 > \rho_D$, and so no decision.) Since the probability of such traces is smaller than ϕ_1, their FA rate is also smaller than ϕ_1.

For the nonfault traces in K_2 that possess no fault extensions and are non-diagnosable, there exists $m_2 \in \mathbb{N}$ such that for every trace in K_2 that is longer than m_2, further extensions will not change the P_N value (i.e., P_N will converge to a value smaller than 1; otherwise the traces would be diagnosable). Denote $\rho_2 = \min_{s \in K_2, |s| \leq m_2} P_N(s)$. Similar to ρ_1, we have $\rho_2 > 0$. By choosing $\rho_D < \rho_2$ we can ensure the detector issues no decision for traces in K_2 and hence no false alarm in K_2.

For the nonfault traces in K_3 that possess no fault extensions and are diagnosable, according to Theorem 1, for any $\phi_3 > 0$ and $\rho_3' \in (0, 1)$ there exists $m_3 \in \mathbb{N}$ such that the traces longer than m_3 and having P_N value smaller than ρ_3' occur with probability smaller than ϕ_3. Denote $\rho_3'' = \min_{s \in K_3, |s| \leq m_3} P_N(s)$. Similar to ρ_1 and ρ_2, we have $\rho_3'' > 0$. By choosing $\rho_D < \rho_3 = \min(\rho_3', \rho_3'')$ we can ensure that the detector issues a decision only for those traces in K_3 that are longer than m_3 and have P_N value smaller than $\rho_D < \rho_3'$. Since the probability of such traces is smaller than ϕ_3, their FA rate is smaller than ϕ_3.

Therefore for any system (*regardless* whether or not it is S-Diagnosable), if we choose ϕ_1 and ϕ_3 in such a way that $\phi_1 + \phi_3 \leq \phi$ and accordingly set $\rho_D = \min_{i=\{1,2,3\}} \rho_i$, then the overall FA rate will be given by:

$$P_D^{fa} \leq \phi_1 + \phi_3 \leq \phi.$$

Thus using our detection scheme, any FA rate can be achieved for any system (regardless of whether or not it is S-Diagnosable), while as will be seen later, this is not the case for the MD rate.

Next we need to establish the existence of n_D to meet the MD rate requirement. Since the system is S-Diagnosable, for any $\tau > 0$ and $\rho_D > 0$ that guarantee FA rate, there always exists $n_D \in \mathbb{N}$ such that $\forall s \in L - K$,

$$Pr(t : t \in L \backslash s, |t| \geq n_D, P_N(st) > \rho_D) < \tau. \tag{3.5}$$

With such a choice of n_D we have, $P_D^{md}(s) < \tau$, and so the overall MD rate is bounded by:

$$P_D^{md} = \sum_{s \in L-K} Pr_D^{md}(s) Pr(s) < \tau Pr(L-K) \leq \tau.$$

Thus the sufficiency of Theorem 3 holds.

(Necessity) Suppose for a system (L, K), given any $\phi > 0$ and $\tau > 0$, there exist ρ_D and n_D such that $P_D^{fa} \leq \phi$ and $P_D^{md} \leq \tau$. Letting $S_D^{md} = \{st : s \in L - K, t \in L \backslash s, |t| \geq n_D, P_N(st) > \rho_D\} \subseteq L - K$ denote the set of fault traces that are miss-detected, we have $P_D^{md} = Pr(S_D^{md}) < \tau$. Then for given $s \in S_D^{md} \subseteq L - K$, we have

$$Pr(st : t \in L \backslash s, |t| \geq n_D, P_N(st) > \rho_D) < \tau.$$

Since the LHS is the same as $Pr(s)Pr(t : t \in L\backslash s, |t| \geq n_D, P_N(st) > \rho_D)$, for any $s \in S_D^{md}$, we have:

$$Pr(t : t \in L\backslash s, |t| \geq n_D, P_N(st) > \rho_D) < \frac{\tau}{Pr(s)}.$$

Let $p = \min_{s \in S_D^{md}} Pr(s)$ and $\tau' = \tau/p$, then for any $s \in S_D^{md}$, we have

$$Pr(t : t \in L\backslash s, |t| \geq n_D, P_N(st) > \rho_D) < \tau'.$$

Note τ can be chosen to be arbitrarily small to make τ' arbitrarily small. Furthermore for any $s \in (L - K) - S_D^{md}$, we have:

$$Pr(t : t \in L\backslash s, |t| \geq n_D, P_N(st) > \rho_D) = 0 < \tau'.$$

Then $\forall s \in L - K$,

$$Pr(t : t \in L\backslash s, |t| \geq n_D, P_N(st) > \rho_D) < \tau'. \tag{3.6}$$

Since for any $\phi > 0$ (and hence any ρ_D) and $\tau > 0$ (and hence any $\tau' > 0$), such n_D always exists to make the above analysis true, then for any $\rho_D > 0$ and $\tau' > 0$, $\exists n_D \in \mathbb{N}$ such that (3.6) holds, which indicates the condition for S-Diagnosability is held. Thus the necessity of Theorem 3 holds. □

3.2 Computation of Detection Threshold and Delay

In previous section we established that S-Diagnosability is a necessary and sufficient condition for the existence of a detection threshold ρ_D and a detection delay bound n_D to achieve any desired level of FA and MD rates. In the this section we provide algorithms for computing the parameters ρ_D and n_D so as to achieve the desired level of MD and FA rates.

3.2.1 Algorithms for ρ_D and n_D

Given a S-Diagnosable system G^R and FA and MD rates requirements ϕ and τ, we provide the computation of detection threshold ρ_D and delay bound n_D so that $P_D^{fa} \leq \phi$ and $P_D^{md} \leq \tau$. In order to compute detection threshold ρ_D for a given FA rate requirement ϕ, Algorithm

1 constructs an "extended observer tree", that for each observation sequence, estimates the states (as any observer does), and organizes it in a tree form where nodes are observations tagged with the estimated states and the edges are transitions on a next new observation, with the extension that each state in the estimate is labeled by the probability of reaching it. The construction of Algorithm 1 makes the "extended observation tree" formal. These probability labels are then used to compute the probability P_N for each observation, or equivalently, each node of the extended observer tree. The tree extends to a depth so that if no detection decision are made for any of the nodes (equivalently, corresponding observations) in the tree, then the FA rate caused by the detection decisions at the future successors is upper bounded by the desired rate ϕ. The existence of such a depth is guaranteed by Theorem 4, and to ensure no detection decision for any of the nodes in T, we simply choose the detection threshold to be smaller than the minimum P_N value among all nodes of T (recall by (3.1) that a detection decision is only issued when the P_N value falls below the threshold).

Algorithm 1. For a given refined plant G^R and a FA rate requirement ϕ, do the following:

1. This step is just a preparatory step to identify certain classes of states before beginning to construct an extended observer tree. Identify all the states in $X \times Q$ from which a fault state in G^R is reachable, and denote this set of states as Y_1 (these are nonfault states from where fault states are reachable, and correspond to states reached by traces in K_1 defined in the proof of Theorem 3). Identify $Y_{2,3} = X \times Q - Y_1$ (these are nonfault states reached by traces in $K_2 \cup K_3$ defined in the proof of Theorem 3).

2. Iteratively construct an extended observer tree T with set of nodes, $\overline{Z} = Z \times M(L)$, where $Z = 2^{((X \times \overline{Q}) \times (0,1])}$, and the depth of tree grows by 1 in each iteration until the stopping criterion is satisfied—see below. Then each node of T is of the form $\overline{z} = (z, o(\overline{z}))$, where $o(\overline{z}) \in M(L)$ is a unique observation associated with the node $\overline{z}$ and $z = \{((x_i, \overline{q}_i), p_i)\} \subseteq (X \times \overline{Q}) \times (0,1]$ is set of state estimates, with the ith one denotes $(x_i, \overline{q}_i)$, tagged with its occurrence probability p_i. The tree T is rooted at $\overline{z}_0 = \{((0,0),1), \epsilon\}$. $\overline{z}_2 \in \overline{Z}$ is a δ-child ($\delta \in \Delta = M(\Sigma) - \{\epsilon\}$) of $\overline{z}_1 \in \overline{Z}$ if and only if $o(\overline{z}_2) = o(\overline{z}_1)\delta$ and for every $((x_2, \overline{q}_2), p_2) \in z_2$, it holds that $p_2 = \sum_{((x_1,\overline{q}_1),p_1)\in z_1} \sum_{s\in\Sigma^*:M(s)=\delta} p_1 \times \gamma((x_1,\overline{q}_1), s, (x_2,\overline{q}_2))$. *It can be*

seen that $((x_2, \overline{q}_2), p_2)$ is included in z_2 if and only if $(x_2, \overline{q}_2)$ can be reached from a state included in z_1 following extra observation δ and p_2 is the probability of reaching $(x_2, \overline{q}_2)$ from initial state following the observation $o(\overline{z}_2)$.

Using the probability values of states in any node $\overline{z}$ of T, we can compute the likelihood of no-fault following the observation $o(\overline{z})$, by way of adding the probabilities of the nonfault states of the node, and next normalizing over all states of the node as follows:

$$\forall \overline{z} = (z, o(\overline{z})) : \qquad P_N(\overline{z}) := \frac{\sum_{((x,\overline{q}),p)\in\overline{z},\overline{q}\neq F} p}{\sum_{((x,\overline{q}),p)\in\overline{z}} p}.$$

Then $P_N(\overline{z})$ equals $P_N(o(\overline{z}))$, and corresponds to the conditional probability of no-fault given the observation $o(\overline{z})$.

Terminate the tree at a uniform depth so the set of leaf nodes $\overline{Z}_m \subseteq \overline{Z}$ satisfy:

- $(\overline{z}, \overline{z}' \in \overline{Z}_m) \Rightarrow (|o(\overline{z})| = |o(\overline{z}')| =: d_1)$ (each terminal node is reached after the same number of observations, which guarantees the uniformity of the depth of T, which we denote as d_1), and
- $\sum_{\overline{z}\in\overline{Z}_m} \sum_{((x,\overline{q}),p)\in\overline{z}:(x,\overline{q})\in Y_1} p + \sum_{\overline{z}\in\overline{Z}_m:P_N(\overline{z})\leq\rho_{\min}} \sum_{((x,\overline{q}),p)\in\overline{z}:(x,\overline{q})\in Y_{2,3}} p < \phi$, where $\rho_{\min} := \min_{\overline{z}\in\overline{Z}:P_N(\overline{z})\neq 0} P_N(\overline{z})$ (for states in Y_1 contained in terminal nodes, their added probabilities of the first term equals $Pr(K_1 \cap M^{-1}(\Delta^{>d_1}))$, which upper bounds the FA rate of their successors (see proof of Theorem 3); for the states in $Y_{2,3}$ contained in the terminal nodes having $P_N \leq \rho_{\min}$, their added probabilities of the second term equals $Pr(s \in [K_2 \cup K_3] \cap M^{-1}(\Delta^{>d_1}) : P_N(M(s)) \leq \rho_{\min})$, which upper bounds the FA rate of their successors (see proof of Theorem 3); we require the combined upper bounds to be less that ϕ, which ensures that even if all successors produce false alarm, the FA rate requirement is still met).

3. Return any $\rho_D < \rho_{\min}$. (Note that with this choice of ρ_D, all nonfault traces whose observations are included in T will have no detection decisions (and so no false alarms either), and only their extensions can have detection decisions (some of which may be false alarms). But by construction, the probability of those extensions is upper bounded by ϕ, as desired.)

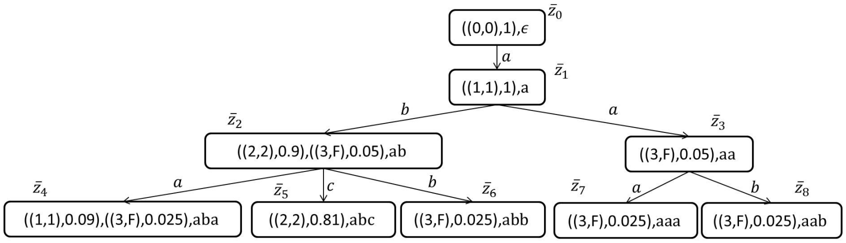


Figure 3.1 Part of an extended observer tree for Example 6.

The following theorem guarantees the correctness of Algorithm 1. Correctness proof is given in the Appendix A.

Theorem 4. There exists $d_1 \in \mathbb{N}$ such that Algorithm 1 terminates with tree depth d_1 and returns a threshold ρ_D under which the overall FA rate is upper bounded by ϕ.

Note as the tree depth is increased, the set of traces contained in the tree, and hence their probability, also grows. Since no detection decision is issued for traces in the tree, they don't incur any false alarms, and hence the false alarm rate is upper bounded by the probability of traces not included in the tree. By increasing the tree depth, we can essentially guarantee that this upper bound is as small as desired.

Example 6. For the system G^R shown in Fig. 2.3, $Y_1 = \{(0,0),(1,1),(2,2)\}$ and $Y_{2,3} = \emptyset$. We construct the extended observer tree for the computation of detection threshold; the first 4 steps of which are as shown in Fig. 3.1, where $P_N(\overline{z}_0) = P_N(\overline{z}_1) = 1$, $P_N(\overline{z}_2) = 0.9474$, $P_N(\overline{z}_3) = 0$, $P_N(\overline{z}_4) = 0.7826$, $P_N(\overline{z}_5) = 1$, $P_N(\overline{z}_6) = P_N(\overline{z}_7) = P_N(\overline{z}_8) = 0$. Selecting any $\rho_D < \min_{\overline{z}\in\overline{Z}:P_N(\overline{z})\neq 0} P_N(\overline{z}) = 0.7826$, the FA rate is upper bounded by $\sum_{\overline{z}\in\overline{Z}_m} \sum_{((x,\overline{q}),p)\in\overline{z}:(x,\overline{q})\in Y_1} p = 0.09 + 0.81 = 0.9$. Algorithm 1 would proceed to a next step unless this FA rate is found to be acceptable. □

Having provided an algorithm to compute the detection threshold ρ_D that meets the FA rate requirement ϕ, we next present an algorithm to compute the delay bound n_D to satisfy the given MD rate requirement τ. Here we provide a brief outline of the algorithm: In order to compute delay bound n_D, Algorithm 2 constructs a refined version of the extended observer tree that for each observation sequence estimates the states and their probabilities, *with the* *refinement that keeps track of the number of post fault transitions executed for each state in the estimated state set.* The tree extends to a depth so that if no missed detections occur for any of the nodes in the tree, then the MD rate caused by the future successors is upper bounded by the desired rate τ. For S-Diagnosable systems, the existence of such a depth is guaranteed by Theorem 5, and to ensure no missed detection for any of the nodes in T, we simply choose n_D to be greater than the maximum number of post fault transitions among all nodes of T.

Algorithm 2. For a given refined plant G^R, a detection threshold ρ_D and a MD rate requirement τ, do the following:

1. Iteratively construct a refined extended observer tree T with set of nodes, $\overline{Z} = Z \times M(L)$, where $Z = 2^{((X\times\overline{Q})\times(0,1]\times\mathbb{N})}$ ($\mathbb{N} = \{0, 1, 2, \dots\}$), and the depth of T grows by 1 in each iteration until the stopping criterion is satisfied—see below. Similar to Algorithm 1, each node of T is of the form $\overline{z} = (z, o(\overline{z}))$, where $z = \{((x_i, \overline{q}_i), p_i, n_i)\} \subseteq (X \times \overline{Q}) \times (0, 1] \times \mathbb{N}$, $o(\overline{z}) \in M(L)$ and the additional term n_i counts the number of post-fault transitions in reaching $(x_i, \overline{q}_i)$. The tree T is rooted at $\overline{z}_0 = \{((0, 0), 1, 0), \epsilon\}$. $\overline{z}_2 \in \overline{Z}$ is a δ-child ($\delta \in \Delta = M(\Sigma) - \{\epsilon\}$) of $\overline{z}_1 \in \overline{Z}$ if and only if $o(\overline{z}_2) = o(\overline{z}_1)\delta$, and for every $((x_2, \overline{q}_2), p_2, n_2) \in z_2$, it holds that $p_2 = \sum_{((x_1,\overline{q}_1),p_1,n_1)\in z_1}$ $\sum_{s\in\Sigma^*:M(s)=\delta,\#\text{post-fault}(s,(x_1,\overline{q}_1))+n_1=n_2} p_1 \times \gamma((x_1, \overline{q}_1), s, (x_2, \overline{q}_2))$. Here "#post-fault" counts the number of events in s beyond a fault as follows: if $\overline{q}_1 = F$, it returns the value $|s|$, and otherwise it returns the number of transitions executed in s after a fault state is reached. *It can be seen that $((x_2, \overline{q}_2), p_2, n_2)$ is included in z_2 if and only if $(x_2, \overline{q}_2)$ can be reached from a state included in z_1 following extra observation δ, p_2 is the probability of reaching $(x_2, \overline{q}_2)$ from initial state following observation $o(\overline{z}_2)$ and n_2 is the number the post fault transitions executed.*

   For each node $\overline{z} = (z, o(\overline{z}))$, define the likelihood of no-fault given the observation $o(\overline{z})$ as in Algorithm 1:

$$P_N(\overline{z}) \quad := \quad \frac{\sum_{((x,\overline{q}),p,n)\in z,\overline{q}\neq F} p}{\sum_{((x,\overline{q}),p,n)\in z} p}.$$

   Terminate a branch of the tree if a detection decision has been made (P_N value smaller than ρ_D), and terminate the rest of the tree at a uniform depth so the set of leaf nodes

$\overline{Z}_m \subseteq \overline{Z}$ satisfy:

- $P_N(\overline{z}) \leq \rho_D$ (for these nodes detection decision can be issued, implying these nodes will have no missed detections), or
- $\sum_{\overline{z}\in\overline{Z}_m:P_N(\overline{z})>\rho_D}\sum_{((x,\overline{q}),p,n)\in\overline{z}:(x,\overline{q})\in Y_1\vee\overline{q}=F} p < \tau$ (for these nodes, no detection decision will be issued since $P_N(\overline{z}) > \rho_D$, and by the choice of n_D in step 2 below there is no missed detection yet; so their added probabilities upper bounds the MD rate of their future successors, and the stopping criterion requires this to be below the desired value τ).

2. Return any $n_D > \max_{((x,\overline{q}),p,n)\in z,\overline{z}\in\overline{Z}} n$, and let d_2 denote the depth of tree T. Note that with this choice of n_D all fault traces, whose observations are included in T, are not missed detection. So clearly that the MD rate P_D^{md} is upper bounded by $\overline{P_D^{md}}$ given by:

$$\overline{P_D^{md}} := \sum_{\overline{z}\in\overline{Z}_m:P_N(\overline{z})>\rho_D}\;\sum_{((x,\overline{q}),p,n)\in\overline{z}:(x,\overline{q})\in Y_1\vee\overline{q}=F} p. \tag{3.7}$$

The following theorem guarantees the correctness of Algorithm 2. Correctness proof is given in the Appendix A.

Theorem 5. For S-Diagnosable systems, there exists $d_2 \in \mathbb{N}$ such that Algorithm 2 terminates with tree depth d_2 and returns a delay bound n_D under which the overall MD rate is upper bounded by τ.

Note as before, as the tree depth is increased, the set of traces contained in the tree, and hence their probability, also grows. For all traces included in the tree, S-Diagnosability guarantees that a correct detection decision is issued within a bounded delay bound, and so any missed detection can only occur for those traces not included in the tree. So the MD rate is upper bounded by the probability of traces not included in the tree. By increasing the tree depth, we can essentially guarantee that this upper bound is as small as desired, and then read the detection delay of the traces included in the tree for which detection decision is made (i.e., whose P_N values are smaller than the detection threshold).

Example 7. For the system G^R in Fig. 2.3, and assuming detection threshold of $\rho_D = 0.7825$ as determined in Example 6, we construct the refined extended observer tree for the computation

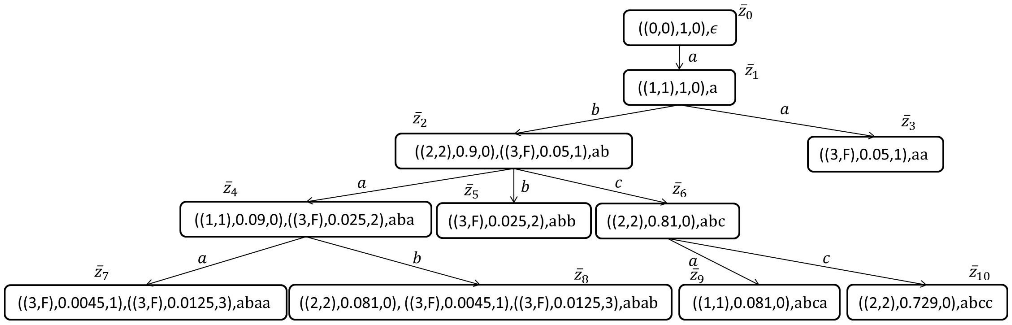


Figure 3.2 Part of a refined extended observer tree for Example 7.

of delay bound; the first 5 steps of which are as shown in Fig. 3.2. Here $P_N(\overline{z}_0) = P_N(\overline{z}_1) = 1$, $P_N(\overline{z}_2) = 0.9474$, $P_N(\overline{z}_3) = 0$, $P_N(\overline{z}_4) = 0.7826$, $P_N(\overline{z}_5) = 0$, $P_N(\overline{z}_6) = 1$, $P_N(\overline{z}_7) = 0$, $P_N(\overline{z}_8) = 0.8265$ and $P_N(\overline{z}_9) = P_N(\overline{z}_{10}) = 1$. The branches of $\overline{z}_3$ and $\overline{z}_5$ terminate since the likelihood of no-fault is smaller than $\rho_D = 0.7826$, whereas the depth of the rest of the tree is 5. With $n_D = 1 + \max_{((x,\overline{q}),p,n)\in z, \overline{z}\in\overline{Z}} n = 4$, the MD rate is upper bounded by $\overline{P_D^{md}} = \sum_{\overline{z}\in\{\overline{z}_8,\overline{z}_9,\overline{z}_{10}\}} \sum_{((x,\overline{q}),p,n)\in\overline{z}:(x,\overline{q})\in Y_1} p = 0.081+0.0045+0.0125+0.081+0.729 = 0.908$. Algorithm 2 would proceed to a next step unless this MD rate is found to be acceptable. □

Remark 5. Both Algorithm 1 and Algorithm 2 require the construction of an extended observer (with depths d_1 and d_2 and branching degree at most $|\Delta|$) that can have $O(|\Delta|^{d_1})$ and $O(|\Delta|^{d_2})$ nodes, respectively, and each node can have up to $|X| \times |\overline{Q}|$ elements. Therefore the complexity for *offline* computation for detection parameters ρ_D and n_D is $O(|X| \times |\overline{Q}| \times |\Delta|^d)$, where $d = \max\{d_1, d_2\}$. Note that d can depend on the system and specification models, the observation mask, and the desired bounds on MD and FA rates, and is bounded. On the other hand, as mentioned in Remark 3, the complexity of *online* monitoring is quadratic, $O(|X|^2 \times |\overline{Q}|^2)$. □

3.2.2 Non-S-Diagnosable Systems

In the absence of S-Diagnosability, the termination of Algorithm 2 is not guaranteed, but a slight modification yields a terminating algorithm that finds an upper bound for the minimum achievable MD rate. In the case when the system is not S-Diagnosable, then (3.5) in the proof for Theorem 3 may not hold for some $s \in L - K$. For given ϕ and τ, let ρ_D be chosen so that $P_D^{fa} \leq \phi$, and let $S_D^{nd} \subseteq L - K$ be the set of non-diagnosable fault traces for which there exists a MD rate $\tau' > 0$ such that the condition $Pr_D^{md}(S_D^{nd}) = Pr(st : s \in S_D^{nd}, t \in L\backslash s, |t| \geq n_D, P_N(st) > \rho_D) < \tau'$ is not satisfied by any $n_D \in \mathbb{N}$. Then for the traces in $(L - K) - S_D^{nd}$ there exists a detection delay bound n_D so that $\forall s \in (L - K) - S_D^{nd}$,

$$Pr(t : t \in L\backslash s, |t| \geq n_D, P_N(st) > \rho_D) < \tau',$$

and so the overall MD rate is upper bounded by:

$$P_D^{md} = \sum_{s \in L-K} Pr_D^{md}(s)Pr(s) < \tau' Pr(L - K - S_D^{nd}) + Pr_D^{md}(S_D^{nd}) \leq \tau' + Pr_D^{md}(S_D^{nd}).$$

Thus for non-S-Diagnosable systems, while any desired FA rate $\phi > 0$ can be always achieved by an appropriate choice of $\rho_D > 0$, a MD rate $\tau > 0$ can only be achieved if $\tau' + Pr_D^{md}(S_D^{nd}) \leq \tau$. Since n_D can be chosen to make τ' arbitrarily small, a MD rate $\tau > 0$ can be achieved if and only if $Pr_D^{md}(S_D^{nd}) < \tau$. This is captured in the following theorem, which generalizes Theorem 3 to the case of non-S-Diagnosable systems.

