MULTI-RESOLUTION FAULT DIAGNOSIS IN DISCRETE-EVENT SYSTEMS

JiangJing Pan

A Dissertation

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Doctor of philosophy (Electrical Engineering) at

Concordia University

April 2008

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-37757-4
Our file Notre référence
ISBN: 978-0-494-37757-4

ABSTRACT

Multi-Resolution Fault Diagnosis in Discrete-Event Systems .

JiangJing Pan
Concordia University

In this thesis, a framework for multi-resolution fault diagnosis in discrete-event systems (DES) is introduced. Here a sequence of plant models, with increasing resolution, are used in fault diagnosis and the range of possible diagnosis is narrowed down step by step, until the failure mode is isolated. In this way, the original problem of fault diagnosis is replaced by a sequence of smaller problems. The plant models used at each step of diagnosis are abstractions of the original plant model. We propose to use model reduction through the solutions of the Relational Coarsest Partition problem to obtain these abstractions. For each diagnosis step, minimal sensor sets are chosen to have a coarser output map, and hence, to improve the efficiency of model reduction.

In this thesis, a polynomial algorithm is proposed that verifies failure diagnosability by examining the distinguishability of two plant (normal/faulty) conditions at a time. A procedure is presented that finds minimal sensor sets, referred to as minimal distinguishers, for distinguishability of one condition from another. A polynomial procedure is introduced that combines minimal distinguishers to obtain a minimal sensor set for fault diagnosis. The proposed method reduces the computational complexity of sensor selection.

A benefit of using minimal distinguishers is that their computation maybe speeded up using expert knowledge. The proposed method for sensor selection is particularly suitable for multi-resolution diagnosis since it permits some of the results of computations, performed for sensor selection at the lowest (finest) level of multi-resolution diagnosis to be reduced at higher levels. This feature is particularly useful in reducing the computations necessary for online reconfiguration of the multi-resolution diagnosis system.

An important procedure used in sensor selection is testing diagnosability. In this thesis, a new procedure for testing diagnosability in timed DES is introduced based on the relatively timing of plant output sequence. It is shown through example that the proposed test maybe executed with significantly fewer computations compared to tests developed for untimed models and adapted for timed systems. Furthermore, two new sets of sufficient conditions are provided under which diagnoser design and diagnosability tests based on relative timing of output sequence can be performed efficiently.

ACKNOWLEDGEMENTS

I would like to deeply express my thanks to my supervisor, Dr. S. Hashtrudi-Zad, whose version in discrete-event theory and failure diagnosis provided the initial impetus for the present work, and his inspiration and guidance have inspired developments at many of the conclusions in this dissertation. Also I am much indebted for his endless patience, careful reading and correction of the manuscript.

Special thanks to my previous supervisor, Dr. K. Khorasani, who shares some of his experience and offers valuable suggestions in the years of my study.

To finish my thesis I also depended on the support of my family who were not subjects of my research but nevertheless helped me significantly by not giving me any financial pressure and spiritual pressure.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

Figure Page

LIST OF TABLES

Chapter 1

INTRODUCTION

Fault diagnosis in engineering systems is of great practical significance in protecting life and property, and in increasing reliability and productivity [1]. The systems concerned encompass a broad spectrum of human-made machineries and management systems, including industrial production facilities (water treatment plant, power plant, etc.), transportation systems (aerospace, automobile, traffic control, etc.) and household appliances (washers, air conditioners, etc.). Solving diagnostic problems for these engineering systems is a complicated task and requires a systematic approach.

The "activity" in many rapidly evolving modern systems is governed by operational rules designed by humans; therefore, their dynamics are characterized by asynchronous occurrences of discrete events. These features lend themselves to the term Discrete-Event System (DES) for this class of dynamic systems [2, 24]. As a result, solving diagnostic problems for discrete-event systems has been the subject of extensive research (e.g., [3–6, 18, 19]).

One of the problems associated with the use of discrete-event models is the computational complexity (i.e., combinational explosion). This has motivated researches to find solutions to mitigate the computational problem. In this thesis, we develop a procedure for a multi-resolution fault diagnosis which attempts to replace the original diagnosis problem with a sequence of simpler problems.

1.1 Literature Review

A **failure**, or a **fault**[1], refers to a non-permitted deviation in the behavior of a system from that required by the system specifications [11]. Faults can be either **permanent** or **nonpermanent**. Permanent faults are those in which once the component fails, the system remains in the faulty condition indefinitely. Nonpermanent faults are faults of limited duration, caused by temporary malfunction of the system or due to some external interference. Given the possibilities of faults, three direct approaches are used to improve the reliability of a system: fault prevention (avoidance), fault tolerance, and fault diagnosis [7].

Fault prevention is to try to prevent faults from occurring or getting introduced into the system. In the traditional approach of fault prevention, high reliability is achieved by eliminating as many faults as possible before the system is put in regular use. Since all possible faults cannot be anticipated and eliminated before deployment of the system, fault avoidance assumes that system failures will occasionally take place. The goal of fault tolerance is to provide service despite the presence of faults in the system. Fault tolerant systems employ redundancy to mask various types of failures. That is, the system contains components that are not needed in the regular use but used to avert disruption in system behavior in case some components fail.

[1] In this thesis, "failure" and "fault" have been used interchangeably [7].

Fault diagnosis is to try to detect and isolate the fault once it occurs and before it causes a catastrophe in the system. The goal of fault diagnosis is to protect the life and property, and to increase operational time and productivity.

These three approaches are consistent in nature. The fault prevention methods focus on methodologies for design, testing and validation; fault tolerance methods focus on how to use components in a manner such that failures can be masked; whereas, fault diagnosis methods often focus on how to detect and isolate the failures both in design and implementation philosophy. In this work, we concentrate on fault diagnosis.

1.1.1 Fault Diagnosis

The conventional techniques for fault diagnosis are classified to model-free and model-based methods. Expert system and hardware redundancy are two commonly used techniques in model-free methods. Expert systems, which employ fault tree (also called belief network to represent relationships between the faults and their causes) or an inference engine to detect failures, are experience-based or knowledge-based. An expert system is generated by gathering the expertise; this is usually difficult and time-consuming (see, e.g., [8, 9]). Therefore, for new systems a considerable amount of time might elapse before enough knowledge has been accumulated to make reliable diagnosis possible. Furthermore, the rule bases developed for the expert system are very domain dependent and not reusable, and the system performance is strictly limited by the quality and quantity of the implemented knowledge. However, in cases where models are not easy to develop expert systems can be very effective in offline diagnosis (when the system is not operational).

Hardware redundancy is often implemented as triple modular redundancy (TMR). For the purpose of fault diagnosis, triplicate sensors are used to measure the same system variable. This method is not only expensive but also shows a conspicuous difficulty in detecting actuators and plant faults.

In addition to model-free methods, several model-based techniques for fault diagnosis have been proposed in the literature. In a model-based method, the observed behavior of systems is compared with that expected from the system model. Based on this comparison, the condition of the system (normal or faulty) is inferred.

Many approaches to model-based fault diagnosis are based on parameter estimations and state estimations for continuous variable systems (e.g., [1,10–15]). In these methods, the systems are modelled using differential and difference equations (see, e.g., [16,17]). On the other hand, the rapid evolution of computing, communication, and sensor technologies has brought about the proliferation of the modern dynamic systems, mostly technological and often highly complex. In order to reason these systems through at some level of abstraction, it is more efficient, and regularly more reliable to model some of its aspects as discrete. A Discrete Event System (DES) is then introduced to be a dynamic system with discrete inputs and outputs, whose behavior can be described in terms of discrete state transitions. Based on these abstractions, researchers seek to find out reasoning mechanisms for performing various tasks. The fault diagnosis using DES models, as one of them, has arisen for systematic approach to component failure detection and isolation([3–5,18,19]).