Theorem 6. Given a stochastic, nonfault specification-refined plant G^R with generated language L and nonfault behavior K, FA rate requirement $\phi > 0$ and MD rate requirement $\tau > 0$, there exists a detection threshold $\rho_D > 0$ such that $P_D^{fa} \leq \phi$, and for this detection threshold there exists a detection delay bound n_D such that $P_D^{md} \leq \tau$ if and only if $Pr_D^{md}(S_D^{nd}) \leq \tau$, where $S_D^{nd} \subseteq L - K$ is the set of non-diagnosable fault traces for which there exists $\tau' > 0$ such that the condition $Pr(st : s \in S_D^{nd}, t \in L\backslash s, |t| \geq n_D, P_N(st) > \rho_D) < \tau'$ is not satisfied by any $n_D \in \mathbb{N}$.

Remark 6. For a fixed FA rate, $Pr_D^{md}(S_D^{nd})$ is also fixed and serves as a lower bound for MD rate that the detection scheme can achieve. Note that $Pr_D^{md}(S_D^{nd})$ is a function of the FA rate requirement ϕ. When ϕ is made tighter by decreasing it, a smaller ρ_D is needed, and the resulting non-diagnosable fault traces subsume those corresponding to larger ρ_D. Therefore the minimum achievable MD rate increases as FA rate is made stringent by decreasing it. □

Next we present a variant of Algorithm 2 that for a fixed threshold ρ_D computes an upper bound for $Pr_D^{md}(S_D^{nd})$. Algorithm 3 iteratively builds a refined extended observer tree T, and at each step computes an upper bound for the MD rate that either decreases or remains constant from one iteration to the next. When the latter happens, a future iteration may eventually decrease the bound, but since the optimal (least) upper bound is unknown, it is also not known how long one should continue iterating. So, to ensure termination, we adopt the heuristics of terminating the algorithm when the upper bound continues to remain constant while n_D gets doubled.

Algorithm 3. For a given refined plant G^R and a threshold ρ_D, do the following:

1. Iteratively construct a refined extended observer tree T as in the step 1 of Algorithm 2;

2. For each depth of the tree T, set $n_D = 1 + \max_{((x,\overline{q}),p,n)\in z, \overline{z}\in\overline{Z}} n$ and compute an upper bound $\overline{P_D^{md}}$ for MD rate P_D^{md} according to (3.7);

3. If the upper bound $\overline{P_D^{md}}$ doesn't decrease while n_D computed in step 2 gets doubled over any two iteration steps (not necessarily consecutive), stop and return this upper bound.

3.3 Illustrative Example

We consider the problem of leakage detection in a two-tank system as shown in Fig. 3.3, which is adopted from [86]. The tanks are connected with a valve. The water is pumped into the system in the left tank at a constant rate and outflows from the right tank. The only observation produced by this system is the symbolic sensor output (Low, Medium, High) which measures the outflow rate of the right tank at discrete times. There is a 0.05 probability that a leakage occurs in the left tank, which is to be detected. The aforementioned system is described by the stochastic automaton shown in Fig. 3.4(a), where the event set is $\Sigma = \{L, M, H, \text{leak}\}$, corresponding to the sensor outputs and the occurrence of leakage. All events except "leak" are fully observable, whereas "leak" is fully unobservable, i.e., $\Sigma_{uo} = \{\text{leak}\}$. The water levels in the tanks are quantized into "LOW", "MEDIUM" and "HIGH" for the left tank, and just "LOW" and "HIGH" for the right tank, and each state in the stochastic automaton denotes a combination of these water levels along with a record whether a leak occurred in past, summarized in Fig. 3.4(b). The system is initialized at state 2, i.e., medium level of water in the left tank and low level of water in the right tank. The states $\{1, \dots, 6\}$ are pre-fault states

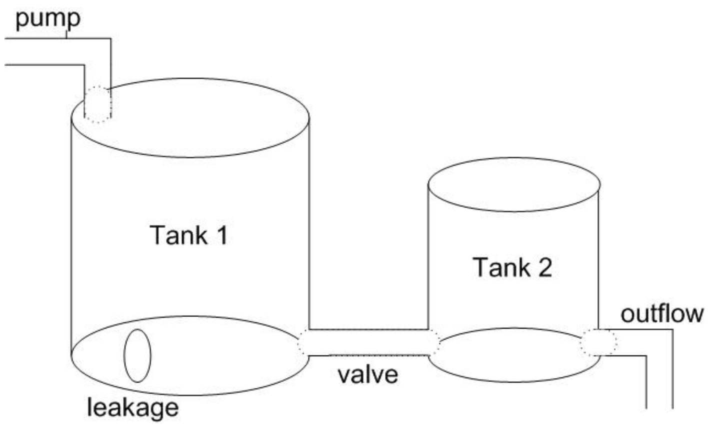
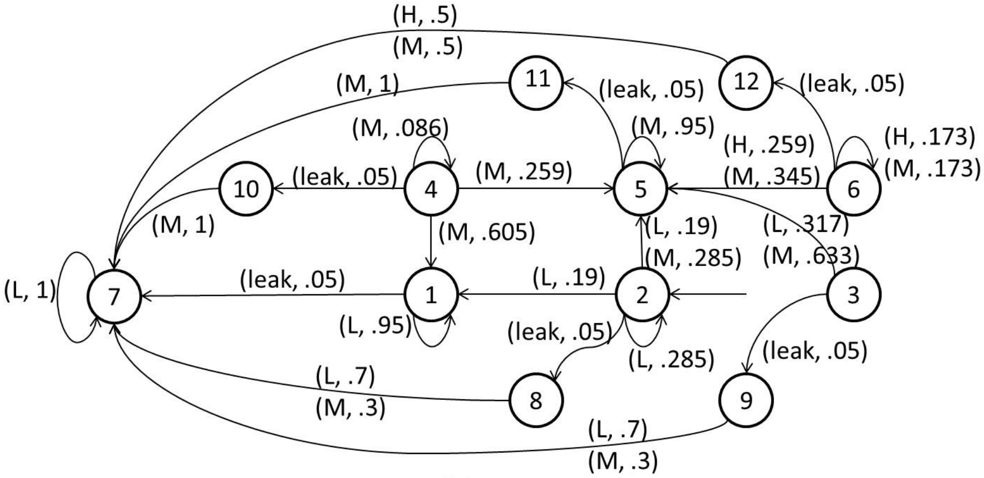


Figure 3.3 Two-tank system.

and states $\{i+6, i=1,\ldots,6\}$ are post-fault states, and so the nonfault specification is simply a subautomaton of the plant automaton restricted to the pre-fault states, and without the probability labels. The possibility of occurrence of leakage at each pre-fault state $i, i=1,\ldots,6$, is captured by the transition from state i to state $i+6$ labeled with the event "leak" and occurrence probability 0.05. The transitions are obtained by way of abstraction, and for further details readers are referred to [86, 87, 88]. It can be checked that the system is S-Diagnosable, so Theorem 3-5 apply.

We implement the proposed Algorithms 1 and 2 to compute the detection threshold ρ_D and delay bound n_D to ensure any given FA and MD rate requirements. The results are shown in Tables 3.1-3.2 and Fig. 3.5. Table 3.1 lists for various FA rates the detection threshold ρ_D returned by Algorithm 1, as well as the tree depth d_1, the number of tree nodes and the running time of the implementation of Algorithm 1 on a standard desktop PC; and the first two columns is plotted in Fig. 3.5(a). For example, when the FA rate is required to be under 5%, the detection threshold returned by Algorithm 1 is $\rho_D = 0.044$. When we fix $\rho_D = 0.044$, i.e., fix $\phi = 5\%$, the delay bound n_D returned by Algorithm 2 for various MD rates is shown in Table 3.2 and Fig. 3.5(b); the table additionally lists for each MD rate the tree depth d_2, the number of tree nodes and the running time of the implementation of Algorithm 2 on a standard desktop PC. As can be seen, when the MD rate is required to be under 5%, the detection delay bound returned by Algorithm 2 is $n_D = 60$. If we wish to decrease the detection delay bound, then



(a)

Nonfault state	Fault state	Water level in left tank	Water level in right tank
1	7	LOW	LOW
2	8	MEDIUM	LOW
3	9	HIGH	LOW
4	10	LOW	HIGH
5	11	MEDIUM	HIGH
6	12	HIGH	HIGH

(b)

Figure 3.4 (a) Stochastic automaton G for the two-tank system shown in Fig. 3.3; (b) interpretation of states.

the upper bound for the MD rate will increase and possibly violate the MD rate requirement of 5%. For example if we choose $n_D = 55$, then it could only be assured that the MD rate is upper bounded by 36.24%. Recall by previous discussion, the delay bound can depend on both FA rate ϕ and MD rate τ, and this dependency is shown in Fig. 3.5(c). This figure along with Fig. 3.5(a) can be used to determine the parameters ρ_D and n_D for the specified FA and MD rates for the two-tank example. It so happens that for $n_D = 59$, the upper bound given by (3.7) is higher than 35%, whereas it suddenly becomes lower than 5% for $n_D = 60$. This sudden drop in upper bound explains the reason why the tree depth saturates at 60 when MD rate is decreased from 35% to 5%.

FA rate	Threshold	Tree depth	# of	Running
ϕ	ρ_D	d_1	nodes	time (sec.)
0.95	0.8717	2	7	0.004
0.9	0.8004	3	14	0.007
0.85	0.7473	4	25	0.016
0.8	0.7051	5	41	0.019
0.75	0.6682	6	63	0.028
0.7	0.6343	7	92	0.048
0.65	0.5722	9	175	0.074
0.6	0.5436	10	231	0.097
0.55	0.4906	12	377	0.158
0.5	0.4428	14	575	0.249
0.45	0.3996	16	833	0.383
0.4	0.3606	18	1159	0.558
0.35	0.3092	21	1793	0.984
0.3	0.2651	24	2625	1.582
0.25	0.2159	28	4089	2.908
0.2	0.1759	32	6017	5.211
0.15	0.1361	37	9177	10.64
0.1	0.0903	45	16261	32.94
0.05	0.0440	59	36050	182.3

Table 3.1 Computational results of Algorithm 1.

3.4 Conclusion

In this Chapter, the problem of online fault diagnosis for stochastic DESs was studied. An online detector based on recursive likelihood computation was proposed, whose existence for achieving any arbitrary performance requirement was shown to be equivalent to the S-Diagnosability property. Algorithms for computing the detector parameters of detection threshold and delay bound so as to achieve a given performance requirement of false alarm and missed detection rates were presented, using a proposed procedure for constructing an extended observer. The extended observer computes, for each observation sequence, the set of states reached in the system model, along with their probabilities and the number of post-fault transitions executed. The algorithms were guaranteed to terminate and upper bounds on the number of iterations prior to termination were provided.

MD rate	Delay	Tree depth	# of	Running
τ	n_D	d_2	nodes	time (sec.)
0.95	4	4	21	0.018
0.9	5	5	31	0.029
0.85	7	7	57	0.052
0.8	9	9	91	0.084
0.75	11	11	133	0.126
0.7	14	14	211	0.217
0.65	16	16	273	0.299
0.6	19	19	381	0.436
0.55	23	23	553	0.688
0.5	28	28	813	1.150
0.45	34	34	1191	1.828
0.4	43	43	1893	3.356
0.35	60	60	3661	8.068
0.3	60	60	3661	8.056
0.25	60	60	3661	8.114
0.2	60	60	3661	8.060
0.15	60	60	3661	8.038
0.1	60	60	3661	8.027
0.05	60	60	3661	8.028

Table 3.2 Computational results of Algorithm 2 with $\rho_D = 0.044$.

The detector has a quadratic complexity for the *online* monitoring, likelihood computation and issuing decision upon the arrival of a new observation, while the *offline* computation of the detector parameters, namely, detection threshold and delay bound requires constructing an extended observer whose size is exponential in the depth of the observer tree constructed, while the depth of the tree is a complex function of the system and specification models, the observation mask, and the desired bounds on MD and FA rates, and is bounded. As can be inferred by the illustrative example in previous sub-section, the detector parameters of detection threshold and delay bound for various levels of MD and FA rates can be computed offline and stored in a database, and during online monitoring and detection the required set of parameters can be looked up each time a new level of MD and FA rates are specified.

It was also shown that our detection strategy works for S-Diagnosable as well as non-S-Diagnosable systems in the same manner. For S-Diagnosable systems it is possible to achieve

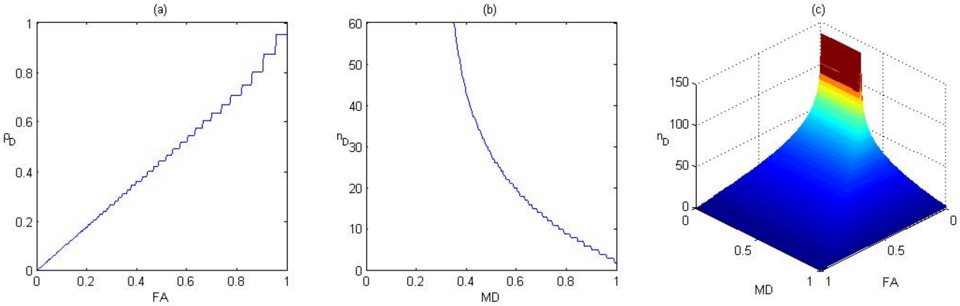


Figure 3.5 Computational results of Algorithms 1 and 2 for leakage detection in two-tank system: (a) the detection threshold ρ_D as a function of ϕ; (b) the delay bound n_D as a function of τ, when $\rho_D = 0.044$ ($\phi = 5\%$); (c) n_D as a function of both ϕ and τ.

arbitrary performance for FA and MD rates, while for a non-S-Diagnosable system an arbitrary performance is achievable only for the FA rate, whereas a lower bound exists for the achievable MD rate that is a function of the FA rate, and increases as FA rate is decreased. A variant of the algorithm for the S-Diagnosable case was used to compute an upper bound for the minimum achievable missed detection rate for a non-S-Diagnosable system.

CHAPTER 4. FAILURE DIAGNOSIS OF HYBRID SYSTEMS

This chapter studies the fault detection of discrete-time stochastic systems with linear-time temporal logic (LTL) as correctness requirement—A fault is a violation of the LTL specification. The temporal logic allows the system correctness properties to be specified compactly and in an user-friendly manner (being close to natural-languages), and supports automatic translation into other formal models such as automata. We introduce the notion of input-output stochastic hybrid automaton (I/O-SHA) and show that a continuous physical system (modeled as stochastic difference equations) when refined against a certain class of LTL correctness requirement, the refinement can be modeled as an I/O-SHA, which preserves the behaviors of the physical system and also captures the requirement-violation as a reachability property. The probability distribution over the discrete locations of the hybrid system is estimated recursively by computing the distributions for continuous variables for each discrete location. This is then used to compute the *likelihood of no-fault*, a statistic that we employ for the purpose of fault detection. The performance of the detection scheme is measured in terms of false alarm (FA) and missed detection (MD) rates, and the condition for the existence of a detector to achieve any desired rates of FA and MD is captured in form of *Stochastic-Diagnosability*, a notion that we introduce for stochastic hybrid systems. The proposed method of fault detection is illustrated by a practical example.

4.1 Problem Formulation

Suppose the dynamics of a physical system G under diagnosis can be described by the discrete-time stochastic difference equations (4.1)-(4.3):

$$x_{k+1} = f(x_k, u_k, v_k) \tag{4.1}$$

$$\begin{aligned} r_k &= g(x_k, u_k) && (4.2) \\ y_k &= h(x_k, u_k, w_k). && (4.3) \end{aligned}$$

where u, x, r, y, v, w represent, respectively, the input, state, requirement (unobserved), output (observed), disturbance and noise variables, and k is the time-index. The initial state x_0, the *disturbance* v_k as well as the *noise* w_k are all assumed mutually i.i.d. with known distributions. Note the requirement variable, which specifies a required value for each input-state pair through the function g, is used to capture a user-defined specification that, at each step, depends on system state and input, and being a user-defined requirement, it is not corrupted by noise. We assume that the properties of the required system behaviors can be described by using a LTL formula ϕ involving *predicates* defined over the requirement variables r_k, $k \in N$. Then the predicates, appearing in the LTL formula, and their boolean combinations act as atomic propositions guarding the transitions in the Büchi automaton. The set of all infinitely long feasible sequences of aforementioned predicates is denoted as A_G.

Since detection of requirement-violation must occur based on a finite history of input/output observations, it is natural to assume that every infinite run of a system, that violates the given LTL formula, possesses a finite prefix, called an *indicator*, such that all its infinite extensions that are feasible in the system also violate the LTL formula. This property was captured under the name of *prediagnosability* in [49], and is a *necessary* condition for any detector's ability to detect the violation of the specified LTL formula based on finite ength observations. So, without loss of generality, we assume that the prediagnosability holds. Next we provide a formal definition of indicator and also of prediagnosability.

Definition 5. Given a system G and a LTL formula ϕ, a finite sequence of requirement variables is said to be an *indicator* if all of its infinite extensions in G violate ϕ. We denote the set of all indicators as $I_\phi(G)$. G is said to be *prediagnosable* with respect to ϕ if each infinite sequence of requirement variables violating ϕ possesses a finite prefix that is an indicator.

Remark 7. By utilizing the notion of indicator, detecting the occurrence of infinite trace violating a LTL formula is transformed into detecting the execution of finite indicators. As mentioned in [49], when an indicator is executed, the actual fault may not have happened yet. Hence, our framework includes both cases of fault detection (that a fault has already occurred) and prediction (that a fault will inevitably occur). Note that the notion of indicator has also been utilized for the purpose of fault prognosis (see for example [75, 77]), where the prediction of a future fault is performed by detecting the occurrence of a nonfault prefix indicator. □

Remark 8. Note that a system is automatically prediagnosable if the correctness requirement ϕ is a safety one [81], i.e., it only requires that some "bad" things must never occur. However, when the correctness requirement is a more general one, the system may not be prediagnosable (See Example 8), in which case, the violation of ϕ can not be detected even if the system is perfectly observable, i.e., $y_k = r_k$ for all $k \in \mathbb{N}$. By this reason, we assume without loss of generality that the system is prediagnosable with respect to the LTL formula. □

As established in [49, Theorem 1], the prediagnosability of system G with respect to a LTL formula ϕ, is equivalent to the existence of a deterministic Büchi automaton accepting $S_\phi \cap A_G$, which can also be characterized as the *limits* of the finite prefixes accepted by the same model treated as a standard finite state automaton. Then we can augment the Büchi automaton, by adding an absorbing state called F reaching which indicates the execution of an indicator, to yield an augmented deterministic requirement model, denoted as R. (Note the augmentation requires adding the "missing" transitions from each state to the newly added fault state F, guarded by the complement of the existing transitions of the state.)

Example 8. Consider a system G with dynamics:

$$\begin{aligned} x_{k+1} &= x_k + v_k \\ r_k &= 2x_k - 1 \end{aligned}$$

where v_k is i.i.d. Gaussian random variable. Suppose the LTL formula is given as $\phi = GF(r < 0)$, i.e., it is always (G) possible that in future (F), the requirement variable becomes negative. Then it can be verified (see Fig. 4.1(a)) that for any infinite sequence $(r_0, r_1, \ldots, r_m, \ldots)$ with $r_i \geq 0, \forall i \geq m$ (i.e., a sequence violating ϕ), any of its prefix has certain infinite extension in which $(r_k < 0)$ is satisfied for infinitely many k (i.e., a sequence satisfying ϕ). Therefore G is not prediagnosable with respect to ϕ. In this case even with perfect observation $y_k = r_k$, the violation of ϕ cannot be detected.

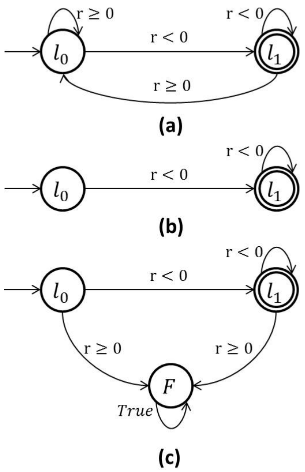


Figure 4.1 The specification model R for Example 8.

Now consider the disturbance to be $v_k = sign(x_k)v'_k$, where v'_k is a positive-valued random variable, i.e., the noise v_k is dependent on the state variable x_k and is negative (resp., positive) if x_k is negative (resp., positive). As a result, the sequence $(x_0, x_1, \dots)$, and also $(r_0, r_1, \dots)$, are monotonically increasing (resp., decreasing) if x_0 is positive (resp., negative). Consider again the LTL formula $\phi = GF(r < 0)$. Then in this case, for every infinite sequence $(r_0, r_1, \dots, r_m, \dots)$ with $r_i \geq 0, \forall i \geq m$ (i.e., a sequence violating ϕ), there exists a finite prefix $(r_0, \dots, r_k)$ with $r_k \geq 0$ (so that $x_k = (r_k+1)/2 \geq 0.5$) whose all infinite extensions also violate ϕ. Then G is prediagnosable with respect to $GF(r < 0)$. In this case the Büchi automaton accepting $S_\phi \cap A_G$ is given in Fig 4.1(b), where $\mathcal{L}_\phi = \{l_1\}$, i.e., $S_\phi \cap A_G$ is the limits of $(r < 0)^*$. The requirement model R is shown in Fig. 4.1(c), where the system behaviors satisfying ϕ visit l_1 infinitely often while those violating ϕ are absorbed at F. □

4.2 Approach to Detection Problem

Consider the detection structure of Fig. 4.2, where the monitored physical system G evolves according to stochastic difference equations (4.1)-(4.3), and the requirement model R tracks its own discrete location as the requirements variable r_k evolves. At any given time, the true state of the requirement model R is not available to the detector and must be estimated from the observed history of inputs and outputs. We transform this problem of estimating requirement-violation to fault-location reachability estimation in an input-output stochastic hybrid automaton (I/O-SHA) model that captures the behaviors of both G and R in a unified manner.

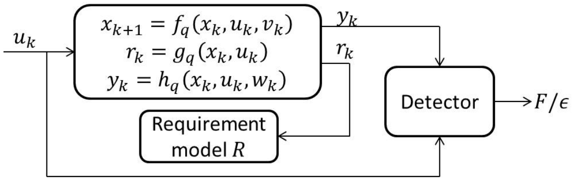


Figure 4.2 The detection structure.

We first introduce the notion of an I/O-SHA, extending that of a logical input-output hybrid automaton (I/O-HA) given in [72].

4.2.1 Input-Output Stochastic Hybrid Automaton

Definition 6. An input-output stochastic hybrid automaton (I/O-SHA) is a 10-tuple $P = (L, D, U, Y, \Sigma, \Delta, \ell_0, d_0, L_m, E)$, where

- L is the set of locations (symbolic states), and each $l \in L$ is a 3-tuple $l = (G_l, f_l, h_l)$, where
  - $G_l : D \times U \to [0, 1]$ is the location invariant probability satisfying (4.4) below,
  - $f_l : D \times U \times D \to [0, 1]$ assigns for each $(d, u) \in D \times U$ a probability density function $f_l(\cdot|d, u)$ on the data space D, and
  - $h_l : D \times U \times Y \to [0, 1]$ assigns for each $(d, u) \in D \times U$ a probability density function $h_l(\cdot|d, u)$ on the output space Y.
- $D = D_1 \times \cdots \times D_n \subseteq \mathbb{R}^n$ is the set of data (numerical states), and hence the hybrid state space of P is given by $L \times D$,
- $U = U_1 \times \cdots \times U_m \subseteq \mathbb{R}^m$ is the set of numerical inputs,
- $Y = Y_1 \times \cdots \times Y_p \subseteq \mathbb{R}^p$ is the set of numerical outputs,

- Σ is the set of symbolic inputs,
- Δ is the set of symbolic outputs,
- $\ell_0 : L \to [0, 1]$ is the initial probability distribution for the locations,
- $d_0 : D \to [0, 1]$ is the initial probability distribution for the data values,
- $L_m \subseteq L$ is the set of final locations,
- E is the set of edges (transitions), and each $e \in E$ is a 7-tuple $e = (o_e, t_e, \sigma_e, \delta_e, G_e, f_e, h_e)$, where
  - $o_e \in L$ is the original location,
  - $t_e \in L$ is the terminal location,
  - $\sigma_e \in \Sigma \cup \{\epsilon\}$ is the symbolic input,
  - $\delta_e \in \Delta \cup \{\epsilon\}$ is the symbolic output,
  - $G_e : D \times U \to [0, 1]$ is the guard probability satisfying (4.4) below,
  - $f_e : D \times U \times D \to [0, 1]$ assigns for each $(d, u) \in D \times U$ a probability density function $f_e(\cdot|d, u)$ on the data space D,
  - $h_e : D \times U \times Y \to [0, 1]$ assigns for each $(d, u) \in D \times U$ a probability density function $h_e(\cdot|d, u)$ on the output space Y.