The investigation on the topic of diagnosis of Discrete Event Systems was prompted by some preliminary works within the automatic control (see, e.g., [3,4,20]) and the other areas such as artificial intelligence (AI) (e.g., [21]). In modelling, F. Lin proposed a discrete event system approach for fault diagnosis in [3]. He assumes that each system component has some normal states and some faulty states, and uses system outputs for diagnosis. The main goal of diagnosis is to find the block of the normal / faulty partition that the system state belongs to by issuing a sequence of control commands and observing the outputs. The system is online diagnosable if there exists a control sequence that diagnoses the system. The method is called a *state-based approach*, and is further treated by S. Bavishi *et al.* [20]. Because in this methodology, sequence of control commands are issued as inputs to the system, the method is an *active* diagnosis approach. It should be noted that in practice, the set of testing commands that can be used is constrained by the operational requirements of the system.

In contrast to the abovementioned state-based approach, M. Sampath et al [4] proposed an *event-based approach* to failure diagnosis problems. In this approach, inference is made about the occurrence of unobservable failure events based on observed events. Here, the *diagnoser*, which provides estimates of the state of the system after the occurrence of every observable event, can be viewed as an extended sensor to detect the unobservable faults in DES. This approach does not apply any control command as the input to the system, and therefore, is *passive*. Later in [22], Sampath *et al.* generalize their notion of diagnosis to active DES fault diagnosis and present an integrated approach to fault diagnosis and supervisory control.

In [23], a *state-based* but *passive* fault diagnosis is introduced where the objective is to monitor system outputs and determine the condition (normal/faulty) of the system. In this work, an equivalence relation on the system state set based on the solution of the *Relational Coarsest Partition* (RCP) problem is used to reduce the system model and therefore, the diagnosis system.

All of the above modelling approaches use automata as basic building blocks to model system components. The complete system model can be generated using synchronous or parallel composition of component models (see e.g., [24,25],). On the other hand, A. Darwiche and A. Misra *et al.* propose a structured system description of DES models ([26,27]), called causal network, for the purpose of fault diagnosis. This qualitative model seems to be suitable for process diagnosis in local power station plants [27]. Compared to DES modelling techniques in [3,20] and [4], this approach exhibits surprising computational efficiency for diagnosing discrete event systems. However, it also employs logical sentences describing the status of each component, which is partly from human experience and expertise [27].

In [65], modelling and diagnosis of discrete event systems has been examined under the strong assumption that state transitions of systems are observable. Algorithms for partitioning faults into sets of indistinguishable faults and for determining the detectability of faults are also derived in [65]. The detectability test (Definition 5.2 in [65]) relies on not only the states but also the inputs. Thus, the problem addressed is an *active* diagnosis problem.

In cases where timing constraints can be used to improve diagnosis accuracy, the timed models are considered in fault diagnosis ([23,28–30]). [28] presents models in the form of temporal causal networks, which are capable of capturing both the ordering of events as well as the relative time between events, and modelling the persistence of faults. However, it did not consider the complete timing information such as time bounds and delays.

In [29], for failure diagnosis, the timed event sequence generated by the DES is compared with a set of specifications for normal operations, called templates. This technique of sequencing and timing relationships is demonstrated in manufacturing lines where the system can be modelled as a set of finite timed DES models based on an unspecified number of instances of the process. However, the construction of templates is probably not feasible and failure isolation and diagnosability have not been discussed in this work.

[30] discusses the direct application of the event-based approach of [4] to timed discrete-event systems by simply considering the clock tick as an observable event. [23] notes that following [30], an update of state estimate of the system is required after every clock tick which could make the diagnosis computations costly. [23] proposes an alternative approach in which state and condition estimates are generated after every new output based on the new output and the number of clock ticks since the last output. In this way, state and condition updates after every clock tick can be avoided. In this approach, the information about the timing of output changes is computed (using the system model) and stored in a Timed Reachability Transition System (timed RTS). This technique can be useful if the required transition-times can be computed easily. [23] mentions timed DES in which transition-times are bounded as an example of such cases.

As the development of these fundamental works in the area of DES model-based diagnosis, the topic has generated tremendous interests in various areas of applications such as communication (see, e.g., [34–36]) and distributed systems (see, e.g., [37–39]) and hybrid systems (e.g., [40–42]).