Remark 9. In Definition 6, G_l and G_e, where $l \in L, e \in E$, capture the probabilities that an I/O-SHA stays in the current location l or executes a transition e, and so it satisfies the following stochasticity constraint:

$$\forall (d, u) \in D \times U, \sigma \in \Sigma \cup \{\epsilon\}, \quad G_l(d, u) + \sum_{e \in E : \sigma_e = \sigma} G_e(d, u) \leq 1. \tag{4.4}$$

Note that in certain special setting, the range space of G_l and G_e can simply be the binary set $\{0, 1\}$ [72], i.e., given any (d, u), an I/O-SHA will either stay at current location, or execute one transition, with probability 1. Then the guard/invariant can be equivalently written as logical predicates, $\overline{G}_l := \{(d, u) : G_l(d, u) = 1\} \subseteq D \times U$ and $\overline{G}_e := \{(d, u) : G_e(d, u) = 1\} \subseteq D \times U$.

Since in this dissertation, we consider refinement of discrete-time stochastic systems against their logical LTL formula, only logical guards/invariants are needed in the refined I/O-SHA models. □

An I/O-SHA P starts from an initial distribution ℓ_0 over L and an initial distribution d_0 over D. At each time step, given a current location l, current data value d and input value u, upon the arrival of a symbolic input $\sigma \in \Sigma \cup \{\epsilon\}$, P evolves either within the current location with probability $G_l(d, u)$ or executes an outgoing edge e such that $\sigma_e = \sigma$ with probability $G_e(d, u)$. In the former case, it updates the data variable d according to the distribution $f_l(\cdot|d, u)$, and the output variable y is assigned a value according to the distribution $h_l(\cdot|d, u)$. In the latter case, the distributions $f_e(\cdot|d, u)$ and $h_e(\cdot|d, u)$ are used for updating d and y, and a symbolic output δ_e is emitted.

Remark 10. In [63, 66], the authors proposed discrete time stochastic hybrid systems (DTSHS), which includes hybrid state/control space. The I/O-SHA model introduced here is more general than the DTSHS model: state variables of a DTSHS are fully observed, whereas the data variables of an I/O-SHA are only partially and unreliably observed. The notion of I/O-SHA can also be utilized to model cyber-physical systems [89, 90] where a cyber (discrete) component interacts with a physical (continuous) component. □

Next we present the refinement of a system against its LTL formula. Given a physical system G with dynamics described by (4.1)-(4.3) and the requirement model R, the refinement is modeled by an I/O-SHA G^R, where

- L is given by the state space of R, $l_0 = \delta(l_0^\phi)$ where δ is the Dirac delta function, d_0 is the initial distribution of x_0, and $L_m = \{F\}$,
- D, U, Y are given by the state/input/output space of G, respectively, and $\Sigma = \Delta = \emptyset$,
- the discrete transition structure of G^R is preserved from that of R,
- for each location $l \in L$,
  - location invariant $\overline{G}_l$ is given by $\overline{G}_l = \{(d, u) : g(d, u)$ violates the predicates over each outgoing transition from l in $R\}$,

  - probability density functions $f_l(\cdot|d,u)$ and $h_l(\cdot|d,u)$ for data updates and output assignments are determined by the distributions of v_k and w_k, together with the functions f and h of G,

- for each $e = (l, l', \sigma_e, \delta_e, \overline{G}_e, f_e, h_e)$, e is a transition of G^R (i.e., $e \in E$), if and only if,
  - there exists a transition from l to l' in R, and
  - $\overline{G}_e = \{(d,u) : g(d,u)$ satisfies the predicates over the above transition of $R\}$, and
  - $\sigma_e = \delta_e = \epsilon$, $f_e(d_r|d,u) = \delta(d_r - d)$, and $h_e(\cdot|d,u)$ is the identity function that keeps the output values unchanged on discrete transitions.

Remark 11. The refinement G^R captures the behaviors of both G and R in an unified manner such that, any system run associated with an indicator, transitions G^R to the fault-location $L_m = \{F\}$. □

4.2.2 State Estimation for I/O-SHA

In order to aid the estimation of fault location reachability, we present the stochastic filtering equations to recursively estimate the state distributions of I/O-SHA. Denote the history of observed inputs/outputs up to a time k as $u^k = (u_0, \dots, u_k)$, $y^k = (y_0, \dots, y_k)$ and let $z^k = (y^k, u^k)$. Define $\pi_{k+1}(\cdot|z^k) : L \to [0,1]$ as:

$$\forall l \in L, \qquad \pi_{k+1}(l|z^k) := Pr(l_{k+1} = l|z^k),$$

which is the conditional probability distribution over the discrete locations given the observations until time k. We further define two probability distribution functions over continuous variables of an I/O-SHA. The first one is the *prior* distribution $p_{k|k-1}(\cdot|z^{k-1}, l_k) : D \to [0,1]$ given by:

$$\forall l_k \in L, d \in D, p_{k|k-1}(d|z^{k-1}, l_k) := p_{d_k|z^{k-1},l_k}(d|z^{k-1}, l_k),$$

which is the probability density function over continuous variables at time k, given z^{k-1} (i.e., prior to the input/output at time k) and l_k (the discrete location at time k). The second one is the *posterior* distribution $p_{k|k}(\cdot|z^k, l_k) : D \rightarrow [0, 1]$ given by:

$$\forall l_k \in L, d \in D, p_{k|k}(d|z^k, l_k) := p_{d|z^k,l_k}(d_k|z^k, l_k),$$

which is the probability density function over continuous variables at time k, given z^k (i.e., post to the input/output at time k) and l_k (the discrete location at time k).

The following equations (4.5)-(4.9) initialize and recursively update the state distributions π_k, $p_{k|k}$ and $p_{k+1|k}$ for an I/O-SHA upon the arrival of the kth input/output pair. The detailed derivations of (4.7)-(4.9) are given in Appendix B. For each $l \in L, d \in D$:

$$\begin{aligned}
\pi_0(l|z^{-1}) &= l_0(l) && (4.5)\\
p_{0|0}(d_0|z^0, l) &= d_0(d_0') && (4.6)\\
p_{k|k}(d|z^k, l_k) &= \frac{h_{l_k}(y_k|d, u_k)p_{k|k-1}(d|z^{k-1}, l_k)}{\int_D h_{l_k}(y_k|d_k, u_k)p_{k|k-1}(d_k|z^{k-1}, l_k)d(d_k)} && (4.7)\\
\pi_{k+1}(l|z^k) &= \sum_{l_k \in L} \pi_k(l_k|z^{k-1}) \times \int_{D(l_k \rightarrow l|u_k)} p_{k|k}(d_k|z^k, l_k)d(d_k) && (4.8)\\
p_{k+1|k}(d|z^k, l_{k+1}) &= \frac{1}{\pi_{k+1}(l_{k+1}|z^k)} \sum_{l_k} \pi_k(l_k|z^{k-1})\\
&\quad \times \int_{D(l_k \rightarrow l_{k+1}|u_k)} f_{l_{k+1}}(d|d_k, u_k)p_{k|k}(d_k|z^k, l_k)d(d_k), && (4.9)
\end{aligned}$$

where $D(l_i \rightarrow l_j|u_i) \subseteq D$ for each l_i, l_j and u_i is defined as $D(l_i \rightarrow l_j|u_i) := \{d_i \in D : \exists e \in E, o_e = l_i, t_e = l_j, (u_i, d_i) \in \overline{G}_e\}$, i.e., it is the set of data values that enable the edge from l_i to l_j while the input is u_i.

4.2.3 Detection Statistics and Detection Scheme

Now that we have computed the state probability distributions given the input/output sequence up to a current time k, we can use this to compute the *likelihood of no-fault*, which is the probability of the refinement G^R being outside of the fault-location $L_m = \{F\}$, and is given by:

$$P_N^k \quad := \quad \sum_{l \notin L_m} \pi_{k+1}(l|z^k). \qquad (4.10)$$

Note P_N^k can be found by first computing π_k, which in turn is computed by the filter (4.5)-(4.9). A detector issues a fault decision "F" whenever this likelihood of no-fault is lower than a threshold, i.e., when $P_N^k \leq \rho$, and remains silent otherwise. The detector $\mathcal{D} : (U \times Y)^{\mathbb{N}} \rightarrow \{F, \epsilon\}$ is formally defined as:

$$\forall z^k \in (U \times Y)^{\mathbb{N}}, [\mathcal{D}(z^k) = F] \Leftrightarrow [\exists j \leq k, P_N^j \leq \rho]. \tag{4.11}$$

Note that once the detector issues F, it issues F for all subsequent steps, i.e., the detector "doesn't change its mind".

Remark 12. Note that while we only consider discrete-time stochastic systems with single mode of dynamics, the framework can be straightforwardly extended to the case where the system under diagnosis is itself an I/O-SHA. In this case, the locations of the refinement G^R are given by the location-pairs of G and R, and the guards/invariants are given by intersections of guards/invariants in G and R. The detection algorithm (4.5)-(4.11) continues to apply to this more general setting where G itself is an I/O-SHA. □

Remark 13. In this chapter, we consider a fault to be a violation of certain correctness requirement expressed as linear-time temporal logic (LTL) formulas. As studied in literature [11, 42, 91, 92, 93, 94], a fault may be modeled as a change in system dynamics. We can subsume this situation in our framework by considering the refinement G^R in which the probability density functions $f_l(\cdot|d, u)$ for location $l = F$ undergoes a dynamics change due to the occurrence of fault. Then the fault detection problem is again reduced to fault-location reachability detection problem for G^R, which can be solved by our proposed algorithm (4.5)-(4.11). □

4.3 Illustrative Example: A Room-Heating Problem

In this section we present the results for fault detection computations presented above by applying to a room heating benchmark, which aims to regulate the temperature in a single room with a single heater, and is inspired from [62, 95]. Let the continuous variable x_k present the room temperature at time k, and the binary variable u_k denote the status of the heater, with $u_k = 1$ if the heater is on at time k and 0 otherwise. The room temperature x_k is assumed to evolve according to the linear stochastic difference equation:

$$x_{k+1} = x_k + a(x_a - x_k) + bu_k + v_k,$$

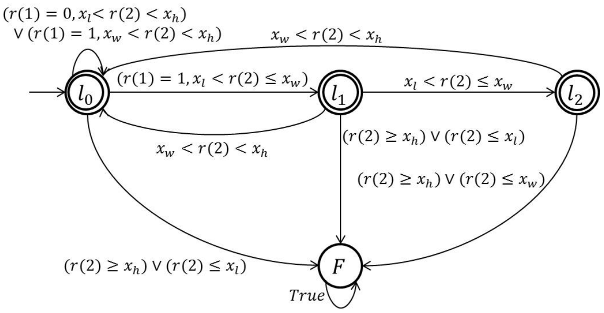

$(r(1) = 0, x_l < r(2) < x_h)$
$\vee\ (r(1) = 1, x_w < r(2) < x_h)$
$x_w < r(2) < x_h$
$(r(1) = 1, x_l < r(2) \leq x_w)$
l_0
l_1
l_2
$x_l < r(2) \leq x_w$
$(r(2) \geq x_h) \vee (r(2) \leq x_l)$
$x_w < r(2) < x_h$
$(r(2) \geq x_h) \vee (r(2) \leq x_w)$
$(r(2) \geq x_h) \vee (r(2) \leq x_l)$
F
$True$

Figure 4.3 The requirement model R for single room heating problem.

and the requirement and output variables are given by:

$$\begin{aligned} r_k &= \begin{bmatrix} u_k \\ x_k \end{bmatrix}, \\ y_k &= x_k + w_k, \end{aligned}$$

where x_a is the (constant) ambient temperature, and the disturbance v_k and the noise w_k are zero mean Gaussian random variables with variances σ_v^2 and σ_w^2, respectively.

For safety purposes, it is required that the room temperature satisfies $x_l \leq x_k \leq x_h$ for all k. It is also required that the room temperature is guaranteed to be higher than x_w in at most 2 steps after the heater is turned on. Note $x_h > x_w > x_l$ are constants, specified by user/designer. Such correctness requirement can be expressed as LTL formula ϕ:

$$\phi = G[\{x_l < r(2) < x_h\} \wedge \{(r(1) = 1) \Rightarrow (r(2) > x_w) \vee X(r(2) > x_w) \vee XX(r(2) > x_w)\}]. \tag{4.12}$$

It can be verified that the aforementioned system is prediagnosable with respect to ϕ, and the requirement model R is shown in Fig. 4.3, which has four states and 9 edges, while reaching the state F indicates the violation of formula (4.12).

The refinement G^R is such that $L = \{l_0, l_1, l_2, F\}$, $U = \{0, 1\}$, $D = X = Y = \mathbb{R}$, $l_0 = \delta(l_0)$, $d_0 = \delta(x_0)$, $L_m = \{F\}$ and the edges are as shown in Fig. 4.3. For each $l \in L$,

$$\begin{aligned} f_l(\cdot|d, u) &= \mathcal{N}(\cdot|d + a(x_a - d) + bu, \sigma_v^2), \text{ and} \\ h_l(\cdot|d, u) &= \mathcal{N}(\cdot|d, \sigma_w^2), \end{aligned}$$

where $\mathcal{N}(\cdot|\mu, \sigma^2)$ denotes Gaussian distribution with mean μ and variance σ^2. For each $l_j, l_j \in L$ and $u \in U$, $D(l_i \to l_j|u)$ can be easily computed and is shown in Table 4.1.

$D(l_0 \to l_0 \vert u = 0)$	(x_l, x_h)
$D(l_0 \to l_0 \vert u = 1)$	(x_w, x_h)
$D(l_0 \to l_1 \vert u = 1)$	$(x_l, x_w]$
$D(l_1 \to l_0 \vert u \in \{0, 1\})$	(x_w, x_h)
$D(l_1 \to l_2 \vert u \in \{0, 1\})$	$(x_l, x_w]$
$D(l_2 \to l_0 \vert u \in \{0, 1\})$	(x_w, x_h)
$D(l_0 \to F \vert u \in \{0, 1\})$	$(-\infty, x_l] \cup [x_h, \infty)$
$D(l_1 \to F \vert u \in \{0, 1\})$	$(-\infty, x_l] \cup [x_h, \infty)$
$D(l_2 \to F \vert u \in \{0, 1\})$	$(-\infty, x_w] \cup [x_h, \infty)$
$D(F \to F \vert u \in \{0, 1\})$	$(-\infty, \infty)$
Others	$\emptyset$

Table 4.1 List of $D(l_i \to l_j|u)$.

For the computational study, we set $x_a = 70$, $a = 0.1, b = 3, \sigma_v^2 = \sigma_w^2 = 0.4$, and suppose the system is initialized at $x_0 = 80$ and l_0. Note that with these selection of parameters, the system is stable. Suppose the specification parameters are $x_l = 70$, $x_h = 90$ and $x_w = 80$. For simulation, the continuous space is discretized by a grid size of 0.1 over the range $[65, 100]$. The input is such that the heater switches between on and off at each discrete time.

A total of 5000 runs, with terminal time $T = 200$, were simulated, out of which there were 457 runs violating the correctness requirement. We implemented the detection algorithm (4.5)-(4.11), and the results are shown in Figs. 4.4-4.6. In Fig. 4.4, the room temperature exceeds the upper limit, whereas in Fig. 4.5, the correctness requirement is violated since the room temperature remains below $x_w = 80$ two steps after the heater is on. In both cases, the likelihood of no-fault, P_N, drops soon after the specification model R reaches state F, and the fault can be detected with a delay of 7 steps by using a detection threshold $\rho < 0.5$. The performance of the detection scheme can be evaluated by the errors in terms of false alarms and missed detections (formally defined in next section), and Fig. 4.6 shows the number of runs that are false-alarmed or missed-detected over the 5000 runs, as the detection threshold ρ and detection delay n are changed. The number of runs that are false-alarmed is a function of the detection threshold and increases as the detection threshold increases, while the number of runs that are missed-detected is a function of both detection threshold and detection delay. When the detection delay is fixed, the number of runs that are missed-detected decreases as the detection threshold increases, whereas it decreases also as the detection delay increases while the detection threshold is fixed.

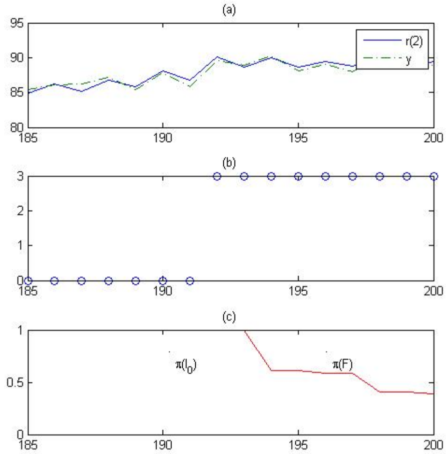


Figure 4.4 The detection result for a run that violates the correctness requirement by exceeding the upper limit x_h. (a) true $r(2) = x$ v.s. $y = x + w$; (b) the true state of specification model R where the fault-location F is represented by the number 3; (c) the estimate of state probability distribution.

4.4 Performance Evaluation and Stochastic Diagnosability

As illustrated in the case study in previous section, the performance of the detection scheme proposed above can be measured in terms of false alarm (FA) and missed detection (MD) rates. Here we formally define FA and MD rates, by first introducing the following notions.

A finite run of the system is a finite execution of the stochastic difference equations (4.1)-(4.3), denoted as $\overline{z} := (u^{|\overline{z}|}, x^{|\overline{z}|}, r^{|\overline{z}|}, y^{|\overline{z}|})$, where $|\overline{z}| < \infty$ and for each $o \in \{u, x, r, y\}$, $o^{|\overline{z}|} := (o_0, \ldots, o_{|\overline{z}|})$. A run is a fault run if the associated sequence of requirement variables $r^{|\overline{z}|}$ is an indicator, i.e., $r^{|\overline{z}|} \in I_\phi(G)$, where recall that $I_\phi(G)$ is the set of all indicators. A run is a nonfault run if it is not a fault run. Given two runs $\overline{z}_1 := (u_1^{|\overline{z}_1|}, x_1^{|\overline{z}_1|}, r_1^{|\overline{z}_1|}, y_1^{|\overline{z}_1|})$ and

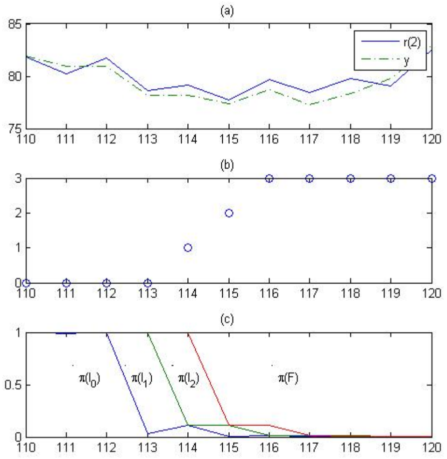


Figure 4.5 The detection result for a run that violates the correctness requirement by failing to reach x_w within 2 steps after the heater is on. (a) true $r(2) = x$ v.s. $y = x + w$; (b) the true state of specification model R where the fault-location F is represented by the number 3; (c) the estimate of state probability distribution.

$\overline{z}_2 := (u_2^{|\overline{z}_2|}, x_2^{|\overline{z}_2|}, r_2^{|\overline{z}_2|}, y_2^{|\overline{z}_2|})$, $\overline{z}_1$ is said to be a prefix of $\overline{z}_2$, denoted as $\overline{z}_1 \leq \overline{z}_2$, if $|\overline{z}_1| \leq |\overline{z}_2|$ and $o_2^{|\overline{z}_1|} \equiv o_1^{|\overline{z}_1|}$ for each $o \in \{u, x, r, y\}$. In this case we denote $\overline{z}_2 \backslash \overline{z}_1$ as an extension of $\overline{z}_1$.

Associated with each run $\overline{z}$ is a sequence of detection statistics, $P_N^0, P_N^1, \ldots, P_N^{|\overline{z}|}$, computed using (4.5)-(4.10). Then a FA occurs if the detector issues F decision for a nonfault run, and so the FA rate can be defined as:

$$P^{fa} := Pr(\overline{z} : r^{|\overline{z}|} \notin I_\phi(G) \wedge P_N^{|\overline{z}|} \leq \rho). \tag{4.13}$$

A MD occurs if the detector remains silent n steps after the system executes an indicator, where n is the detection delay bound allowed by the detector. Then the MD rate can be defined as:

$$P^{md} := Pr(\overline{z} : \exists k < |\overline{z}| - n, r^k \in I_\phi(G), P_N^{|\overline{z}|} > \rho). \tag{4.14}$$

In the following we present a characterization of the class of systems for which detectors with arbitrary accuracies can be designed, by introducing the notion of *Stochastic Diagnosability* which requires that for any tolerable threshold ρ and error bound τ, there must exist a delay bound n such that for any fault run, its extensions, longer than n and having likelihood of no-fault lower than ρ, occur with probability at most τ.

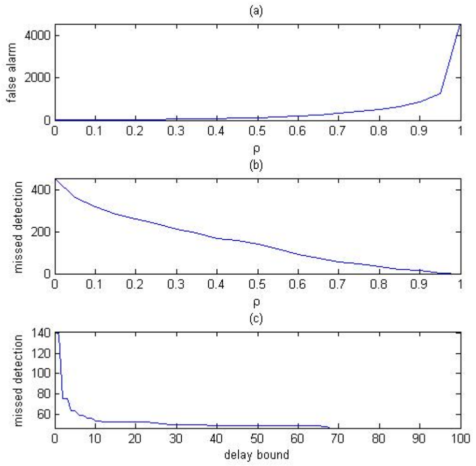


Figure 4.6 (a) The number of false alarms as a function of the threshold; (b) the number of missed detections as a function of the threshold; (c) the number of missed detections as a function of detection delay, when the threshold is $\rho = 0.75$.

Definition 7. Given a system G subjected to an input sequence drawn from a distribution μ, with correctness requirement expressed in LTL formula ϕ, (G, μ, ϕ) is said to be *Stochastically Diagnosable*, or simple *S-Diagnosable*, if $\forall \rho, \tau > 0$, $\exists n \in \mathbb{N}$, such that for any fault run $\overline{z}_0$,

$$Pr(\overline{z}\backslash\overline{z}_0 : |\overline{z}| - |\overline{z}_0| > n, P_N^{|\overline{z}|} > \rho) < \tau. \tag{4.15}$$

The following theorem establishes the significance of the S-Diagnosability property, by showing its necessity and sufficiency for the existence of a detector to achieve any desired level of accuracy as measured in terms of FA and MD rates.

Theorem 7. For any FA rate $\nu > 0$ and MD rate $\tau > 0$, there exists a detection threshold ρ and delay bound n so that the rates of FA and MD defined by (4.13)-(4.14) satisfy $P^{fa} < \nu$ and $P^{md} < \tau$ if and only if (G, μ, ϕ) is S-Diagnosable.

Proof. (Sufficiency) As shown in (4.13), for $\rho_1 > \rho_2 > 0$, $\{\overline{z} : r^{|\overline{z}|} \notin I_\phi(G) \wedge P_N^{|\overline{z}|} \leq \rho_1\} \supseteq \{\overline{z} : r^{|\overline{z}|} \notin I_\phi(G) \wedge P_N^{|\overline{z}|} \leq \rho_2\}$, and so the FA rate decreases as the detection threshold gets lower. Therefore, any FA rate ν can be achieved by adequately lowering the detection threshold. Let ρ_ν be the threshold that ensures FA rate ν. When (G, μ, ϕ) is S-Diagnosable, there exists an integer $n \in \mathbb{N}$ such that (4.15) holds. Therefore

$$\begin{aligned} P^{md} &= Pr(\overline{z} : \exists k < |\overline{z}| - n, r^k \in I_\phi(G), P_N^{|\overline{z}|} > \rho_\nu) \\ &= \sum_{\overline{z}_0 : r^{|\overline{z}_0|} \in I_\phi(G)} Pr(\overline{z}_0) \times Pr(\overline{z} \backslash \overline{z}_0 : |\overline{z}| - |\overline{z}_0| > n, P_N^{|\overline{z}|} > \rho_\nu) \\ &< \sum_{\overline{z}_0 : r^{|\overline{z}_0|} \in I_\phi(G)} Pr(\overline{z}_0)\tau < \tau. \end{aligned}$$

Thus the sufficiency holds.

(Necessity) When (G, μ, ϕ) is not S-Diagnosable, there exists $\rho_0, \tau_0 > 0$ and a fault run $\overline{z}_0$ such that for any $n \in \mathbb{N}$, (4.15) does not hold, i.e.,

$$Pr(\overline{z} \backslash \overline{z}_0 : |\overline{z}| - |\overline{z}_0| > n, P_N^{|\overline{z}|} > \rho_0) \geq \tau_0. \tag{4.16}$$

Let $\nu > 0$ be such that $\rho_\nu = \rho_0$. Then for any $n \in \mathbb{N}$,

$$\begin{aligned} P^{md} &= Pr(\overline{z} : \exists k < |\overline{z}| - n, r^k \in I_\phi(G), P_N^{|\overline{z}|} > \rho_0) \\ &\geq Pr(\overline{z}_0) Pr(\overline{z} \backslash \overline{z}_0 : |\overline{z}| - |\overline{z}_0| > n, P_N^{|\overline{z}|} > \rho_0) \\ &\geq Pr(\overline{z}_0)\tau_0 =: \tau_{low}. \end{aligned}$$

Therefore in this case, a MD rate of τ_{low} can not be achieved. Thus the necessity holds. □

Remark 14. Theorem 7 identifies the class of systems for which a detector of any desired accuracy can be constructed. Therefore, the S-Diagnosability property should be checked before designing a detector—A desired accuracy may not be achievable if S-Diagnosability is not satisfied. The future work will focus on the verification algorithm for S-Diagnosability, together with algorithm that computes a detector so as to ensure the desired rates of FA and MD. □

Example 9. Let us revisit the second system in Example 8. The state equation is given by:

$$x_{k+1} = x_k + v_k,$$

where the disturbance $v_k = sign(x_k)v'_k$ and v'_k is a positive-valued random variable with density function $f_{v'}$. The requirement and output variables are given by:

$$r_k = 2x_k - 1$$

$$y_k \quad = \quad 2x_k - 1 + w_k,$$

where w_k is i.i.d. zero mean Gaussian random variable with variance σ_w. Consider again the LTL formula $\phi = GF(r < 0)$. As shown in Example 8, the system is prediagnosable with respect to ϕ. Moreover, according to Fig. 4.1(c), detecting the requirement violation by time k is equivalent to detecting the existence of $l \leq k$ such that $r_l \geq 0$ (or $x_k \geq 0.5$).

Now consider a fault run $\overline{z}_0$ and its extension $\overline{z} \backslash \overline{z}_0$, we have

$$\begin{aligned} P_N^{|\overline{z}|} &= Pr(\forall 0 \leq l \leq |\overline{z}|, x_l < 0.5 \mid y_0, \dots, y_{|z|}) \\ &= \int_{-\infty}^{0.5} \cdots \int_{-\infty}^{0.5} \mathcal{N}(y_{|\overline{z}|} \mid 2x_{|\overline{z}|} - 1, \sigma_w) f(x_{|\overline{z}|} - x_{|\overline{z}|-1}) \\ &\qquad\qquad \times \cdots \times \mathcal{N}(y_1 \mid 2x_1 - 1, \sigma_w) f(x_1 - x_0) \\ &\qquad\qquad \times \mathcal{N}(y_0 \mid 2x_0 - 1, \sigma_w) d(x_0) dx_0 \cdots dx_{|\overline{z}|} \\ &\leq \int_{-\infty}^{0.5} \mathcal{N}(y_{|\overline{z}|} \mid 2x_{|\overline{z}|} 1, \sigma_w) dx_{|\overline{z}|}. \end{aligned}$$

For any $\rho > 0$, define y_ρ be such that

$$\int_{-\infty}^{0.5} \mathcal{N}(y_\rho \mid 2x_{|\overline{z}|} - 1, \sigma_w) dx_{|\overline{z}|} = \rho.$$

Then $(y_{|\overline{z}|} \geq y_\rho) \Rightarrow (P_N^{|\overline{z}|} \leq \rho)$, and so $(P_N^{|\overline{z}|} > \rho) \Rightarrow (y_{|\overline{z}|} < y_\rho)$. Hence,

$$Pr(\overline{z} \backslash \overline{z}_0 : P_N^{|\overline{z}|} > \rho) \quad \leq \quad Pr(\overline{z} \backslash \overline{z}_0 : y_{|\overline{z}|} < y_\rho)$$

According to the discussion of Example 8, for any fault run $\overline{z}_0$, the sequence of state variables $(x_0, x_1, \dots)$ is monotonically increasing. Therefore $\lim_{|\overline{z}| \to \infty} x_{|\overline{z}|} = \infty$ and so for a fixed ρ (or y_ρ), $\lim_{|\overline{z}| \to \infty} Pr(y_{|\overline{z}|} < y_\rho) = 0$ (See Fig. 4.7). Then we have $\lim_{n \to \infty} Pr(\overline{z} \backslash \overline{z}_0 : |\overline{z}| - |\overline{z}_0| > n, y_{|\overline{z}|} < y_\rho) = 0$, i.e., for any $\tau > 0$, there exists $n \in \mathbb{N}$, such that

$$Pr(\overline{z} \backslash \overline{z}_0 : |\overline{z}| - |\overline{z}_0| > n, P_N^{|\overline{z}|} > \rho) \leq Pr(\overline{z} \backslash \overline{z}_0 : |\overline{z}| - |\overline{z}_0| > n, y_{|\overline{z}|} < y_\rho) < \tau.$$

Since the above analysis works for any $\rho > 0$, one can conclude that S-Diagnosability holds in this example. According to Theorem 7, any desired rates of FA and MD can be achieved by suitably choosing threshold ρ and delay bound n. When the FA rate ν is made tighter by decreasing it, a smaller detection threshold ρ is required, while when the MD rate τ is made tighter by lowering it, a detector needs to wait for a longer delay bound n. □

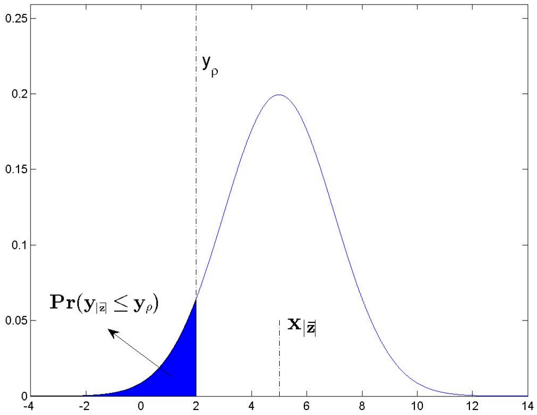


Figure 4.7   Gaussian distribution with mean $x_{|\overline{z}|}$ and variance σ_w.

4.5 Conclusion

In this chapter we studied the failure diagnosis of discrete-time stochastic systems subject to linear-time temporal logic correctness requirement. The continuous physical system (modeled as stochastic difference equations) was refined against its LTL correctness requirement to yield an input-output stochastic hybrid automaton which preserves the behavior of the physical system and captures the requirement-violation as a reachability property to a fault-location. The likelihood of no-fault was proposed as a detection statistic, and was recursively computed for issuing a detection decision (a fault decision is issued when the likelihood of no-fault drops below a suitably chosen threshold, implying the likelihood of no-fault has become “low” and so a fault is concluded). Although in the proposed framework, a fault is defined to be a violation of certain correctness requirement and does not necessarily result in a dynamics change, the framework can be straightforwardly adopted to capture fault models which involve a change in system dynamics as in [11, 93, 94, 91, 42, 92]. The proposed diagnosis procedure was implemented for a benchmark room heating problem to show the validity and applicability of the results. The performance of the procedure was evaluated in terms of false alarm and missed detection rates, and the existence of detector for achieving any desired false alarm and missed detection rates was captured as Stochastic Diagnosability introduced in this chapter. In future, the analytical computation of the rates of false alarm and missed detection will be investigated, together with the verification of the Stochastic-Diagnosability property.

CHAPTER 5. FAILURE PROGNOSIS OF STOCHASTIC DES

In this chapter we consider the fault prognosis problem, where the goal is to predict a fault prior to its occurrence. The problem of predicting a fault prior to its occurrence is a well researched area (see for example [73, 74, 75, 76, 77]). In [74] the notion of uniformly bounded prognosability of fault was formulated for logical discrete event systems (DESs), where each fault trace must possess a nonfault prefix such that for all indistinguishable traces, a future fault is inevitable within a bounded delay that is uniform across all fault traces. The notion was later extended to the decentralized setting in [75] and the requirement of the existence of a uniform bound was also removed. Reference [75] also established that the notion of prognosability is equivalent to the existence of a prognoser with no false alarm (FA) and no missed detection (MD). The issue of prognosability under a general decentralized inferencing mechanism was proposed in [79], where a prognostic decision involved inferencing among a group of local prognosers over their local decisions and their ambiguity levels, and the notion of inference-prognosability and its verification was introduced to capture the necessity and sufficiency of inferencing based decentralized prognosis. The problem of distributed prognosability under bounded-delay communications among the local prognosers was studied in [80], where the notion of joint-prognosability and its verification was proposed.

In order to generalize the notion of prognosability to stochastic DESs, in this chapter, we introduce m-steps Stochastic Prognosability, or simply S_m-Prognosability, which requires for any tolerance level ρ and error bound τ, there exists a reaction bound $k \geq m$, such that the set of fault traces for which a fault cannot be predicted k steps in advance with tolerance level ρ, occurs with probability smaller than τ. We formalize the notion of a prognoser that maps observations to decisions by comparing a suitable statistic with a threshold, and show that S_m-Prognosability is a necessary and sufficient condition for the existence of a prognoser with reaction bound at least m (i.e., prediction at least m-steps prior to the occurrence of a fault) that can achieve any specified FA and MD rate requirement. In this sense S_m-Prognosability can be viewed as a generalization of the logical prognosability, since it provides a basis for the existence and synthesis of a prognoser that can achieve a user-specified level of FA and MD. In contrast, the logical version is rather rigid, offering no further options for systems that fail to be logically prognosable, even when there may exist a prognoser that can achieve a satisfying performance as measured in terms of FA and MD rates. Further, we also provide a polynomial algorithm for verifying S_m-Prognosability. We show that even the weakest form of stochastic-prognosability where the reaction bound is zero, namely, S_0-Prognosability, is stronger than S-Diagnosability, meaning that whenever it is possible to predict faults (even with zero reaction bound), it is also possible to diagnose those, as can be expected.

5.1 Stochastic Prognosability of DESs

We first formalize the notion of prognosability, called *m-steps Stochastic Prognosability*, or simply *S_m-Prognosability*, for stochastic DESs, and provide necessary and sufficient conditions for the verification of S_m-Prognosability. In the next section we show that for finite state systems, S_m-Prognosability is necessary and sufficient for the existence of a prognoser that can predict a fault at least m-steps prior to occurrence, while achieving any arbitrary false alarm and missed detection rates.

Let L be a nonempty closed language and $K \subseteq L$ be a nonempty closed language representing a nonfault specification. The fault prognosis problem is to predict an execution in $L - K$ before its occurrence. In order to be able to make a prognostic decision, we define the *n-step prognostic probability of no-fault* following an observation $o \in M(L)$ as:

$$P_N^n(o) = \frac{Pr(\{M^{-1}(o)\}\Sigma^n \cap K)}{Pr(\{M^{-1}(o)\}\Sigma^n \cap L)} = \frac{Pr(\{M^{-1}(o) \cap K\}\Sigma^n \cap K)}{Pr(M^{-1}(o) \cap L)}, \tag{5.1}$$

and the *least prognostic probability of no-fault* following $o \in M(L)$ as:

$$P_N^*(o) = \min_{n \in \mathbb{N}} P_N^n(o) = \frac{\min_{n \in \mathbb{N}} Pr(\{M^{-1}(o)\}\Sigma^n \cap K)}{Pr(\{M^{-1}(o)\} \cap L)}. \tag{5.2}$$

Note $P_N^n(o)$ is the probability, following the observation o, that the system does not execute a fault in the next n steps; and $P_N^*(o)$ is the least probability, following the observation o, that the system does not execute a fault over all finite-step futures. Note in the denominator of (5.1), we used the fact that probability of all extensions of length n, beyond the traces in $M^{-1}(o)$, is the same as the probability of traces in $M^{-1}(o)$, for there is no termination at any of the states. As a result, the denominator is constant with respect to n, and the minimum only applies to the numerator in (5.2).

To help formalize the prognosability for stochastic DESs, we introduce the notions of *boundary* fault traces whose all strict prefixes are nonfault, *m-steps interior* nonfault traces for which a fault can occur in the next $(m+1)$th step while no fault can occur within the next m steps, *persistent* nonfault traces whose all extensions are nonfault, *indicator* nonfault traces for which a future fault is guaranteed with arbitrary confidence and *nonindicator* nonfault traces that are not the indicator traces.

Definition 8. Given a pair (L, K) of closed languages with $K \subseteq L$, we define the set of

- *boundary* fault traces as, $\partial := \{s \in L - K : pr(s) - \{s\} \subseteq K\}$;
- *m-steps interior* nonfault traces of K with respect to L (where $m \geq 0$) as, $\partial_m^- := \{s \in K : \{s\}\Sigma^m \cap \partial = \emptyset, \{s\}\Sigma^{m+1} \cap \partial \neq \emptyset\}$;
- *persistent* nonfault traces of K with respect to L as, $\aleph := \{s \in K : \forall n \in \mathbb{N}, \{s\}\Sigma^n \cap (L - K) = \emptyset\} = \{s \in K : \forall n \in \mathbb{N}, Pr(\{s\}\Sigma^n \cap K) = Pr(s)\}$;
- *indicator* nonfault traces of K with respect to L as, $\mathfrak{J} := \{s \in K : \forall \rho > 0, \exists n \in \mathbb{N}, Pr(\{s\}\Sigma^n \cap K) \leq \rho\}$;
- *nonindicator* nonfault traces of K with respect L as, $\Upsilon := K - \mathfrak{J}$.

Note that $\Upsilon = \{s \in K : \exists \rho > 0, \forall n \in \mathbb{N}, Pr(\{s\}\Sigma^n \cap K) > \rho\}$, and since $\aleph$ is obtained by replacing ρ by $Pr(s)$ in the right hand side of this equality, it follows that $\aleph \subseteq \Upsilon$. Also note that $\aleph$ is "extension-closed" in the sense that if it possesses $s \in K$, then it also possesses all extensions $t \in L$ with $s \leq t$.

Next we introduce the definition of S_m-Prognosability which requires that, for any threshold value $\rho > 0$ and error bound $\tau > 0$, there exists a reaction bound $k \geq m$, such that the set of boundary fault traces, that are either shorter than k in length or for which a prognostic decision can not be made k steps in advance with confidence level ρ, occurs with probability smaller than τ.

Definition 9. A pair (L, K) of closed languages with $K \subseteq L$ is said to be *m-steps Stochastically Prognosable*, or simply *S_m-Prognosable*, if

$$(\forall \tau, \rho > 0)(\exists k \geq m) Pr(s \in \partial : [|s| \leq k] \vee [\forall u \in s/\Sigma^{>k}, P_N^*(M(u)) > \rho]) < \tau, \tag{5.3}$$

where P_N^* is as defined by (5.1) and (5.2).

The next lemma states that we can always choose the reaction bound k in Definition 9 to equal m, thereby simplifying the definition a bit.

Lemma 1. A pair (L, K) of closed languages with $K \subseteq L$ is S_m-Prognosable if and only if $\forall \tau, \rho > 0$,

$$Pr(s \in \partial : [|s| \leq m] \vee [\forall u \in s/\Sigma^{>m}, P_N^*(M(u)) > \rho]) < \tau. \tag{5.4}$$

Proof. The sufficiency is obvious by choosing $k = m$. Now to see the converse, assume (5.4) is not true, i.e., $\exists \tau > 0, \rho > 0$, s.t. $Pr(s \in \partial : [\forall u \in s/\Sigma^{>m}, P_N^*(M(u)) > \rho] \vee [|s| \leq m]) \geq \tau$. Since we have for all $k \geq m$, $\{s \in \partial : [\forall u \in s/\Sigma^{>k}, P_N^*(M(u)) > \rho] \vee [|s| \leq k]\} \supseteq \{s \in \partial : [\forall u \in s/\Sigma^{>m}, P_N^*(M(u)) > \rho] \vee [|s| \leq m]\}$, and hence $Pr(s \in \partial : [\forall u \in s/\Sigma^{>k}, P_N^*(M(u)) > \rho] \vee [|s| \leq k]) \geq Pr(s \in \partial : [\forall u \in s/\Sigma^{>m}, P_N^*(M(u)) > \rho] \vee [|s| \leq m]) \geq \tau$. Therefore according to Definition 9, (L, K) is not S_m-Prognosable. Hence the necessity also holds. □

Denote $\ell(\partial) = \min\{|s|, s \in \partial\}$ as the length of the shortest fault trace in $L - K$. Then the following theorem provides a necessary and sufficient condition for S_m-prognosability requiring the reaction bound m to be smaller than the length of the shortest fault trace, $\ell(\partial)$, and every boundary fault trace in ∂ to possess a nonfault prefix which is more than m-steps shorter and is unambiguously an indicator.

Theorem 8. A pair (L, K) of closed languages with $K \subseteq L$ is S_m-Prognosable if and only if $m < \ell(\partial)$ and

$$(\forall s \in \partial)(\exists u \in s/\Sigma^{>m})(M^{-1}M(u) \cap K \subseteq \mathfrak{I}). \tag{5.5}$$

Proof. (Sufficiency) For any $s \in \partial$, let $u \in s/\Sigma^{>m}$ be such that $M^{-1}M(u) \cap K \subseteq \mathfrak{J}$. Then

$$P_N^n(M(u)) = \frac{Pr(\{M^{-1}M(u) \cap K\}\Sigma^n \cap K)}{Pr(M^{-1}M(u) \cap L)} = \frac{\sum_{u' \in M^{-1}M(u) \cap K} Pr(\{u'\}\Sigma^n \cap K)}{Pr(M^{-1}M(u) \cap L)}.$$

For any $\rho > 0$, define $\rho_{u'} := \rho Pr(u') > 0$ for each $u' \in M^{-1}M(u) \cap K$. Then since $M^{-1}M(u) \cap K \subseteq \mathfrak{J}$, for each $u' \in M^{-1}M(u) \cap K$, exists $n_{u'} \in \mathbb{N}$ such that $Pr(\{u'\}\Sigma^{n_{u'}} \cap K) \leq \rho_{u'}$. Let $d := \max_{u' \in M^{-1}M(u) \cap K} n_{u'}$. Note that d here is a finite integer even if $M^{-1}M(u)$ is an infinite set (resulted by unobservable loops). To see this, let $u_1 = u_{11}u_{12}$ and $u_2 = u_{11}\sigma_1 \ldots \sigma_k u_{12}$ such that $\sigma_1 \ldots \sigma_k$ is an unobservable loop. Then we have $Pr(\{u_2\}\Sigma^{n_{u_1}} \cap K) = Pr(\sigma_1 \ldots \sigma_k)Pr(\{u_1\}\Sigma^{n_{u_1}} \cap K) < \rho Pr(\sigma_1 \ldots \sigma_k)Pr(u_1) = \rho Pr(u_2) = \rho_{u_2}$, and thus $n_{u_2} \leq n_{u_1}$. Therefore to find d, we only need to consider $u' \in M^{-1}M(u) \cap K$ such that u' doesn't contain any unobservable loop, making d finite. Then

$$\begin{aligned}
P_N^d(M(u)) &= \frac{\sum_{u' \in M^{-1}M(u) \cap K} Pr(\{u'\}\Sigma^d \cap K)}{Pr(M^{-1}M(u) \cap L)} \\
&\leq \frac{\sum_{u' \in M^{-1}M(u) \cap K} \rho_{u'}}{Pr(M^{-1}M(u) \cap L)} \\
&= \frac{\sum_{u' \in M^{-1}M(u) \cap K} \rho Pr(u')}{Pr(M^{-1}M(u) \cap L)} \\
&= \frac{Pr(M^{-1}M(u) \cap K)}{Pr(M^{-1}M(u) \cap L)}\rho \leq \rho. \text{ Hence,} \\
P_N^*(M(u)) &\leq P_N^d(M(u)) \leq \rho.
\end{aligned}$$

Also since $m < \ell(\partial)$ implies $\{s \in \partial : |s| \leq m\} = \emptyset$, we have for all $\rho > 0$ and $\tau > 0$, $Pr(s \in \partial : [\forall u \in s/\Sigma^{>m}, P_N^*(M(u)) > \rho] \vee [|s| \leq m]) = 0 < \tau$. According to Lemma 1, (L, K) is S_m-Prognosable.

(Necessity) When $m \geq \ell(\partial)$, let $s \in \partial$ be such that $|s| = \ell(\partial) \leq m$. Obviously for any $\tau \leq Pr(s)$, $Pr(s \in \partial : [\forall u \in s/\Sigma^{>m}, P_N^*(M(u)) > \rho] \vee [|s| \leq m]) \geq Pr(s \in \partial : |s| \leq m) \geq Pr(s) \geq \tau$ for all $\rho > 0$. Therefore (L, K) is not S_m-Prognosable. When $m < \ell(\partial)$, but (5.5) is not true, let $s \in \partial$ be such that $(\forall u \in s/\Sigma^{>m})(M^{-1}M(u) \cap K \cap \Upsilon \neq \emptyset)$. Then for any $u \in s/\Sigma^{>m}$ and $u' \in M^{-1}M(u) \cap K \cap \Upsilon$,

$$P_N^n(M(u)) = \frac{Pr(\{M^{-1}M(u) \cap K\}\Sigma^n \cap K)}{Pr(M^{-1}M(u) \cap L)} \geq \frac{Pr(\{u'\}\Sigma^n \cap K)}{Pr(M^{-1}M(u) \cap L)}.$$

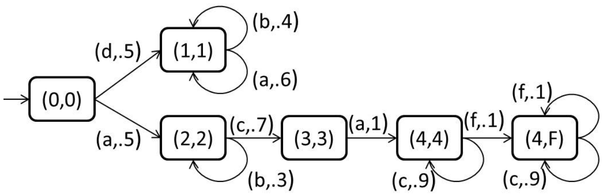


Figure 5.1 Refinement G^R for Example 10.

Since $u' \in \Upsilon$, there exists $\rho_{u'} > 0$ such that $\forall n \in \mathbb{N}, Pr(\{u'\}\Sigma^n \cap K) > \rho_{u'}$. Therefore for any $n \in \mathbb{N}$,

$$P_N^n(M(u)) \geq \frac{Pr(\{u'\}\Sigma^n \cap K)}{Pr(M^{-1}M(u) \cap L)} > \frac{\rho_{u'}}{Pr(M^{-1}M(u) \cap L)} =: \rho_u,$$

and hence

$$P_N^*(M(u)) \quad = \quad \min_{n\in\mathbb{N}} P_N^n(M(u)) > \rho_u.$$

Thus for any $u \in s/\Sigma^{>m}$, there exists $\rho_u > 0$ such that $P_N^*(M(u)) > \rho_u$. Therefor for any $0 < \rho < \min_{u\in s/\Sigma^{>m}} \rho_u$ and $0 < \tau < Pr(s)$, $Pr(s \in \partial : [\forall u \in s/\Sigma^{>m}, P_N^*(M(u)) > \rho] \vee [|s| \leq m]) \geq Pr(s) > \tau$. Hence (L, K) is not S_m-Prognosable, according to Lemma 1. □

Example 10. For refined system shown in Fig. 5.1, the observation mask M is such that $M(\{d, f\}) = \{\epsilon\}$ and $M(\sigma) = \sigma$ for $\sigma \in \Sigma - \{d, f\}$. In G^R there are two closed SCCs, one is formed by the nonfault state $(1, 1)$ and its selfloop transitions whereas the other is formed by the fault state $(4, F)$ and its selfloop transitions. Since $\ell(\partial) = 4$, by Theorem 8, the system can not be S_m-Prognosable with $m \geq 4$. The set of indicator traces is $\mathfrak{J} = \{a\}\Sigma^* \cap K$, and the set of nonindicator traces is $\Upsilon = \{\epsilon\} \cup \{d\}\Sigma^* \cap L$, while the set of boundary fault traces is $\partial = ab^*cac^*f$. One can check that for any $s \in \partial$, there exists $u \in s/\Sigma^{>1} \subseteq \{ab^*c\}\Sigma^* \cap K$ such that $M^{-1}M(u) \cap K \subseteq \mathfrak{J}$. Therefore by Theorem 8, (L, K) is S_1-Prognosable. On the other hand, for $s = acaf \in \partial$, $u = a \in s/\Sigma^{>2}$ is such that $M^{-1}M(u) \cap K \cap \Upsilon = \{da\} \neq \emptyset$. Therefore by Theorem 8, (L, K) is not S_2-Prognosable. □

The following corollary is directly obtained from Theorem 8, and captures the expected property that prognosability continues to hold even with smaller reaction bound.

Corollary 1. Given a pair (L, K) of closed languages with $K \subseteq L$, if (L, K) is S_m-Prognosable, then (L, K) is $S_{m'}$-Prognosable for all nonnegative $m' \leq m$, whereas if (L, K) is not S_m-Prognosable, then (L, K) is not $S_{m'}$-Prognosable for all $m' \geq m$.

For a S_m-Prognosable system, Theorem 8 requires that each boundary fault trace possess a more than m-steps shorter prefix that is unambiguously an indicator. We can strengthen this theorem by requiring that *exactly* the $(m+1)$-shorter prefix possess the said property. This requires the result of the next lemma stating that indicators are "extension-closed" (nonfault extensions of indicators are also indicators), while nonindicators are prefix-closed (prefixes of nonindicators are also nonindicators).

Lemma 2. For a pair (L, K) of closed languages with $K \subseteq L$, it holds that $\mathfrak{J}\Sigma^* \cap K \subseteq \mathfrak{J}$, and $pr(\Upsilon) \subseteq \Upsilon$.

Proof. Let $s \in \mathfrak{J}$ be arbitrary, i.e., $\forall \rho > 0$, $\exists n \in \mathbb{N}$ s.t. $Pr(\{s\}\Sigma^n \cap K) \leq \rho$. Since for any $t \in K \backslash s$, $Pr(\{st\}\Sigma^l \cap K) \leq Pr(\{s\}\Sigma^{l+|t|} \cap K)$, we have $\forall \rho > 0$, $\exists l = n - |t| \in \mathbb{N}$ s.t. $Pr(\{st\}\Sigma^l \cap K) \leq Pr(\{s\}\Sigma^{l+|t|} \cap K) = Pr(\{s\}\Sigma^n \cap K) \leq \rho$. According to Definition 8, $st \in \mathfrak{J}$, i.e., $\forall s \in \mathfrak{J}, t \in K \backslash s$, $st \in \mathfrak{J}$. Therefore $\mathfrak{J}\Sigma^* \cap K \subseteq \mathfrak{J}$.

Similarly let $s \in \Upsilon$ be arbitrary, i.e., $\exists \rho > 0$ s.t. $\forall n \in \mathbb{N}$, $Pr(\{s\}\Sigma^n \cap K) > \rho$. Then for any $u \in pr(s)$, $Pr(\{u\}\Sigma^l \cap K) \geq Pr(\{s\}\Sigma^{l-|s|+|u|} \cap K) > \rho$ for any $l - |s| + |u| \in \mathbb{N}$ and hence for any $l \in \mathbb{N}$. According to Definition 8, $u \in \Upsilon$, i.e., $\forall s \in \Upsilon, u \in pr(s)$, $u \in \Upsilon$. Therefore $pr(\Upsilon) \subseteq \Upsilon$. □

Using Lemma 2, we can strengthen Theorem 8 to obtain a new result which we employ later for verifying S_m-Prognosability. The new theorem states that S_m-Prognosability holds if and only if the reaction bound $m < \ell(\partial)$, and all m-steps interior traces are distinguishable from any nonindicator trace.

Theorem 9. A pair (L, K) of closed languages with $K \subseteq L$ is S_m-Prognosable if and only if $m < \ell(\partial)$ and

$$M^{-1}M(\partial_m^-) \cap \Upsilon = \emptyset. \tag{5.6}$$

Proof. If $m < \ell(\partial)$ and (5.6) is true, then it follows from the fact that every fault trace $s \in \partial$ possesses a nonfault prefix $u \in \partial_m^-$ satisfying $u \in s/\Sigma^{>m}$ and Theorem 8 that (L, K) is S_m-Prognosable, and the sufficiency follows. On the other hand, if $m \geq \ell(\partial)$, then by Theorem 8, (L, K) is not S_m-Prognosable. Meanwhile if $m < \ell(\partial)$ but (5.6) is not true, then we can select $s \in \partial_m^-$ and $s' \in \Upsilon$ such that $M(s) = M(s')$. Then for any $u \in pr(s)$, there exists $u' \in pr(s')$ such that $M(u) = M(u')$ and $u' \in \Upsilon$ (Lemma 2), i.e., $\forall u \in pr(s)$, $M^{-1}M(u) \cap K \cap \Upsilon \neq \emptyset$. It follows from the definition of ∂_m^- that there exists $st \in \partial$ such that $st/\Sigma^{>m} = pr(s)$, and hence $\forall u \in st/\Sigma^{>m} = pr(s)$, $M^{-1}M(u) \cap K \cap \Upsilon \neq \emptyset$. According to Theorem 8, (L, K) is not S_m-Prognosable. Thus the necessity also holds. □

Example 11. For refined system shown in Fig. 5.1, $\mathfrak{J} = \{a\}\Sigma^* \cap K$, $\Upsilon = \{\epsilon\} \cup \{d\}\Sigma^* \cap L$, $\partial_2^- = ab^*$ and $\partial_1^- = ab^*c$. One can easily check that $M^{-1}M(\partial_2^-) \cap \Upsilon = dab^* \neq \emptyset$ and $M^{-1}M(\partial_1^-) = ab^*c \subseteq \mathfrak{J}$. Therefore (L, K) is S_1-Prognosable but not S_2-Prognosable, as discussed in Example 10. □

5.2 Prognoser and its Existence Condition

In order to predict a fault in advance, the prognoser computes for each $o \in M(L)$, the prognostic probability of no-fault $P_N^*(o)$ as defined by (5.1)-(5.2), and compares it with an appropriately chosen threshold ρ. Whenever $P_N^*(o)$ is below this threshold, implying that there is only a small likelihood of no-fault in future, the prognoser issues a fault warning F, predicting/prognosing a future fault, and otherwise it remains silent (issues ϵ). In other words, a prognoser is formally a map, $D : M(L) \to \{F, \epsilon\}$ defined as:

$$\forall o \in M(L), [D(o) = F] \Leftrightarrow [\exists \bar{o} \leq o : P_N^*(\bar{o}) \leq \rho], \tag{5.7}$$

where P_N^* is as defined by (5.1) and (5.2). Note that according to (5.7), once a warning is issued, it remains unchanged for the subsequent extensions.

For a prognoser that aims to predict a fault at least m steps before its occurrence, a *miss detection* (MD) occurs when a fault happens while the prognoser fails to issue a warning m steps in advance. On the other hand a *false alarm* (FA) occurs when a warning is issued for a trace whose all extensions are nonfault, i.e., a trace in $\aleph$. Therefore the MD rate P^{md} and the FA rate P^{fa} for a m-prognoser can be defined as:

$$\begin{aligned} P^{md} &= Pr(s \in \partial : [|s| \leq m] \vee [D(M(s/\Sigma^{m+1})) = \epsilon] \quad (5.8)\\ P^{fa} &= Pr(s \in \aleph : D(M(s)) = F). \quad (5.9)\end{aligned}$$

Considering the fact the once the prognoser issues F, it issues F for any subsequent observations, the above equations can also be equivalently presented as:

$$\begin{aligned} P^{md} &= Pr(s \in \partial : [|s| \leq m] \vee [\forall u \in s/\Sigma^{>m}, P_N^*(M(u)) > \rho])\\ P^{fa} &= Pr(s \in \aleph : \exists u \in pr(s), P_N^*(M(u)) \leq \rho).\end{aligned}$$

Example 12. For the system G^R shown in Fig. 5.1. Suppose G^R executes $dabbb$ and produces observation $o = abbb$, then $P_N^*(o) = 0.5872$. Hence for any m-prognoser with threshold $\rho \geq 0.5872$, traces in $\{dabbb\}\Sigma^* \cap L$ will be false alarmed. When G^R executes a trace in $ab^*cac^*f \subseteq \partial$ and produces an observation $o \in ab^*cac^*$, then $P_N^*(o)$ approaches 0. Therefore for a 1-prognoser with any threshold ρ, all fault traces can be prognosed, and hence no missed detection. However, for a 2-prognoser with $\rho = 0.3$, when G^R executes the fault trace $abcaf$, a prognostic decision can be made only upon observing abc (since for all its prefixes, the threshold remains lower than the prognostic probability of no fault: $P_N^*(\epsilon) = 0.5$, $P_N^*(a) = 0.375$, $P_N^*(ab) = 0.444$, $P_N^*(abc) = 0$), which violates the least reaction bound $m = 2$, and hence $abcaf$ gets missed detected. □

In order to establish a condition for the existence of a m-prognoser in terms of the property of S_m-prognosability, we first establish the following corollary of Theorem 8 and Lemma 2.

Corollary 2. If a pair (L, K) of closed languages with $K \subseteq L$ is S_m-Prognosable, then $M^{-1}M(\Upsilon) \cap (L - K) = \emptyset$.

Proof. Suppose for contradiction that (L, K) is S_m-Prognosable and there exists $s \in \Upsilon$ such that $M^{-1}M(s) \cap (L - K) \neq \emptyset$. Let $s' \in M^{-1}M(s) \cap (L - K)$. Then for all $u' \in pr(s')$, there exists $u \in pr(s)$ such that $M(u) = M(u')$. According to Lemma 2, $u \in \Upsilon$. Therefore, $\forall u' \in pr(s') \cap K$, $M^{-1}M(u') \cap K \cap \Upsilon \neq \emptyset$. By Theorem 8, (L, K) is not S_m-Prognosable for any $m \in \mathbb{N}$, which contradicts the assumption that (L, K) is S_m-Prognosable. □

The next lemma which states that under the assumption of regularity of languages L and K, equivalently the finiteness of the state-space of G^R, no extension of an indicator can be persistently nonfault, whereas some extension of a nonindicator must be persistently nonfault. The lemma requires the finiteness of the state-space that guarantees the probability of staying in a transient state approaches 0 while the system evolves.

Lemma 3. For a pair (L, K) of closed regular languages with $K \subseteq L$, we have $\mathfrak{I}\Sigma^* \cap \aleph = \emptyset$ and $\Upsilon\Sigma^* \cap \aleph \neq \emptyset$.

Proof. Assume for contradiction that there exists $s \in \mathfrak{I}$ such that $\{s\}\Sigma^* \cap \aleph \neq \emptyset$. Let $u = \sigma_1 \dots \sigma_n \in K \backslash s$ be such that $su \in \aleph$. Then for any $l \in \{1, \dots, n\}$, $Pr(\{s\}\Sigma^l \cap K) \geq Pr(s\sigma_1 \dots \sigma_l) \geq Pr(su)$, and for $l > n$, $Pr(\{s\}\Sigma^l \cap K) \geq Pr(\{su\}\Sigma^{l-n} \cap K) = Pr(su)$, i.e., there exists $0 < \rho < Pr(su)$ such that for any $l \in \mathbb{N}$, $Pr(\{s\}\Sigma^l \cap K) > \rho$. Therefore $s \notin \mathfrak{I}$, a contradiction.

Similarly assume for contradiction that there exists $s \in \Upsilon$ such that $\{s\}\Sigma^* \cap \aleph = \emptyset$. Then for any $u \in L\backslash s$, it possesses a fault extension $t \in (L - K)\backslash su$, i.e., the “nonfaulty-ness of s” is a transient property. Since the language L and K are regular and have finite state representations, for any $\rho > 0$, there exists $n \in \mathbb{N}$ such that $Pr(t \in K\backslash s, |t| \geq n) \leq \rho$, i.e., $Pr(\{s\}\Sigma^n \cap K) = Pr(s)Pr(t \in K\backslash s, |t| = n) \leq \rho Pr(s) := \rho'$ holds for any $\rho' > 0$. Hence $s \in \mathfrak{I}$, which contradicts the assumption that $s \in \Upsilon$. □

Remark 15. Note by Lemma 3, no extension of an indicator trace can persistently be a nonfault trace. This requirement is weaker than the corresponding requirement for an indicator trace in the logical setting: All extensions of an indicator trace must be a fault trace within a bounded steps. A consequence of this is that, in the logical setting, an indicator trace cannot visit a cycle of nonfault states [75], which can be restrictive. In contrast, in stochastic setting, an indicator is allowed to visit a cycle of nonfault states as long as the cycle is non-absorbing (i.e., it has a positive exit probability, which ensures the non-persistence of remaining nonfault). □

Now we are ready to present the main result of the section, which shows that for regular languages L and K, S_m-Prognosability is necessary and sufficient for the existence of a m-prognoser to satisfy any level of FA and MD rates.

Theorem 10. Consider a pair (L, K) of closed regular languages with $K \subseteq L$. Then for any FA rate $\phi > 0$ and MD rate $\tau > 0$, there exists a m-prognoser (and its associated prognostic decision threshold) defined by (5.7) such that the MD and FA rates defined by (5.8)-(5.9) satisfy $P^{md} \leq \tau$ and $P^{fa} \leq \phi$ if and only if (L, K) is S_m-Prognosable.

Proof. (Sufficiency) Suppose (L, K) is S_m-Prognosable. Then for a nonfault trace $s \in K - \aleph$, its extensions continuing to remain in $K - \aleph$ is a transient property. Since the language L and K are regular and have finite state representations, we have for any $\phi_1 > 0$, $\exists d_1 \in \mathbb{N}$ such that $Pr(s \in (K - \aleph) \cap \Sigma^{>d_1}) < \phi_1$. For any $s \in \aleph \cap \Sigma^{d_1}$, if we pick $\rho'_s := \min_{u \in pr(s)} P^*_N(M(u)) > 0$, we can ensure that s is not false alarmed. For any $s \in \aleph \cap \Sigma^{d_1}$, according to Lemma 5 (presented in Appendix C), for any $\phi_2 > 0$ and $\rho'_2 > 0$, there exists $d_2 \in \mathbb{N}$, such that the set of extensions of s that are longer than d_2 and have P^*_N values of their observations smaller than ρ'_2, occur with probability smaller than ϕ_2, i.e., $P^{fa}(s) < \phi_2$.

Let $d = d_1 + d_2$. If we pick $\rho' = \min_{u \in pr(s), s \in \aleph \cap \Sigma^d} P^*_N(M(u)) > 0$, $\rho < \min(\rho'_2, \rho')$ and $\phi_1 + \phi_2 < \phi$, then P^{fa} is upper bounded by

$$\begin{aligned}
P^{fa} &= Pr(s \in \aleph : \exists u \in pr(s), P^*_N(M(u)) \leq \rho) \\
&= Pr(s \in \aleph : pr(s) \cap \Sigma^d \cap \aleph = \emptyset, \exists u \in pr(s), P^*_N(M(u)) \leq \rho) \\
&\quad + Pr(s \in \aleph : pr(s) \cap \Sigma^d \cap \aleph \neq \emptyset, \exists u \in pr(s), P^*_N(M(u)) \leq \rho) \\
&\leq Pr(s \in (K - \aleph) \cap \Sigma^{>d_1}) + \sum_{s \in \aleph \cap \Sigma^{d_1}} Pr(s)\phi_2 < \phi_1 + \phi_2 < \phi.
\end{aligned}$$

Therefore with the above choice of ρ, an arbitrary FA rate ϕ could be achieved. Next since (L, K) is S_m-Prognosable, according to Lemma 1, with this choice of ρ, for any $\tau > 0$, we have $P^{md} \leq Pr(s \in \partial : [\forall u \in s/\Sigma^{>m}, P^*_N(M(u)) > \rho] \vee [|s| \leq m]) < \tau$. Therefore the sufficiency holds.

(Necessity) To show the necessity, consider the contrapositive where (L, K) is not S_m-Prognosable. Then by Theorem 8, there are two possibilities. First, if $m \geq \ell(\partial)$, then let $s \in \partial$ be such that $|s| = \ell(\partial)$, and in which case,

$$P^{md} \geq Pr(s \in \partial : [|s| \leq m] \vee [\forall u \in s/\Sigma^{>m}, P^*_N(M(u)) > \rho]) \geq Pr(s \in \partial : |s| \leq m) \geq Pr(s).$$

Therefore a MD rate $\tau < Pr(s)$ can not be achieved.

On the other hand, if $m < \ell(\partial)$ but (5.5) is not true, then exists $s \in \partial$, such that for all $u \in s/\Sigma^{>m}$, there exists $u' \in \Upsilon$ with $M(u) = M(u')$. Since $u' \in \Upsilon$, according to Lemma 3, there exists $t' \in K\backslash u'$ such that $u't' \in \aleph$. If we choose $\rho < \min_{u \in s/\Sigma^{>m}} P_N^*(u)$, then s will be missed detected, and a MD rate $\tau < Pr(s)$ can not be achieved. On the other hand if we choose $\rho \geq \min_{u \in s/\Sigma^{>m}} P_N^*(u)$, then $u't'$ will be false alarmed, and a FA rate $\phi < Pr(u't')$ can not be met. Therefore in this case, at most one of arbitrarily small FA or MD rates can be achieved, completing the contraposition argument. □

5.3 Verification of Stochastic Prognosability

Having established S_m-Prognosability as a central property, needed for the existence of a m-prognoser, we next provide a polynomial algorithm for the verification of S_m-Prognosability utilizing Theorem 9. We need the following definitions that identify *m-steps interior* nonfault states from where no fault can occur within m steps but will occur at $(m+1)$th step, *indicator* nonfault states from where a future fault is inevitable with arbitrary confidence, and *nonindicator* nonfault states which are not indicator states.

Definition 10. Given a stochastic DES $G = (X, \Sigma, \alpha, x_0)$, deterministic nonfault specification $R = (Q, \Sigma, \beta, q_0)$, with their refinement $G^R = (X \times \overline{Q}, \Sigma, \gamma, (x_0, q_0))$, the set of

- *m-steps interior* nonfault states $\partial_m^-(X \times \overline{Q}) \subseteq X \times \overline{Q}$ (where $m \geq 0$) are states $(x, \overline{q})$ such that $\overline{q} \neq F$, and there exists $(x', \overline{q}')$ with $\overline{q}' = F$ and $s \in \Sigma^{m+1}$ s.t. $\gamma((x, \overline{q}), s, (x', \overline{q}')) > 0$ and for all $(x', \overline{q}')$, $s \in \Sigma^m$, $[\gamma((x, \overline{q}), s, (x', \overline{q}')) > 0] \Rightarrow [\overline{q}' \neq F]$;

- *indicator* nonfault states $\mathfrak{I}(X \times \overline{Q})$ are states $(x, \overline{q})$ such that $\overline{q} \neq F$ and from which the system can not reach a closed SCC in G^R that contains a nonfault state;

- *nonindicator* nonfault states $\Upsilon(X \times \overline{Q})$ are states from which the system can reach a closed SCC in G^R that contains a nonfault state.

The following lemma is immediate from Definition 8, Definition 10 and Lemma 3.

Lemma 4. Given a pair $(L = L(G), K = L(R))$ of closed regular languages with $K \subseteq L$, then for any $s \in K$,

- $[s \in \partial_m^-] \Leftrightarrow [\exists (x, \overline{q}) \in \partial_m^-(X \times \overline{Q}), \gamma((x_0, q_0), s, (x, \overline{q})) > 0]$;
- $[s \in \mathfrak{J}] \Leftrightarrow [\exists (x, \overline{q}) \in \mathfrak{J}(X \times \overline{Q}), \gamma((x_0, q_0), s, (x, \overline{q})) > 0]$;
- $[s \in \Upsilon] \Leftrightarrow [\exists (x, \overline{q}) \in \Upsilon(X \times \overline{Q}), \gamma((x_0, q_0), s, (x, \overline{q})) > 0]$.

The following algorithm verifies the condition of Theorem 9.

Algorithm 4. For a given stochastic automaton $G = (X, \Sigma, \alpha, x_0)$ and a deterministic nonfault specification $R = (Q, \Sigma, \beta, x_0)$, perform the following steps:

1) Check if the length of the shortest trace to a state $X \times \{F\}$ in G^R is smaller than m, if the answer is yes, proceed to step 2), otherwise (L, K) is not S_m-Prognosable;

2) Construct a testing automaton $T = G^R \times G^R$ such that at each step the first copy of G^R takes lead in executing transitions, whereas the second copy responds by executing an indistinguishable nonfault trace. This automaton is denoted as $T = (Z, \Sigma \times \overline{\Sigma}, \delta, z_0)$, where

   - $Z = X \times \overline{Q} \times X \times \overline{Q}$;
   - $z_0 = ((x_0, q_0), (x_0, q_0))$ is the initial state;
   - $\delta : Z \times \Sigma \times \overline{\Sigma} \times Z \to [0, 1]$ is defined as: $\forall ((x_1, \overline{q}_1), (x_2, \overline{q}_2)), ((x_1', \overline{q}_1'), (x_2', \overline{q}_2')) \in Z, (\sigma, \sigma') \in \Sigma \times \overline{\Sigma}$,

$$\delta(((x_1, \overline{q}_1), (x_2, \overline{q}_2)), (\sigma, \sigma'), ((x_1', \overline{q}_1'), (x_2', \overline{q}_2')))$$

$$= \begin{cases} \gamma((x_1, \overline{q}_1), \sigma, (x_1', \overline{q}_1')), & \text{if } (\sigma \in \Sigma_{uo}) \wedge (\sigma' = \epsilon) \\ & \quad \wedge ((x_2, \overline{q}_2) = (x_2', \overline{q}_2')) \wedge (\overline{q}_2' \neq F); \\ & \quad \wedge ((x_2, \overline{q}_2) = (x_2', \overline{q}_2')) \wedge (\overline{q}_2' \neq F); \\ \frac{\gamma((x_1, \overline{q}_1), \sigma, (x_1', \overline{q}_1'))\alpha(L_{G^R}((x_2, \overline{q}_2), \sigma', (x_2', \overline{q}_2')))}{\alpha(L_{G^R}((x_2, \overline{q}_2), M(\sigma)))}, & \text{if } (\sigma \in \Sigma - \Sigma_{uo}) \wedge (M(\sigma) = M(\sigma')) \\ & \quad \wedge (L_{G^R}((x_2, \overline{q}_2), \sigma', (x_2', \overline{q}_2'))) \neq \emptyset) \\ & \quad \wedge (\overline{q}_2' \neq F); \\ 0 & \text{otherwise.} \end{cases}$$

According to the definition of δ, when the first copy of G^R executes an unobservable event, the second copy responds by ϵ (since it observes nothing); if the first copy executes an observable event σ, then the second copy responds by executing a nonfault trace consisting of sequence of unobservable events followed by an observable event that has the same mask value as $M(\sigma)$. Note a conditioning is applied to limit the executions of the second copy to indistinguishable nonfault traces.

3) Check if every state $((x_1, \overline{q}_1), (x_2, \overline{q}_2))$ with $(x_1, \overline{q}_1) \in \partial_m^-(X \times \overline{Q})$ satisfies $(x_2, \overline{q}_2) \notin \Upsilon(X \times \overline{Q})$, (L, K) is S_m-Prognosable if and only if the answer is yes.

The following theorem guarantees the correctness of Algorithm 4.

Theorem 11. A pair $(L = L(G), K = L(R))$ of closed regular languages with $K \subseteq L$ is S_m-Prognosable if and only if any fault state can only be reached in more than m-steps in G^R and every reachable state $((x_1, \overline{q}_1), (x_2, \overline{q}_2))$ of T with $(x_1, \overline{q}_1) \in \partial_m^-(X \times \overline{Q})$ satisfies $(x_2, \overline{q}_2) \notin \Upsilon(X \times \overline{Q})$.

Proof. Obviously we have: any fault state can only be reached in more than m-steps if and only if $m < \ell(\partial)$. Next, by the construction of T, for any $s \in L$ and $s' \in K$, $M(s) = M(s')$ if and only if there exists $((x_1, \overline{q}_1), (x_2, \overline{q}_2))$ such that $\delta(((x_0, q_0), (x_0, q_0)), (s, s'), ((x_1, \overline{q}_1), (x_2, \overline{q}_2))) > 0$. So if every reachable state $((x_1, \overline{q}_1), (x_2, \overline{q}_2))$ with $(x_1, \overline{q}_1) \in \partial_m^-(X \times \overline{Q})$ satisfies $(x_2, \overline{q}_2) \notin \Upsilon(X \times \overline{Q})$, then by Lemma 4, every $s \in \partial_m^-$ is not ambiguous with any nonindicator trace, i.e., $M^{-1}M(\partial_m^-) \cap \Upsilon = \emptyset$. Therefore (L, K) is S_m-Prognosable according to Theorem 9, and the sufficiency follows. On the other hand, if the theorem's condition is not satisfied, then either $m \geq \ell(\partial)$ or there exists (s, s') with $M(s) = M(s')$ and $((x_1, \overline{q}_1), (x_2, \overline{q}_2))$ such that $(x_1, \overline{q}_1) \in \partial_m^-(X \times \overline{Q})$, $(x_2, \overline{q}_2) \in \Upsilon(X \times \overline{Q})$ and $\delta(((x_0, q_0), (x_0, q_0)), (s, s'), ((x_1, \overline{q}_1), (x_2, \overline{q}_2))) > 0$. i.e., $s \in \partial_m^-$ and $s' \in \Upsilon$. Therefore $M^{-1}M(\partial_m^-) \cap \Upsilon \neq \emptyset$. By Theorem 9, (L, K) is not S_m-Prognosable, which proves the necessity. □

Example 13. Let us revisit the system shown in Fig. 5.1. According to Definition 10, $\mathfrak{J}(X \times \overline{Q}) = \{(2,2), (3,3), (4,4)\}$, $\Upsilon(X \times \overline{Q}) = \{(0,0), (1,1)\}$, $\partial_1^-(X \times \overline{Q}) = \{(3,3)\}$ and $\partial_2^-(X \times \overline{Q}) = \{(2,2)\}$. It is easy to check that $1 < 2 < \ell(\partial) = 4$. The testing automaton is shown in Fig. 5.2.

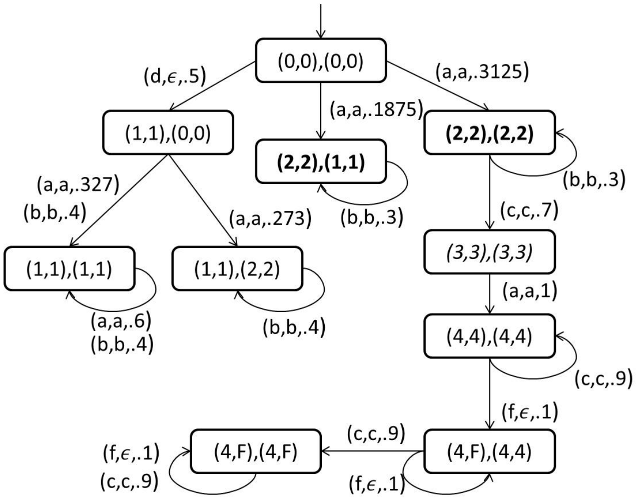


Figure 5.2 Testing automaton for Example 13.

The only state $((x_1, \overline{q}_1), (x_2, \overline{q}_2))$ such that $(x_1, \overline{q}_1) \in \partial_1^-(X \times \overline{Q})$ is labeled in *italic* and satisfies $(x_2, \overline{q}_2) \notin \Upsilon(X \times \overline{Q})$ and therefore (L, K) is S_1-Prognosable. All the states $((x_1, \overline{q}_1), (x_2, \overline{q}_2))$ such that $(x_1, \overline{q}_1) \in \partial_2^-(X \times \overline{Q})$ are labeled in bold, and there exists $((2, 2), (1, 1))$ such that $(2, 2) \in \partial_2^-(X \times \overline{Q})$ and $(1, 1) \in \Upsilon(X \times \overline{Q})$. Therefore (L, K) is not S_2-Prognosable. These are as expected from the discussion in Examples 10 and 11. □

Remark 16. In Algorithm 4. G^R has $O(|X| \times |Q|)$ states and $O(|X|^2 \times |Q| \times |\Sigma|)$ transitions, and the testing automaton $T = G^R \times G^R$ has $O(|X|^2 \times |Q|^2)$ states and $O(|X|^4 \times |Q|^2 \times |\Sigma|^2)$ transitions. The computation of transition probabilities in T requires solving the matrix equation (2.1) for each $\sigma \in \Sigma - \Sigma_{uo}$ with complexity that is cubic in the number of states in G^R and linear in the number of events in G^R, namely, $O(|X|^3 \times |Q|^3 \times |\Sigma|)$. Thus the complexity of constructing T is $O(|X|^4 \times |Q|^2 \times |\Sigma|^2 + |X|^3 \times |Q|^3 \times |\Sigma|)$. The shortest path to a fault state in G^R can be computed in $O(\sqrt{|X| \times |Q|} \times |X|^2 \times |Q| \times |\Sigma|)$ [96]. Identifying the set of m-steps interior nonfault states in G^R can be done linearly in the size of G^R, i.e., $O(|X|^2 \times |Q| \times |\Sigma|)$, and identifying the set of indicator nonfault states can be achieved by determining all the nonfault closed SCC in G^R using the algorithm in [97], which can be done in $O(|X|^3 \times |Q|^3)$. Therefore the overall complexity of Algorithm 4 is $O(|X|^4 \times |Q|^2 \times |\Sigma|^2 + |X|^3 \times |Q|^3 \times |\Sigma|)$, which is polynomial in the number of states and events. Further if G is also deterministic (besides R) so that G^R has a smaller number of transitions, namely, $O(|X| \times |Q| \times |\Sigma|)$, then the verification complexity reduces to $O(|X|^2 \times |Q|^2 \times |\Sigma|^2 + |X|^3 \times |Q|^3 \times |\Sigma|)$. Furthermore, if the mask is "projection-type", the complexity further reduces due to a reduction in the number of transitions in G^R, where each state can now only have at most $|\Sigma|$ outgoing transitions, and thus the $|\Sigma|^2$ term will get replaced by $|\Sigma|$ in the complexity expression. □

5.4 Illustrative Examples

In this section, two simple practical examples are given to illustrate our results.

5.4.1 "Crowd" Protocol

We consider the application of our results to the "Crowd" system, an anonymity protocol introduced in [98] that is used to protect the identity on the world-wide-web, which is recently studied in the stochastic DESs setting [99, 100]. When an user (called initiator) decides to send a message to a web-server without revealing itself as the originator of the message, the user routes the message through a *crowd* of users (possibly itself). When a user in the crowd receives a message, it either sends the message to the web-server or forwards the message to a user in the crowd (possibly itself). Then this protocol is considered to be secure in hiding the identity of the originator. However, there can be a number of *corrupted* users in the crowd which can leak the information of the origin of the message, and as is usually the case with the analysis of Crowd ([101]), we also assume that a corrupted user does not forward a message to others. The process is depicted in Fig. 5.3, where the size of the crowd is taken to be 7, the possible initiators are $\{1, 2\}$ and the corrupted user is $\{7\}$.

Now we consider the case when a user tries to send a message to the web-server and initiates a route, and it also monitors the routing of that message to avoid the message being received by a corrupted user. The corresponding automaton model is given as Fig. 5.4, where a new initial state "0" is added from where the two initiator nodes can be reached with equal probability. It is assumed that each user chooses among its forwarding successors with a uniform probability distribution. Suppose three of the forwarding actions can be observed with the observation labels as shown, whereas the unobservable transitions are unlabeled. A fault is defined as

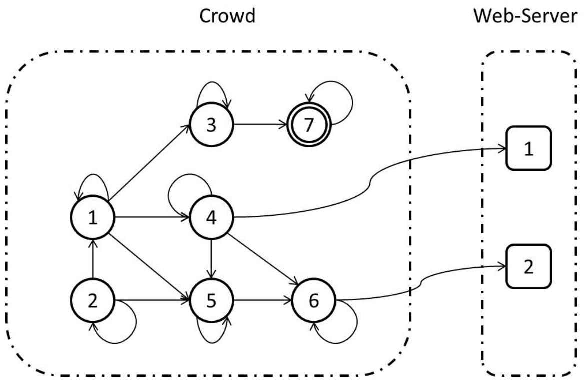


Figure 5.3 A crowd with size 7 and 2 initiators.

the forwarding of a message to the corrupted user “7”, i.e., the nonfault specification can be obtained by removing the corrupted user “7” and all associate transitions. It can be checked that under this observation mask, the system is not S_m-Prognosable for any $m \geq 0$, since for any fault trace reaching “7”, all its prefixes are ambiguous with a certain nonindicator trace. To make the monitoring process meaningful, a control policy can be applied so that the self-loop of state “4” is forbidden, i.e., after receiving a message, the user “4” can only forward it to the user “5”, “6” and web-server. Then one can verify that the system is now S_1-Prognosable. Note in this example, neither the monitor nor the control has any affect on the corrupted user, leaving the corrupted user unaware of the existence of the monitoring or control.

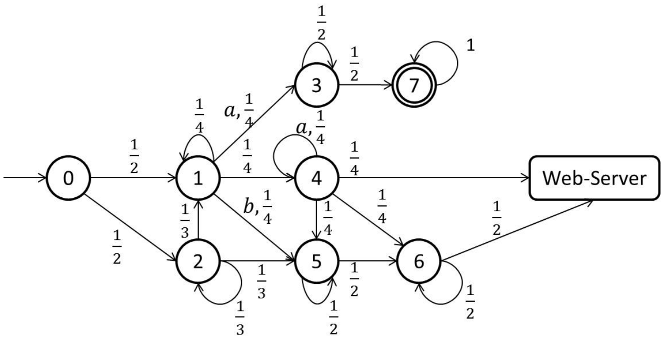


Figure 5.4 The automaton for the Crowd system in Fig. 5.3.

5.4.2 Prognosis of Stuck Faults in HVAC System

Consider the heating, ventilation and air conditioning (HVAC) system as examined in [74, 41], which is modeled as a stochastic DES consisting of four components: a pump, a valve, a controller and a flow sensor. The model is shown in Fig. 5.5, which has 24 states, 11 events and 36 transitions, and is initialized at state 1. Each event in the stochastic DES has two parts, the first of which describes the motion of the controller and the second of which indicates the output of the flow sensor ("F" denotes "there is flow" and "NF" denotes "there is no flow", while no output by the flow sensor is described as ϵ, which for simplicity is omitted in Fig. 5.5). The unobservable events are given by Σ_{uo} = {stuck_closed, stuck_open}, which are also the fault events Σ_f experienced by the controller; all other events are observed fully. The plant model shows the probability labels for each transitions. The deterministic nonfault specification is obtained by excluding all the states resulted by the fault events "stuck_closed" and "stuck_open", and is a subautomaton of the plant automaton, and without the probability labels (the definition of what constitutes a fault is independent of its occurrence probability). As can be seen, the shortest fault trace is "stuck_closed" itself which has a length of 1. Therefore the system can not be S_m-Prognosable with $m \geq 1$. One can check that in this example every nonfault trace has an extension reaching the absorbing nonfault state "24" and hence is a nonindicator. Therefore the system is not S_0-Prognosable. To achieve the S_0-Prognosability, one can exercise a control policy so that the system dynamics does not allow permanent idling by removing state "24" and adding a self-loop on state "10" with the same probability as transitioning to 24. Then one can verify by Algorithm 4 that the system is S_0-Prognosable.

5.5 Comparison With Related Concepts

In this section we will compare S_0-Prognosability with the notion of Prognosability in the logical setting [74, 75] and the notion of S-Diagnosability that are required for fault detection (as opposed to fault prediction) [47, 41, 43, 36]. To compare with the logical version of prognosability, we reproduce the definition from [75], specialized to centralized setting as follows:

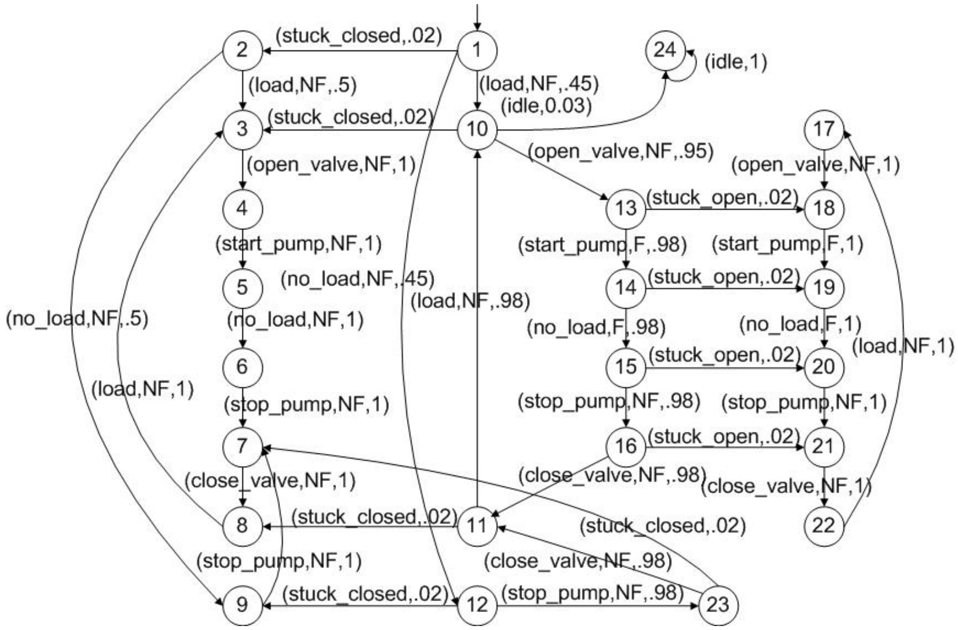


Figure 5.5 Automaton G for the HVAC system under prognosis.

Definition 11 ([75]). A pair (L, K) of closed languages with $K \subseteq L$ is said to be logically *Prognosable* if

$$(\forall s \in \partial)(\exists u \in s/\Sigma^{>0})(M^{-1}M(u) \cap K \subseteq \tilde{\mathfrak{J}}), \tag{5.10}$$

where $\tilde{\mathfrak{J}}$ denotes the set of logical indicators and is given by $\tilde{\mathfrak{J}} := \{s \in K : \exists n \in \mathbb{N}, L\backslash s \cap \Sigma^{\geq n} \subseteq [L - K]\backslash s\}$.

Remark 17. It is trivial to show that, for any $u \in K$, $(M^{-1}M(u) \cap K \subseteq \tilde{\mathfrak{J}}) \Leftrightarrow (P_N^*(M(u)) = 0)$. Therefore (5.10) can be equivalently written as:

$$Pr(s \in \partial : \forall u \in s/\Sigma^{>0}, P_N^*(M(u)) > 0) = 0.$$

Comparing then with the definition of S_m-Prognosability under $m = 0$, so (5.4) can be written as:

$$(\forall \tau, \rho > 0) Pr(s \in \partial : \forall u \in s/\Sigma^{>0}, P_N^*(M(u)) > \rho) < \tau.$$

It is obvious that if a system is logically Prognosable, then it is also S_0-Prognosable by definition. However the converse is not true. For example, the system shown in Fig. 5.1 is S_1-Prognosable and hence is S_0-Prognosable by Corollary 1. However, it is not Prognosable since $\forall s \in \partial, u \in s/\Sigma^{>0}$, $P^*_N(M(u)) > 0$. The stochastic version provides the flexibility of designing prognosers that can predict faults with arbitrary level of accuracy, which may be acceptable for certain applications even if 100% accuracy cannot be achieved (owing to lack of logical prognosability). Another artifact of this difference between the two notions is that, in logical setting, an indicator cannot visit a cycle of nonfault states, which can be restrictive, but in stochastic setting, an indicator can visit a cycle of nonfault states as long as the cycle does not form a closed SCC. In the example of Fig. 5.1, the prefix aca of the fault trace $acaf$ is an indicator that ends in a non-closed cycle of nonfault state $(4,4)$ in G^R. While this does not violate stochastic prognosability, it ends up violating logical prognosability. □

The next result shows that S_0-Prognosability is stronger than S-Diagnosability, meaning that whenever it is possible to predict faults, it is also possible to diagnose those, as can be expected.

Theorem 12. Given a pair $(L = L(G), K = L(R))$ of closed regular languages with $K \subseteq L$, if (L, K) is S_0-Prognosable, then it is S-Diagnosable. However, the converse need not hold.

Proof. We argue by contradiction. Assume (L, K) is S_0-Prognosable but not S-Diagnosable. Let $s \in L - K$ and $s' \in K$ with $M(s) = M(s')$ satisfy the condition in Theorem 2. Then for any $n \in \mathbb{N}$, $Pr(t : t \in K\backslash s' \cap \Sigma^n) = \sum_{o \in \Delta^*} Pr(t : t \in K\backslash s' \cap \Sigma^n, M(t) = o) = \sum_{o \in \Delta^*} Pr(t : t \in L\backslash s \cap \Sigma^n, M(t) = o) = Pr(t : t \in L\backslash s \cap \Sigma^n) = 1$, (the second equality follows from the condition in Theorem 2). Since $\forall n$, $Pr(t : t \in K\backslash s' \cap \Sigma^n) = 1$, it follows from the definition of Υ that $s' \in \Upsilon$ (we can choose $\rho < 1$ to satisfy the definition of Υ). Considering $s \in L - K$ and $M(s) = M(s')$, we have $s \in M^{-1}M(\Upsilon) \cap L - K$, which is contradictory to Corollary 2 since (L, K) is S_0-Prognosable.

To see that the converse need not hold, we consider the system shown in Fig. 5.6, where $\Sigma = \{a, b, c, f\}$, $\Sigma_{uo} = \{b, f\}$ and for $\sigma \in \Sigma - \Sigma_{uo}$, $M(\sigma) = \sigma$. After the occurrence of fault trace af, the only observations that can be produced are the traces in c^+, which are distinguishable from any nonfault trace, and so (L, K) is S-Diagnosable. However, since $M^{-1}M(\partial_0^-) \cap \Upsilon = M^{-1}M(a) \cap \{\epsilon, ba^*\} = \{ba\} \neq \emptyset$, by Theorem 9, (L, K) is not S_0-Prognosable. □

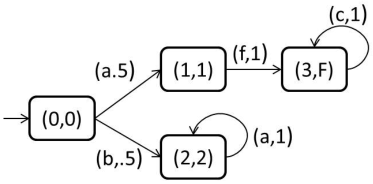


Figure 5.6 Refined system G^R for the proof of Theorem 12

5.6 Conclusion

In this chapter, we studied the prognosis of fault, i.e., its prediction prior to its occurrence, for stochastic discrete event systems. We formulated the notion of S_m-Prognosability for stochastic DESs, generalizing the corresponding notion from the logical setting [74, 75], and showed that it is a necessary and sufficient condition for the existence of a prognoser that can predict a fault at least m-steps prior to its occurrence, while achieving any arbitrary false alarm and missed detection rates. (Higher accuracy of prognostic decision can be obtained by allowing shorter reaction bound.) A polynomial complexity algorithm for the verification of S_m-Prognosability was also provided, which checks on a pair of indistinguishable traces for the reachability of a pair of states, one of which is a m-steps interior nonfault state and the other is a nonindicator state (such a pair is reachable if and only if S_m-Prognosability does not hold). The contribution of the work was further emphasized by comparing with previous related work on fault diagnosability, which was shown to be a weaker requirement than fault prognosability, as can be expected. There are several directions for future research: 1) An online recursive prognosis algorithm to compute the state distribution $\pi(o)$ resulted by an observation o so as to be able to predict a fault by checking whether $P_N^*(o) \leq \rho$, which in turn implies if $\pi(o)$ itself falls within a suitable range, and 2) algorithms for computing the decision threshold ρ and the largest possible reaction bound m for given performance requirements $\phi, \tau > 0$ for FA and MD rates. Also, an extension to the decentralized setting would be another direction for future work.

CHAPTER 6. SUMMARY AND DISCUSSION

6.1 Summary of Dissertation

In this dissertation, we studied fault diagnosis and prognosis of stochastic discrete-event and cyber-physical systems. The main contributions of this dissertation are summarized as follows.

1. An online detector for stochastic DESs based on recursive likelihood computation was proposed, which has a quadratic complexity for the *online* monitoring, likelihood computation and issuing decision upon the arrival of a new observation. The algorithm for *offline* computing the detector parameters of detection threshold and delay bound so as to achieve a given performance requirement of false alarm and missed detection rates were presented, using a proposed procedure for constructing an extended observer. The extended observer computes, for each observation sequence, the set of states reached in the system model, along with their probabilities and the number of post-fault transitions executed.

2. The existence of aforementioned detector to achieve any arbitrary performance requirement was shown to be equivalent to the S-Diagnosability property. And so the aforementioned algorithms are guaranteed to terminate for S-Diagnosable systems and upper bounds on the number of iterations prior to termination were provided. The complexity for the computation of the detector parameters, namely, detection threshold and delay bound requires constructing an extended observer whose size is exponential in the depth of the observer tree constructed, while the depth of the tree is a complex function of the system and specification models, the observation mask, and the desired bounds on MD and FA rates.

3. For a non-S-Diagnosable system an arbitrary performance is achievable only for the FA rate, whereas a lower bound exists for the achievable MD rate that is a function of the FA rate, and increases as FA rate is made more stringent by decreasing it. A variant of the algorithm for the S-Diagnosable case was used to compute an upper bound for the minimum achievable missed detection rate for a non-S-Diagnosable system.

4. We proposed the notion of input-output stochastic hybrid automaton (I/O-SHA), extending its logical counterpart by allowing randomness in invariants, guards, data updates, and output assignments.

5. We presented a method to refine a given discrete-time stochastic system against a *deterministic* (LTL) specification (one that can be accepted by a deterministic Büchi automaton), where the refinement is an I/O-SHA with the property that the violation of the LTL specification can be captured as a reachability property, and the probability of specification violation versus no violation can be estimated via a state estimation computation in the I/O-SHA model.

6. We provided a procedure to recursively compute the probability of fault versus no-fault (specification violation versus no-violation), which is used as a statistic for issuing detection decisions. The performance of the proposed fault detection procedure is measured in terms of FA and MD rates. The notion of S-Diagnosability to capture the capability of detecting faults in a timely manner, within a bounded delay, and with any desired level of accuracy in terms of missed detection and false alarm rates, was also proposed.

7. For fault prognosis problem, we proposed the notion of S_m-Prognosability, which requires for any tolerance level ρ and error bound τ, there exists a reaction bound $k \geq m$, such that the set of fault traces for which a fault cannot be predicted k steps in advance with tolerance level ρ, occurs with probability smaller than τ. A *polynomial* algorithm for testing S_m-Prognosability was also presented.

8. We formalized the notion of a prognoser that maps observations to decisions by comparing a suitable statistic with a threshold, and show that S_m-Prognosability is a necessary and

sufficient condition for the existence of a prognoser with reaction bound at least m (i.e., prediction at least m-steps prior to the occurrence of a fault) that can achieve any specified FA and MD rate requirement.

6.2 Future Work

1. The verification of the diagnosability property for the stochastic hybrid systems presented in Chapter 4 remains open at this time. The diagnosability property, which ensure the existence of a online detector for any arbitrary false alarm and missed detection rate requirement, should be checked before a designer can specify a performance requirement. By the setting in Chapter 4, the verification of aforementioned diagnosability property will reduce to certain reachability property which, as demonstrated in [63, 66], might be solved by dynamic programming.

2. Another future direction is the adoption of probabilistic model checking technique for verification of diagnosability property for the stochastic hybrid systems presented in Chapter 4. Statistical model checking aims to verify whether a system that exhibits stochastic behavior satisfy certain (quantitative) property, [102, 103]. One example of such quantitative property is expressed in PCTL (Probabilistic Computation Tree Logic) [104], where the key operator in PCTL is $P_{>p}[\phi]$, which means that a path formula ϕ is true in a state with probability satisfying $> p$. Given a systems S and a path formula ϕ, there are two questions that the probabilistic model checking is trying to answer: 1), whether the probability that ϕ is true by S is satisfying $> p$, i.e., $S \models P_{>p}[\phi]$; or 2) what is the exact probability that S satisfies ϕ, i.e., what is the value of $P[S \models \phi]$. Two types of approaches to address the probabilistic model checking problem have been developed, namely *numerical* and *statistical*, the first of which iteratively computes or approximates the exact probability of paths satisfying ϕ by exploring the whole state space of S, while the second of which is to simulate the system for finitely number of times, and borrow techniques/theories from statistics to provide statistical inference of the answer to the probabilistic model checking problem. One assumption of statistical model checking is

that, ϕ is a bounded property, i.e., a property that can be verified based on state trace of finite length. Let B_i be a Bernoulli random variable with distribution $Pr[B_i = 1] = \theta$ and $Pr[B_i = 0] = 1 - \theta$, and B_i is such that it equals 1 if the ith simulation of S satisfies ϕ and 0 otherwise. Therefore $\theta = P[S \models \phi]$. Then the statistical model checking aims to check 1) whether $\theta > p$, and 2) estimate the exact value of θ, by borrowing theoretical results from the field of statistics.

3. In current work, the LTL formula used to specify the correctness requirement is restricted to the fragment that has *deterministic* Büchi accepter. This constraint was caused by the recursive computation (4.7)-(4.9), reproduced as following:

$$\begin{aligned}
p_{k|k}(d|z^k, l_k) &= \frac{h_{l_k}(y_k|d, u_k)p_{k|k-1}(d|z^{k-1}, l_k)}{\int_D h_{l_k}(y_k|d_k, u_k)p_{k|k-1}(d_k|z^{k-1}, l_k)d(d_k)} \\
\pi_{k+1}(l|z^k) &= \sum_{l_k \in L} \pi_k(l_k|z^{k-1}) \times \int_{D(l_k \to l|u_k)} p_{k|k}(d_k|z^k, l_k)d(d_k) \\
p_{k+1|k}(d|z^k, l_{k+1}) &= \frac{1}{\pi_{k+1}(l_{k+1}|z^k)} \sum_{l_k} \pi_k(l_k|z^{k-1}) \\
&\quad \times \int_{D(l_k \to l_{k+1}|u_k)} f_{l_{k+1}}(d|d_k, u_k)p_{k|k}(d_k|z^k, l_k)d(d_k),
\end{aligned}$$

In particular, allowing nondeterminism in the specification model R will introduce a redundancy in the computation of $\pi_k(l|z^k)$, which propagates as the recursion of the computation. A future direction would be to remove such constraint so as the whole class of LTL formula can be considered.

4. We are also interested in the application of the presented work to rumor localization, or source identification problem, in social networks (SN). A social network (SN) consists of a set of individuals which are connected by social relations [105, 106], which could be "friends", "spouse" or "tendency to forward a news", and could be represented by a matrix called sociomatrix which is square and with binary elements. If a social network is with more than one kind of social relation, then for each social relation there is one sociomatrix. For a SN with stochastic process, then for a given kind of social relation, there could be more than one sociomatrix (denoted as $x \in X$), and each one has a probability $P(x)$. The problem of interest is formulated as following: Suppose we have a

social network with n individuals, connected by k kinds of social relations, one of which represents how a rumor is propagating. When we observe a rumor over the SN with partial/unreliable observations, how determine which individual initializes the rumor? A similar problem was formulated and solved in [107], where the network is an undirected graph and rumors can spread between connected nodes. Once a node is infected then it could not be recovered, which is known as *susceptible-infected* (SI) model. Once a node is infected, the time it takes to spread the rumor to its neighbor obeys exponential distribution with parameter λ and in the paper it is assumed $\lambda = 1$, i.e., the transition probability is homogeneous and all nodes connected to a same infected node are equal likely to be infected. The observer examines the network at certain time t and knows the subgraph formed by all infected nodes and the research problem is to determine (in a *off-line* fashion) which node in the subgraph is blamed for the rumor. The authors of [107] propose Maximal Likelihood Estimator for this problem and various properties of the estimator are also studied with respect to many kinds of network, e.g., regular trees, irregular trees and general graphs. The difference between the problem in [107] and the one described above is that: 1) in [107] there is only one message (the rumor), spreading in the network, i.e., there is only one social relation, and 2) and by the time t all infected nodes are available to the observer, thus there is no hidden information or partial observability in the problem of [107].

5. The diagnosability/prognosability property of a system is essentially capturing the capability of system to leak some information to some trustful observer. A converse problem, i.e., the secrecy property as in [108, 109], examines the capability of system to hide some information to any untrusting observer. A more realistic scenario is, when a system is simultaneously observed by both desirable and undesirable observers, and the system is required to deliver information to certain trustful observer while blocking the untrusting observers from accessing the system information. Also interested as a future work is to examine the secrecy property in cyber-physical system, where a cyber component interacts with a physical component [110].

6. We have studied the diagnosability and prognosability properties of a given stochastic system. Now when there is flexibility to exercise control in order to meet some correctness specification [57, 58, 59, 60, 61], additional performance specifications, including diagnosability or prognosability may be added. So designing control strategies for correctness as well performance specifications of diagnosability/prognosability remains a future research direction.

APPENDIX A. ADDITIONAL PROOFS FOR CHAPTER 3

Proof for Theorem 1: Let s' be the minimal length trace that has different probabilities in A_1 and A_2. Denote $p_1 = \alpha_{A_1}(s')$, $p_2 = \alpha_{A_2}(s')$ and $|s'| = n'$. Assume w.l.o.g. that $p_1 > p_2$. Let s be our observation where $|s| = n = Kn'$. Divide s into K pieces, each of which has length equal to n'. Count the number of pieces whose observation is exactly s', denoting this number as k, and denote the proportion as $\hat{p}_n = \frac{k}{K}$. Let H_i be the hypothesis that s is generated by A_i, $i = 1, 2$. To determine which hypothesis is correct, consider the likelihood ratio $L(H_2|\hat{p}_n)$

$$\begin{aligned} L(H_2|\hat{p}_n) &= \frac{Pr(\hat{p}_n|H_2)}{Pr(\hat{p}_n|H_1)} \\ &= \frac{\begin{pmatrix} K \\ \hat{p}_n K \end{pmatrix} (p_2)^{\hat{p}_n K}(1-p_2)^{K-\hat{p}_n K}}{\begin{pmatrix} K \\ \hat{p}_n K \end{pmatrix} (p_1)^{\hat{p}_n K}(1-p_1)^{K-\hat{p}_n K}} \\ &= \frac{(p_2)^{\hat{p}_n K}(1-p_2)^{K-\hat{p}_n K}}{(p_1)^{\hat{p}_n K}(1-p_1)^{K-\hat{p}_n K}}. \end{aligned}$$

Taking the logarithm of the likelihood function yields

$$\begin{aligned} \log L(H_2|\hat{p}_n) &= \hat{p}_n K \log\left(\frac{p_2}{p_1}\right) + (K - \hat{p}_n K)\log\left(\frac{1-p_2}{1-p_1}\right) \\ &= K\left\{\log\left(\frac{1-p_2}{1-p_1}\right) + \hat{p}_n\left(\log\left(\frac{p_2}{p_1}\right) - \log\left(\frac{1-p_2}{1-p_1}\right)\right)\right\}. \end{aligned}$$

As n increases (or equivalently K increases), $\log L(H_2|\hat{p}_n)$ decreases large as well, as long as $\log\left(\frac{1-p_2}{1-p_1}\right) + \hat{p}_n\left(\log\left(\frac{p_2}{p_1}\right) - \log\left(\frac{1-p_2}{1-p_1}\right)\right) < 0$ is satisfied, which is the case when

$$\hat{p}_n > \frac{-\log\left(\frac{1-p_2}{1-p_1}\right)}{\log\left(\frac{p_2}{p_1}\right) - \log\left(\frac{1-p_2}{1-p_1}\right)} = r.$$

Note that $P_1 = \{p_1, 1-p_1\}$ and $P_2 = \{p_2, 1-p_2\}$ are two different probability distributions. According to Gibbs' inequality we have

$$p_1 \log p_1 + (1-p_1)\log(1-p_1) > p_1 \log p_2 + (1-p_1)\log(1-p_2),$$

which is equivalent to

$$\log\left(\frac{p_1}{p_2}\right) > \frac{1-p_1}{p_1}\log\left(\frac{1-p_2}{1-p_1}\right).$$

Therefore

$$r \;=\; \frac{\log\left(\frac{1-p_2}{1-p_1}\right)}{\log\left(\frac{p_1}{p_2}\right)+\log\left(\frac{1-p_2}{1-p_1}\right)} < \frac{\log\left(\frac{1-p_2}{1-p_1}\right)}{\frac{1-p_1}{p_1}\log\left(\frac{1-p_2}{1-p_1}\right)+\log\left(\frac{1-p_2}{1-p_1}\right)} = p_1.$$

Similarly we can show that $r > p_2$.

Now if s is generated by A_1 then by the law of large numbers we have

$$(\forall\tau > 0)(\exists n_1 \in \mathbb{N})(\forall s \text{ generated by } A_1 \wedge n > n_1)Pr(|\hat{p}_n - p_1| > p_1 - r) < \tau.$$

Therefore $Pr(\hat{p}_n < r) < \tau$. For any $0 < \rho < 1$, choose $n_2 \in \mathbb{N}$ such that $(n > n_2) \wedge (\hat{p}_n > r) \Rightarrow \log L(H_2|\hat{p}_n) < \log\frac{\rho}{1-\rho} \Rightarrow L(H_2|\hat{p}_n) < \frac{\rho}{1-\rho} \Rightarrow Pr(H_2|\hat{p}_n) = \frac{Pr(\hat{p}_n|H_2)}{Pr(\hat{p}_n|H_2)+Pr(\hat{p}_n|H_1)} < \rho$. Let $n = \max(n_1, n_2)$. Then if s is generated by A_1, $|s| > n$ and $\hat{p}_n > r$, then $Pr(H_2|\hat{p}_n) < \rho$, i.e., $Pr(s_2|M(s_1) = M(s_2)) < \rho$. Therefore

$$Pr(s_1 : |s_1| > n, Pr(s_2|M(s_1) = M(s_2)) > \rho) < Pr(s_1 : |s_1| > n, \hat{p}_n < r) < \tau. \qquad \square$$

Proof of Theorem 2: (Sufficiency) If (3.4) is true, then for any extension t of s, let $U_{st} := M^{-1}M(st) \cap L - \{s, s'\}\Sigma^*$, and for any extension t' of s' such that $s't' \in K \cap M^{-1}M(st)$, let $U_{st}^K := M^{-1}M(st) \cap K - \{s't'\}$. Then we have

$$\begin{aligned}
P_N(st) &= \frac{Pr(s't') + Pr(U_{st}^K)}{Pr(s't') + Pr(st) + Pr(U_{st})}\\
&\geq \frac{Pr(s't')}{Pr(s't') + Pr(st) + Pr(U_{st})}\\
&= \frac{Pr(s')Pr(t')}{Pr(s')Pr(t') + Pr(s)Pr(t) + Pr(U_{st})}\\
&= \frac{Pr(s')}{Pr(s') + Pr(s) + Pr(U_{st})/Pr(t)}
\end{aligned}$$

$$\geq \frac{Pr(s')}{Pr(s')+Pr(s)+Pr(U_{st})/Pr(st)}.$$

Note that the third equality utilizes the fact that s and s' satisfy (3.4) and so $Pr(t) = Pr(t')$. Now consider the extensions of any trace in U_{st}. According to (3.4), for any trace in U_{st}, its extensions are not equally distributed as the extensions of s. By applying Theorem 1, one can conclude that for any $\rho' > 0, \tau' > 0$, there exists $n' \in \mathbb{N}$ such that $Pr(t : t \in L\backslash s, |t| \geq n', Pr(U_{st})/Pr(st) > \rho') < \tau'$, i.e., $Pr(t : t \in L\backslash s, |t| \geq n', Pr(U_{st})/Pr(st) \leq \rho') \geq 1 - \tau'$. When $Pr(U_{st})/Pr(st) \leq \rho'$, we have

$$P_N(st) \geq \frac{Pr(s')}{Pr(s')+Pr(s)+Pr(U_{st})/Pr(st)} \geq \frac{Pr(s')}{Pr(s')+Pr(s)+\rho'}.$$

Now consider fixed $\tau' > 0$ and $\rho' > 0$, and $t \in L\backslash s$ with $|t| \geq n'$ satisfying $Pr(U_{st})/Pr(st) \leq \rho'$. One can conclude that, with at least $(1-\tau')$ probability that the extensions of s would have P_N value larger than $\frac{Pr(s')}{Pr(s')+Pr(s)+\rho'}$. Let $0 < \rho < Pr(s')/(Pr(s)+Pr(s'))$ and $0 < \tau < 1-\tau'$. Then we have:

$$(\forall n \in \mathbb{N}) Pr(t : t \in L\backslash s, |t| \geq n, P_N(st) > \rho) \geq 1 - \tau' > \tau.$$

It follows that the system is not S-Diagnosable.

(Necessity) If (3.4) is not true, then for all indistinguishable pairs of fault and nonfault traces (s, s'), there exists a future observation that has different probability of being fault versus nonfault, i.e.,

$$(\forall s \in L-K, s' \in K \text{ s.t. } M(s) = M(s'))(\exists o \in \Delta^*)$$
$$Pr(t : t \in L\backslash s, M(t) = o) \neq Pr(t : t \in K\backslash s', M(t) = o).$$

Then according to the likelihood ratio test presented in Theorem 1, after the occurrence of any fault trace, by comparing the number of occurrences of the minimal segment of observations that has different probability of being fault versus nonfault, the ambiguity of the occurrence of a fault decreases as the length of extension increases, i.e., there exists $n \in \mathbb{N}$ such that for all $\rho > 0$, $\tau > 0$ and $s \in L-K$, the extensions of s longer than n and having P_N larger than ρ occur with probability smaller than τ, i.e.,

$$(\forall \tau > 0 \wedge \forall \rho > 0)(\exists n \in \mathbb{N})(\forall s \in L-K)$$

$$Pr(t : t \in L\backslash s, |t| \geq n, P_N(st) > \rho) < \tau.$$

Thus we can conclude that the system is S-Diagnosable. □

Proof for Theorem 4: According to the proof of Theorem 3, for $i \in \{1, 2, 3\}$, there exists m_i such that ρ_i obtained by examining traces in K_i shorter than m_i ensures the FA rate of K_i be smaller than ϕ_i. Since $\phi_2 = 0$ (none of the traces in K_2 are false-alarmed because no decision is issued for those traces), by choosing ϕ_1 and ϕ_3 such that $\phi_1 + \phi_3 \leq \phi$, the requirement of the specified FA rate is met. It follows that Algorithm 1 is guaranteed to terminate with tree depth $d_1 \leq \max_i m_i$, returning a threshold $\rho_D \leq \min_i \rho_i$ such that the overall FA rate is upper bounded by ϕ. □

Proof for Theorem 5: In the tree of Algorithm 2, a node is deemed a leaf if the "F" decision is made upon reaching it, and otherwise the tree itself is terminated at a uniform depth so that the upper bound for the MD rate has dropped below the requirement τ. Expand (3.7) and we have $\overline{P_D^{md}} = \sum_{\overline{z}\in\overline{Z}_m:P_N(\overline{z})>\rho_D} \sum_{((x,\overline{q}),p,n)\in\overline{z}:(x,\overline{q})\in Y_1} p + \sum_{\overline{z}\in\overline{Z}_m:P_N(\overline{z})>\rho_D} \sum_{((x,\overline{q}),p,n)\in\overline{z}:\overline{q}=F} p$. Similar to the proof of Theorem 3, the nonfaulty-ness in K_1 is a transient property, and so for any $\tau_1 > 0$, there exists $m' \in \mathbb{N}$ such that $Pr(s \in K \cap pr(L - K) : |s| \geq m') < \tau_1$, and hence the first term on the RHS is less than τ_1. For S-Diagnosable systems, according to Theorem 3, for any $\tau_2 > 0$ there exists n'_D such that with this choice of delay bound, the second term on the RHS is less than τ_2. Therefore by choosing τ_1 and τ_2 such that $\tau_1 + \tau_2 \leq \tau$, Algorithm 2 is guaranteed to terminate with tree depth $d_2 \leq m' + n'_D$, returning a delay bound $n_D = 1 + \max_{((x,\overline{q}),p,n)\in z,\overline{z}\in\overline{Z}} n$ such that the overall MD rate is upper bounded by τ. □

APPENDIX B. DERIVATION OF EQUATIONS (4.7)-(4.9)

Here we derive the equations (4.7)-(4.9) in Chapter 4. According to the definition, we have

$$
\begin{aligned}
p_{k+1|k}(d|z^k, l_{k+1}) &= \frac{Pr(d_{k+1} = d, z^k, l_{k+1})}{Pr(z^k, l_{k+1})} \\
p_{k|k}(d|z^k, l_k) &= \frac{Pr(d_k = d, z^k, l_k)}{Pr(z^k, l_k)} \\
\pi_{k+1}(l|z^k) &= \frac{Pr(l_{k+1} = l, z^k)}{\sum_{l \in L} Pr(l_{k+1} = l, z^k)}.
\end{aligned}
$$

Therefore we have

$$
\begin{aligned}
p_{k|k-1}(d|z^{k-1}, l_k) &= \frac{Pr(d_k = d, z^{k-1}, l_k)}{Pr(z^{k-1}, l_k)} \\
\pi_k(l|z^{k-1}) &= \frac{Pr(l_k = l, z^{k-1})}{\sum_{l \in L} Pr(l_k = l, z^{k-1})}.
\end{aligned}
$$

Combining $p_{k|k}(d|z^k, l_k)$ and $p_{k|k-1}(d|z^{k-1}, l_k)$, we obtain:

$$
\begin{aligned}
&= p_{k|k}(d|z^k, l_k) \\
&= \frac{Pr(d_k = d, z^k, l_k)}{Pr(z^k, l_k)} \\
&= \frac{Pr(d_k = d, z^{k-1}, l_k, (u_k, y_k))}{Pr(z^{k-1}, l_k, (u_k, y_k))} \\
&= \frac{Pr(d_k = d, z^{k-1}, l_k) Pr(y_k | d_k = d, u_k, l_k)}{\sum_{d_k \in D} Pr(d_k, z^{k-1}, l_k) Pr(y_k | d_k, u_k, l_k)} \\
&= \frac{\frac{Pr(d_k = d, z^{k-1}, l_k)}{Pr(z^{k-1}, l_k)} Pr(y_k | d_k = d, u_k, l_k)}{\sum_{d_k \in D} \frac{Pr(d_k, z^{k-1}, l_k)}{Pr(z^{k-1}, l_k)} Pr(y_k | d_k, u_k, l_k)} \\
&= \frac{p_{k|k-1}(d|z^{k-1}, l_k) Pr(y_k | d_k = d, u_k, l_k)}{\sum_{d_k \in D} p_{k|k-1}(d_k|z^{k-1}, l_k) Pr(y_k | d_k, u_k, l_k)} \\
&= \frac{p_{k|k-1}(d|z^{k-1}, l_k) h_{l_k}(y_k|d, u_k)}{\int_D p_{k|k-1}(d_k|z^{k-1}, l_k) h_{l_k}(y_k|d_k, u_k) d(d_k)},
\end{aligned}
$$

i.e.,

$$
p_{k|k}(d|z^k, l_k) = \frac{p_{k|k-1}(d|z^{k-1}, l_k) h_{l_k}(y_k|d, u_k)}{\int_D p_{k|k-1}(d_k|z^{k-1}, l_k) h_{l_k}(y_k|d_k, u_k) d(d_k)}.
$$

Thus we have shown (4.7). Next by combining $\pi_{k+1}(l|z^k)$, $\pi_k(l|z^{k-1})$ and $p_{k|k}(d|z^k, l_k)$, we have:

$$\begin{aligned}
\pi_{k+1}(l|z^k) &= \frac{Pr(l_{k+1}=l, z^k)}{\sum_{l\in L} Pr(l_{k+1}=l, z^k)} \\
&= Pr(l_{k+1}=l|z^k) \\
&= \sum_{l_k\in L}\sum_{d_k\in D} Pr(l_k, l_{k+1}=l, d_k|z^k) \\
&= \sum_{l_k\in L}\sum_{d_k\in D} Pr(l_{k+1}=l, d_k|l_k, z^k)Pr(l_k|z^k) \\
&= \sum_{l_k\in L}\sum_{d_k\in D(l_k\to l|u_k)} Pr(d_k|l_k, z^k)Pr(l_k|z^{k-1}) \\
&= \sum_{l_k\in L}\pi_k(l_k|z^{k-1})\int_{D(l_k\to l|u_k)} p_{k|k}(d_k|z^k, l_k)d(d_k)
\end{aligned}$$

Thus we have established (4.8). Finally combining $p_{k+1|k}(d|z^k, l_{k+1})$, $\pi_{k+1}(l|z^k)$, $\pi_k(l|z^{k-1})$ and $p_{k|k}(d|z^k, l_k)$ yields:

$$\begin{aligned}
p_{k+1|k}(d|z^k, l_{k+1}) &= \frac{Pr(d_{k+1}=d, z^k, l_{k+1})}{Pr(z^k, l_{k+1})} \\
&= \frac{Pr(d_{k+1}=d, l_{k+1}|z^k)Pr(z^k)}{Pr(z^k, l_{k+1})} \\
&= \frac{1}{\pi_{k+1}(l_{k+1}|z^k)}Pr(d_{k+1}=d, l_{k+1}|z^k) \\
&= \frac{1}{\pi_{k+1}(l_{k+1}|z^k)}\sum_{l_k\in L}\sum_{d_k\in D} Pr(d_{k+1}=d, d_k, l_k, l_{k+1}|z^k) \\
&= \frac{1}{\pi_{k+1}(l_{k+1}|z^k)}\sum_{l_k\in L}\sum_{d_k\in D} Pr(d_{k+1}=d, d_k, l_{k+1}|l_k, z^k)Pr(l_k|z^k) \\
&= \frac{1}{\pi_{k+1}(l_{k+1}|z^k)}\sum_{l_k\in L}\sum_{d_k\in D} Pr(l_k|z^{k-1}) \\
&\qquad Pr(d_{k+1}=d, |d_k, l_{k+1}, l_k, z^k)Pr(d_k, l_{k+1}|l_k, z^k) \\
&= \frac{1}{\pi_{k+1}(l_{k+1}|z^k)}\sum_{l_k\in L}\pi_k(l_k|z^{k-1})\sum_{d_k\in D(l_k\to l_{k+1}|u_k)} \\
&\qquad Pr(d_{k+1}=d, |d_k, l_{k+1}, l_k, z^k)Pr(d_k, l_{k+1}|l_k, z^k) \\
&= \frac{1}{\pi_{k+1}(l_{k+1}|z^k)}\sum_{l_k\in L}\pi_k(l_k|z^{k-1})\sum_{d_k\in D(l_k\to l_{k+1}|u_k)} \\
&\qquad f_{l_{k+1}}(d|d_k, u_k)Pr(d_k|l_k, z^k) \\
&= \frac{1}{\pi_{k+1}(l_{k+1}|z^k)}\sum_{l_k\in L}\pi_k(l_k|z^{k-1}) \\
&\qquad \int_{D(l_k\to l_{k+1}|u_k)} f_{l_{k+1}}(d|d_k, u_k)p_{k|k}(d_k|z^k, l_k)d(d_k)
\end{aligned}$$

i.e,

$$\begin{aligned} p_{k+1|k}(d|z^k, l_{k+1}) &= \frac{1}{\pi_{k+1}(l_{k+1}|z^k)} \sum_{l_k \in L} \pi_k(l_k|z^{k-1}) \\ &\qquad \int_{D(l_k \rightarrow l_{k+1}|u_k)} f_{l_{k+1}}(d|d_k, u_k) p_{k|k}(d_k|z^k, l_k) d(d_k) \end{aligned}$$

Thus we have also established (4.9).

APPENDIX C. ADDITIONAL MATERIAL FOR CHAPTER 5

The following lemma is needed in the sufficiency proof of Theorem 10.

Lemma 5. For a pair (L, K) of S_m-Prognosable closed regular languages with $K \subseteq L$, we have

$$(\forall \rho', \phi > 0)(\exists d \in \mathbb{N})(\forall s \in \aleph) Pr(t : t \in \aleph \backslash s : |t| \geq d, P_N^*(M(st)) < \rho') < \phi, \tag{C.1}$$

where the persistent nonfault traces $\aleph$ is defined in Definition 8 and P_N^* is as defined by (5.1) and (5.2).

Proof. Since $P_N^*(M(st)) < \rho'$ if and only if $1 - P_N^*(M(st)) > 1 - \rho'$, letting $\rho := 1 - \rho'$, (C.1) is true if and only if

$$(\forall \rho, \phi > 0)(\exists d \in \mathbb{N})(\forall s \in \aleph) Pr(t : t \in \aleph \backslash s : |t| \geq d, 1 - P_N^*(M(st)) > \rho) < \phi. \tag{C.2}$$

Thus showing (C.1) is equivalent to showing that (C.2) holds. Next we show that (C.2) is equivalent to showing that the pair $(K, K - \aleph)$ is S-Diagnosable. First note that for any $st \in \aleph \subseteq \Upsilon$, it holds that,

$$\begin{aligned} P_N^*(M(st)) &= \frac{\min_{n \in \mathbb{N}} Pr(\{M^{-1}M(st) \cap K\}\Sigma^n \cap K)}{Pr(M^{-1}M(st) \cap L)} \\ &= \frac{Pr(M^{-1}M(st) \cap \aleph)}{Pr(M^{-1}M(st) \cap L)} \\ &= \frac{Pr(M^{-1}M(st) \cap \aleph)}{Pr(M^{-1}M(st) \cap K)}, \end{aligned}$$

where we have used the fact that (L, K) is S_m-Prognosable and so for $st \in \Upsilon$, $M^{-1}M(st) \cap L = M^{-1}M(st) \cap [K \cup (L - K)] = M^{-1}M(st) \cap K$ (follows from Corollary 2). Then,

$$1 - P_N^*(M(st)) = 1 - \frac{Pr(M^{-1}M(st) \cap \aleph)}{Pr(M^{-1}M(st) \cap K)} = \frac{Pr(M^{-1}M(st) \cap (K - \aleph))}{Pr(M^{-1}M(st) \cap K)}, \tag{C.3}$$

which is the probability of ambiguity of st as in (2.3) when the pair of languages (L, K) is replaced with $(K, K - \aleph)$. Thus we can replace $1 - P_N^*(M(st))$ in (C.2) with the right hand side of (C.3), and in which case (C.2) becomes equivalent to S-Diagnosability of $(K, K - \aleph)$ as in (2.2).

Next we show that the pair $(K, K - \aleph)$ is indeed S-Diagnosable. Assume for contradiction that $(K, K - \aleph)$ is not S-Diagnosable. Then there exists $s \in \aleph$ and $s' \in K - \aleph$ satisfying the condition of Theorem 2. Then we have $\forall n \in \mathbb{N}$, $Pr(t : t \in [K - \aleph] \backslash s' \cap \Sigma^n) = \sum_{o \in \Delta^*} Pr(t : t \in [K - \aleph] \backslash s' \cap \Sigma^n, M(t) = o) = \sum_{o \in \Delta^*} Pr(t : t \in K \backslash s \cap \Sigma^n, M(t) = o) = Pr(t : t \in K \backslash s \cap \Sigma^n) = 1$, where the second equality follows from Theorem 2 and the last equality follows from the fact that $s \in \aleph$ (so all its extensions are in K). Thus, $\forall n \in \mathbb{N}, Pr(t : t \in [K - \aleph] \backslash s' \cap \Sigma^n) = 1$, i.e., $\forall n \in \mathbb{N}, Pr(\{s'\}\Sigma^n \cap (K - \aleph)) = 1$, implying that $\forall n \in \mathbb{N}, Pr(\{s'\}\Sigma^n \cap K) = 1$ (since $K - \aleph \subseteq K$), which further implies that $s' \in \aleph$. This contradicts the fact that $s' \in K - \aleph$. So the S-Diagnosability of $(K, K - \aleph)$ follows, which proves (C.2) and equivalently (C.1). □

ACKNOWLEDGEMENTS

This dissertation was supported in part by the National Science Foundation under the grants, NSF-ECCS-0801763, NSF-ECCS-0926029, and NSF-CCF-1331390.

BIBLIOGRAPHY

[1] W. Qiu, Q. Wen, and R. Kumar, “Decentralized diagnosis of event-driven systems for safely reacting to failures,” *IEEE Trans. Auto. Sci. Eng.*, vol. 6, no. 2, pp. 362–366, Apr. 2009.

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

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

[4] T. Yoo and H. Garcia, “Stochastic event counter for discrete-event systems under unreliable observations,” in *Proc. 2008 Amer. Control Conf.*, Seattle, WA, Jun. 2008, pp. 1145–1152.

[5] C. Zhou, R. Kumar, and S. Jiang, “Keynote: Hierarchical fault detection in embedded control software,” in *Proc. 2008 IEEE Int. Computer Software and Applications Conf.*, Jul. 2008, pp. 816–823.

[6] R. Isermann, R. Schwarz, and S. Stolzl, “Fault-tolerant drive-by-wire systems,” *IEEE Control Syst. Mag.*, vol. 27, no. 5, pp. 64–81, Oct. 2002.

[7] M. He and J. Zhang, “A dependency graph approach for fault detection and localization towards secure smart grid,” *IEEE Trans. Smart Grid*, vol. 2, no. 2, pp. 342–351, Jun. 2011.

[8] K. Kim and E. B. Bartlett, "Nuclear power plant fault diagnosis using neural networks with error estimation by series association," *IEEE Trans. Nucl. Sci.*, vol. 43, no. 4, pp. 2373–2388, Aug. 1996.

[9] C. Favre, "Fly-by-wire for commercial aircraft: the airbus experience," *Int. J. Control*, vol. 59, no. 1, pp. 139–157, 1994.

[10] G. Westerman, R. Kumar, C. Stroud, and J. Heath, "Discrete event system approach for delay fault analysis in digital circuits," in *Proc. 1998 Amer. Control Conf.*, Philadelphia, PA, Jun. 1998, pp. 239–243.

[11] I. Hwang, S. Kim, Y. Kim, and C. E. Seah, "A survey of fault detection, isolation, and reconfiguration methods," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 3, pp. 636–653, May 2010.

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

[13] X. Zhang, M. M. Polycarpou, and T. Parisini, "A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems," *IEEE Trans. Autom. Control*, vol. 47, no. 4, pp. 576–593, Apr. 2002.

[14] U. Lerner, R. Parr, D. Koller, and G. Biswas, "Bayesian fault detection and diagnosis in dynamic systems," in *Proc. of the National Conference on Artificial Intelligence (AAAI-00)*, Austin, Texas, Aug. 2000, pp. 531–537.

[15] F. Zhao, X. Koutsoukos, H. Haussecker, J. Reich, and P. Cheung, "Monitoring and fault diagnosis of hybrid systems," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 35, no. 6, pp. 1225–1240, Dec. 2005.

[16] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete-event systems," *IEEE Trans. Autom. Control*, vol. 40, no. 9, pp. 1555–1575, Sep. 1995.

[17] S. Jiang, Z. Huang, V. Chandra, and R. Kumar, “A polynomial algorithm for testing diagnosability of discrete-event systems,” *IEEE Trans. Autom. Control*, vol. 46, no. 8, pp. 1318–1321, Aug. 2001.

[18] D. Thorsley, T.-S. Yoo, and H. E. Garcia, “Diagnosability of stochastic discrete-event systems under unreliable observations,” in *Proc. 2008 Amer. Control Conf.*, Seattle, WA, Jun. 2008, pp. 1158–1165.

[19] Y. Wang, T.-S. Yoo, and S. Lafortune, “Decentralized diagnosis of discrete event systems using unconditional and conditional decisions,” in *Proc. 44th IEEE Conf. Decision Control/Eur. Control Conf.*, Seville, Spain, Dec. 2005, pp. 6298–6304.

[20] T.-S. Yoo and S. Lafortune, “Polynomial-time verification of diagnosability of partially observed discrete-event systems,” *IEEE Trans. Autom. Control*, vol. 47, no. 9, pp. 1491–1495, Sep. 2002.

[21] W. Qiu and R. Kumar, “Decentralized failure diagnosis of discrete event systems,” *IEEE Trans. Syst., Man, Cybern. A, Syst., Human*, vol. 36, no. 2, pp. 384–395, Mar. 2006.

[22] E. Athanasopoulou, L. Li, and C. N. Hadjicostis, “Probabilistic failure diagnosis in finite state machines under unreliable observations,” in *Proc. 8th Int. Workshop on Discrete Event Syst.*, Ann Arbor, MI, Jul. 2006, pp. 301–306.

[23] W. Qiu and R. Kumar, “Distributed diagnosis under bounded-delay communication of immediately forwarded local observations,” *IEEE Trans. Syst., Man, Cybern. A, Syst., Human*, vol. 38, no. 3, pp. 628–643, May 2008.

[24] W.-C. Lin, H. E. Garcia, and T.-S. Yoo, “A diagnoser algorithm for anomaly detection in DEDS under partial and unreliable observations: characterization and inclusion in sensor configuration optimization,” *Discrete Event Dyn. Syst.*, vol. 23, no. 1, pp. 61–91, Mar. 2013.

[25] W. Lin, H. Garcia, and T. Yoo, “Selecting observation platforms for optimized anomaly detectability under unreliable partial observations,” in *Proc. 2011 Amer. Control Conf.*, San Francisco, CA, Jun. 2011, pp. 4470–4477.

[26] S. Hashtrudi Zad, R. Kwong, and W. Wonham, “Fault diagnosis in timed discrete-event systems,” in *Proc. 38th IEEE Conf. Decision Control*, Phoenix, Arizona, Dec. 1999, pp. 1756–1761.

[27] D. Pandalai and L. Holloway, “Template languages for fault monitoring of timed discrete-event systems,” *IEEE Trans. Autom. Control*, vol. 45, no. 5, pp. 868–882, May 2000.

[28] J. Lunze, “Fault diagnosis of discretely controlled continuous systems by means of discrete-event models,” *Discrete Event Dyn. Syst.*, vol. 18, no. 2, pp. 181–210, 2008.

[29] S. Bhattacharyya, Z. Huang, V. Chandra, and R. Kumar, “Discrete event systems approach to network fault management: Detection & diagnosis of faults,” in *Proc. 2004 Amer. Control Conf.*, Boston, MA, Jun 30 - Jul. 2 2004, pp. 5108–5113.

[30] R. Kumar and S. Takai, “A framework for control-reconfiguration following fault-detection in discrete event systems,” in *Proc. 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SafeProcess)*, Mexico City, Mexico, Aug. 2012, pp. 848–853.

[31] S. Takai and R. Kumar, “Decentralized diagnosis for nonfailures of discrete event systems using inference-based ambiguity management,” *IEEE Trans. Syst., Man, Cybern. A, Syst., Human*, vol. 40, no. 2, pp. 406–412, Mar. 2010.

[32] R. Kumar and S. Takai, “Inference-based ambiguity management in decentralized decision-making: Decentralized diagnosis of discrete-event systems,” *IEEE Trans. Auto. Sci. Eng.*, vol. 6, no. 3, pp. 479–491, Jul. 2009.

[33] F. Liu, D. Qiu, H. Xing, and Z. Fan, “Decentralized diagnosis of stochastic discrete event systems,” *IEEE Trans. Autom. Control*, vol. 53, no. 2, pp. 535–546, Mar. 2008.

[34] Y. Wang, T. Yoo, and S. Lafortune, “Diagnosis of discrete event systems using decentralized architectures,” *Discrete Event Dyn. Syst.*, vol. 17, no. 2, pp. 233–263, Jan. 2007.

[35] R. M. G. Ferrari, T. Parisini, and M. M. Polycarpou, “Distributed fault detection and isolation of large-scale discrete-time nonlinear systems: An adaptive approximation approach,” *IEEE Trans. Autom. Control*, vol. 57, no. 2, pp. 275–290, Feb. 2012.

[36] J. Chen and R. Kumar, “Decentralized failure diagnosis of stochastic discrete event systems,” in *Proc. 9th IEEE Int. Conf. Autom. Sci. and Eng.*, Madison, WI, Aug. 2013, pp. 1083–1088.

[37] R. Debouk, S. Lafortune, and D. Teneketzis, “On the effect of communication delays in failure diagnosis of decentralized discrete event systems,” *Discrete Event Dyn. Syst.*, vol. 13, no. 3, pp. 263–289, 2003.

[38] ——, “Coordinated decentralized protocols for failure diagnosis of discrete event systems,” *Discrete Event Dyn. Syst.*, vol. 10, pp. 33–79, 2000.

[39] A. Aghasaryan, E. Fabre, A. Benveniste, R. Boubour, and C. Jard, “Fault detection and diagnosis in distributed systems: an approach by partially stochastic petri nets,” *Discrete Event Dyn. Syst.*, vol. 8, no. 2, pp. 203–231, 1998.

[40] J. Chen and R. Kumar, “Polynomial test for stochastic diagnosability of discrete event systems,” in *Proc. 8th IEEE Int. Conf. Autom. Sci. and Eng.*, Seoul, Korea, Aug. 2012, pp. 521–526.

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

[42] R. H. Chen, D. L. Mingori, and J. L. Speyer, “Optimal stochastic fault detection filter,” *Automatica*, vol. 39, no. 3, pp. 377–390, Mar. 2003.

[43] J. Chen and R. Kumar, “Online failure diagnosis of stochastic discrete event systems,” in *Proc. 2013 IEEE Multi-Conf. Syst. and Control*, Hyderabad, India, Aug. 2013, pp. 194–199.

[44] D. Lefebvre and E. Leclercq, “Stochastic petri net identification for the fault detection and isolation of discrete event systems,” *IEEE Trans. Syst., Man, Cybern. A, Syst., Human*, vol. 41, no. 2, pp. 213–225, Mar. 2011.

[45] Y. Zhang, X. R. Li, and K. Zhou, “A fault detection and diagnosis approach based on hidded Markov chain model,” in *Proc. 1998 Amer. Control Conf.*, Philadelphia, PA, Jun. 1998, pp. 2012–2016.

[46] J. Lunze and J. Schroder, “Sensor and actuator fault diagnosis of systems with discrete inputs and outputs,” *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 34, no. 2, pp. 1096–1107, Apr. 2004.

[47] J. Chen and R. Kumar, “Polynomial test for stochastic diagnosability of discrete event systems,” *IEEE Trans. Auto. Sci. and Eng.*, vol. 10, no. 4, pp. 969–979, Oct. 2013.

[48] S. Jiang and R. Kumar, “Failure diagnosis of discrete-event systems with linear-time temporal logic fault specifications,” in *Proc. 2002 Amer. Control Conf.*, Anchorage, AK, May 2002, pp. 128–133.

[49] ——, “Failure diagnosis of discrete-event systems with linear-time temporal logic specifications,” *IEEE Trans. Autom. Control*, vol. 49, no. 6, pp. 934–945, Jun. 2004.

[50] J. Chen and R. Kumar, “Failure diagnosis of discrete-time stochastic systems subject to temporal logic correctness requirements,” in *Proc. 2014 IEEE Int. Conf. Netw. Sensing, and Control*, Miami, FL, Apr. 2014, pp. 42–47.

[51] R. Kumar and V. K. Garg, *Modeling and Control of Logical Discrete-Event Systems*. Boston, MA: Kluwer Academic Publishers, 1995.

[52] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed. New York: Springer Science+Business Media, 2008.

[53] M. O. Rabin, “Probabilistic automata,” *Inform. and Contr.*, vol. 6, pp. 230–245, 1963.

[54] V. K. Garg, R. Kumar, and S. I. Marcus, “A probabilistic language formalism for stochastic discrete-event systems,” *IEEE Trans. Autom. Control*, vol. 44, no. 2, pp. 280–293, Feb. 1999.

[55] P. Brémaud, *Markov Chains: Gibbs Fields, Monte Carlo Simulation and Queues.* New York: Springer-Verlag, 1999.

[56] S. Jiang, R. Kumar, and H. Garcia, “Optimal sensor selection for discrete-event systems with partial observation,” *IEEE Trans. Autom. Control*, vol. 48, no. 3, pp. 369–381, Mar. 2003.

[57] R. Kumar and V. K. Garg, “Control of stochastic discrete event systems modeled by probabilistic languages,” *IEEE Trans. Autom. Control*, vol. 46, no. 4, pp. 593–606, Apr. 2001.

[58] A. Arapostathis, R. Kumar, and S. Tangirala, “Controlled Markov chains with safety upper bound,” *IEEE Trans. Autom. Control*, vol. 48, no. 7, pp. 1230–1234, Jul. 2003.

[59] A. Arapostathis, R. Kumar, and S.-P. Hsu, “Control of Markov chains with safety bounds,” *IEEE Trans. Auto. Sci. Eng.*, vol. 2, no. 4, pp. 333–343, Oct. 2005.

[60] V. Pantelic, S. Postma, and M. Lawford, “Probabilistic supervisory control of probabilistic discrete event systems,” *IEEE Trans. Autom. Control*, vol. 54, no. 8, pp. 2013–2018, Aug. 2009.

[61] V. Pantelic and M. Lawford, “Optimal supervisory control of probabilistic discrete event systems,” *IEEE Trans. Autom. Control*, vol. 57, no. 5, pp. 1110–1124, May 2012.

[62] A. Abate, S. Amin, M. Prandini, J. Lygeros, and S. Sastry, “Computational approaches to reachability analysis of stochastic hybrid systems,” *Hybrid Systems: Computation and Control*, vol. 4416 of LNCS, pp. 4–17, 2007.

[63] S. Summers and J. Lygeros, “Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem,” *Automatica*, vol. 46, no. 12, pp. 1951–1961, Sep. 2010.

[64] A. Abate, A. D. Innocenzo, and M. D. D. Benedetto, “Approximate abstractions of stochastic hybrid systems,” *IEEE Trans. Autom. Control*, vol. 56, no. 11, pp. 2688–2694, Nov. 2011.

[65] A. A. Julius and G. J. Pappas, “Approximations of stochastic hybrid systems,” *IEEE Trans. Autom. Control*, vol. 54, no. 6, pp. 1193–1203, Jun. 2009.

[66] A. Abate, M. Prandini, J. Lygeros, and S. Sastry, “Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems,” *Automatica*, vol. 44, no. 11, pp. 2724–2734, Oct. 2008.

[67] X. D. Koutsoukos and D. Riley, “Computational methods for verification of stochastic hybrid systems,” *IEEE Trans. Syst., Man, Cybern. A, Syst., Human*, vol. 38, no. 2, pp. 385–396, Mar. 2008.

[68] A. A. Julius and G. J. Pappas, “Probabilistic testing for stochastic hybrid systems,” in *Proc. 2008 IEEE Conf. Decision Control*, Cancun, Mexico, Dec. 2008, pp. 4030–4035.

[69] A. A. Julius, A. Girard, and G. J. Pappas, “Approximate bisimulation for a class of stochastic hybrid systems,” in *Proc. 2006 Amer. Control Conf.*, Minneapolis, MN, Jun. 2006, pp. 4724–4729.

[70] A. Abate, J.-P. Katoen, J. Lygeros, and M. Prandini, “Approximate model checking of stochastic hybrid systems,” *Euro. J. Control*, vol. 16, no. 6, pp. 624–641, 2010.

[71] G. E. Fainekos, S. G. Loizou, and G. J. Pappas, “Translating temporal logic to controller specifications,” in *Proc. 45th IEEE Conf. Decision Control*, San Diego, CA, USA, Dec. 2006, pp. 899–904.

[72] M. Li and R. Kumar, “Reduction of automated test generation for simulink/stateow to reachability and its novel resolution,” in *Proc. 9th IEEE Int. Conf. Autom. Sci. and Eng.*, Madison, WI, Aug. 2013, pp. 1089–1094.

[73] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess, and B. Wu, *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. Hoboken, NJ: Wiley, 2006.

[74] S. Genc and S. Lafortune, “Predictability of event occurrences in partially-observed discrete-event systems,” *Automatica*, vol. 45, no. 2, pp. 301–311, 2009.

[75] R. Kumar and S. Takai, “Decentralized prognosis of failures in discrete event systems,” *IEEE Trans. Autom. Control*, vol. 55, no. 1, pp. 48–59, Jan. 2010.

[76] F. Cassez and A. Grastien, “Predictability of event occurrences in timed systems,” in *Proc. 11th Intl. Conf. Formal Modeling and Analysis of Timed Systems*, Buenos Aires, Argentina, Aug. 2013.

[77] J. Chen and R. Kumar, “Failure prognosability of stochastic discrete event systems,” in *Proc. 2014 Amer. Control Conf.*, Portland, OR, Jun. 2014, pp. 2041–2046.

[78] ——, “Pattern mining for predicting critical events from sequential event data log,” in *Proc. 2014 Int. Workshop on Discrete Event Syst.*, Paris-Cachan, France, May 2014, pp. 1–6.

[79] S. Takai and R. Kumar, “Inference-based decentralized prognosis in discrete event systems,” *IEEE Trans. Autom. Control*, vol. 56, no. 1, pp. 165–171, Jan. 2011.

[80] ——, “Distributed failure prognosis of discrete event systems with bounded-delay communications,” *IEEE Trans. Autom. Control*, vol. 57, no. 5, pp. 1259–1265, May 2012.

[81] C. Baier and J.-P. Katoen, *Principles of model checking.* MIT Press, 2008.

[82] E. A. Emerson, “Temporal and modal logic.” *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, vol. 995, p. 1072, 1990.

[83] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking.* Cambridge, MA:MIT Press, 1999.

[84] W.-G. Tzeng, “A polynomial-time algorithm for the equivalence of probabilistic automata,” *SIAM J. Comput.*, vol. 21, no. 2, pp. 216–227, Apr. 1992.

[85] X. Wang and A. Ray, “A language measure for performance evaluation of discrete-event supervisory control systems,” *Applied Math. Modelling*, vol. 28, no. 9, pp. 817–833, Sep. 2004.

[86] J. Lunze and J. Schröder, “State observation and diagnosis of discrete-event systems described by stochastic automata,” *Discrete Event Dyn. Syst.*, vol. 11, no. 4, pp. 319–369, 2001.

[87] J. Lunze, “Qualitative modelling of linear dynamical systems with quantised state measurements,” *Automatica*, vol. 30, no. 3, pp. 417–431, 1994.

[88] ——, “On the Markov property of quantised state measuirement sequences,” *Automatica*, vol. 34, no. 11, pp. 1439–1444, 1998.

[89] F. Mueller, “Challenges for cyber-physical systems: Security, timing analysis and soft error protection,” in *National Workshop on High Confidence Software Platforms for Cyber-Physical Systems: Research Needs and Roadmap (HCSP-CPS)*, Nov. 2006.

[90] A. P. Sistla, M. Zefran, and Y. Feng, “Runtime monitoring of stochastic cyber-physical systems with hybrid state,” *Lecture Notes in Computer Sci.*, vol. 7186, pp. 276–293, 2012.

[91] M. Zhong, S. X. Ding, J. Lam, and H. Wang, “An LMI approach to design robust fault detection filter for uncertain lti systems,” *Automatica*, vol. 39, no. 3, pp. 543–550, 2003.

[92] W. H. Chung and J. L. Speyer, “A game theoretic fault detection filter,” *IEEE Trans. Autom. Control*, vol. 43, no. 2, pp. 143–161, Feb. 1998.

[93] M. E. Basseville and I. V. Nikiforov, “Detection of abrupt changes: theory and application,” 1993.

[94] M. Basseville, “Detecting changes in signals and systems—a survey,” *Automatica*, vol. 24, no. 3, pp. 309–326, 1988.

[95] A. Fehnker and F. Ivančić, “Benchmarks for hybrid systems verification,” *Hybrid Systems: Computation and Control*, vol. 2293 of LNCS, pp. 326–341, 2004.

[96] A. V. Goldberg, “Scaling algorithms for the shortest paths problem,” *SIAM J. Comput.*, vol. 24, no. 3, pp. 494–504, Jun. 1995.

[97] A. Xie and P. A. Beerel, “Efficient state classification of finite-state Markov chains,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 17, no. 12, pp. 1334–1339, Dec. 1998.

[98] M. K. Reiter and A. D. Rubin, “Crowds: Anonymity for web transactions,” *ACM Trans. Inf. and Syst. Security*, vol. 1, no. 1, pp. 66–92, Nov. 1998.

[99] A. Saboori and C. N. Hadjicostis, “Probabilistic current-state opacity is undecidable,” in *Proc. of the 19th Intl. Symposium on Mathematical Theory of Networks and Systems*, Budapest, Hungary, Jul. 2010.

[100] B. Bérard, J. Mullins, and M. Sassolas, “Quantifying opacity,” in *Proc. of 2010 Intl. Conf. Quantitative Evaluation of Systems.* Williamsburg, VA: IEEE, Sep. 2010, pp. 263–272.

[101] V. Shmatikov, “Probabilistic analysis of anonymity,” in *Prof. of 15th IEEE Computer Security Foundations Workshop*, 2002, pp. 119–128.

[102] A. Legay, B. Delahaye, and S. Bensalem, “Statistical model checking: An overview,” in *Runtime Verification.* Springer, 2010, pp. 122–135.

[103] M. Kwiatkowska, G. Norman, and D. Parker, “Advances and challenges of probabilistic model checking,” in *48th Annual Allerton Conf. Communication, Control, and Computing*, Allerton House, UIUC, Illinois, pp. 1691–1698.

[104] H. Hansson and B. Jonsson, “A logic for reasoning about time and reliability,” *Formal aspects of computing*, vol. 6, no. 5, pp. 512–535, 1994.

[105] R. A. Hanneman and M. Riddle, *Introduction to social network methods.* Riverside, CA: University of California, Riverside, 2005, (published in digital form at http://faculty.ucr.edu/∼hanneman/).

[106] R. P. Kindermann and J. L. Snell, “On the relation between Markov random fields and social networks*,” *Journal of Mathematical Sociology*, vol. 7, no. 1, pp. 1–13, 1980.

[107] D. Shah and T. Zaman, “Rumors in a network: Whos the culprit?” *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5163–5181, Aug. 2011.

[108] S. Takai and R. Kumar, “Verification and synthesis for secrecy in discrete-event systems,” in *Proc. 2009 Amer. Control Conf.*, St. Louis, MO, Jun. 2009, pp. 4741–4746.

[109] M. Ibrahim, J. Chen, and R. Kumar, “Secrecy in stochastic discrete event systems,” in *Proc. 2014 IEEE Int. Conf. Netw. Sensing, and Control*, Miami, FL, Apr. 2014, pp. 48–53.

[110] R. Akella, H. Tang, and B. M. McMillin, “Analysis of information flow security in cyber–physical systems,” *Int. J. Critical Infrastructure Protection*, vol. 3, no. 3, pp. 157–173, 2010.