1.1.2 Testing Diagnosability

An important issue is the issue of failure diagnosability. Sampath *et al.* [4] consider a failure diagnosable if it can be detected and isolated in a bounded number of events. [4] also presents a test for diagnosability based on the construction of a diagnoser. The number of a diagnoser states and hence the complexity of the above-mentioned test for diagnosability is in the worst case exponential in the cardinality of the system state set.

[5] adopts an approach similar to [4] for diagnosability except that [5] also considers the cases in which a faulty condition may be present when diagnosis starts. Therefore, in [5], a failure is considered diagnosable if it can be detected and isolated after its occurrence or the start of diagnosis in a bounded number of events. The diagnosability conditions given in [5] are also in terms of the properties of a diagnoser and hence verifying them has a worst-case exponential complexity.

In contrast to the exponential approaches of testing diagnosability in [4, 5], efficient polynomial algorithms for testing diagnosability (in the framework of [4]) are proposed in [31,32]. The test in [31] has a complexity of $\mathcal{O}(|X|^4)$ and that in [32], has a complexity of $\mathcal{O}(|X|^2)$ (Here $|X|$ is the size of the system state set). The test in [32] is developed for deterministic systems and the test in [31] is applicable to nondeterministic systems as well. Common to these works is that their tests do not rely on diagnosers. Rather, the tests are based on the construction of finite-state automata (called "verifier" in [32]) that can be performed in polynomial time. Recently, A. Ramírez-Treviñn *et al.* [43] proposed another polynomial approach based on interpreted Petri nets (IPN). An algorithm for verification of diagnosability of IPN models is proposed which avoids reachability analysis of other approaches. Other attempts to manage the complexity of testing diagnosability in discrete-event systems include modular approaches (see, e.g., [47,49]).

1.1.3 Sensor Selection

The problem of fault diagnosis in monitored discrete-event system can be characterized by a set of observations to account for. Therefore, sensor selection and the observability of discrete-event system takes a basic role in the fault diagnosis of DES system. Specially, it is desired to have a minimal set of sensors that provides sufficient output information for guaranteeing the diagnosability of all failures.

Sensor selection problems in the context of discrete-event systems have been previously studied in [20,52–55,57–59]. In [53,55] and [59], the authors show that the problems of finding a sensor set with minimal cardinality for satisfying the properties of diagnosability, normality or observability are NP-complete. In [20], Bavishi et al. present an algorithm that determines a minimum-cost set of sensors which ensures testability. [54] provides a strategy for devising a test sequence to obtain a minimum-cost set of sensors ensuring a given property (such as diagnosability). In [52], the authors present algorithms with exponential complexity to find a set of observable events with minimal cardinality to ensure the properties of observability and normality in the context of supervisory control. [58] applies approximation algorithms to sensor set minimization for the supervisory control of DES.

Note that while the problem of finding a sensor set with minimal cardinality is NP-complete, the problem of finding a minimal sensor set for diagnosability (minimal in the sense that no subset of the minimal set satisfies diagnosability) is not NP-complete and in fact, can be computed in polynomial time (e.g., [53]).

1.1.4 Decentralized and Hierarchical Approaches to Fault Diagnosis

As previously mentioned, one of the main challenges of designing fault diagnosis systems based on discrete-event models is computational complexity. To reduce the computational complexity, researchers have turned their attention to using decentralized (see, e.g., [21,37–39,45,46]) and hierarchical approaches (see, e.g., [48,50,62]).

Typically, in decentralized approaches a class of discrete-event systems modelled as a network of communicating automata is considered, where each automaton describes the behavior of a system component. Approach like [45] consist of a set of local observations and local diagnosers linked with a communication network to a coordinator. Therefore, in [45], a decentralized diagnosis is presented in which part of diagnosis is done locally, and then the results are reported to a central diagnoser. This reduces the required communication between the local sites and central site; however, the global model of the system is still required.

Baroni *et al.* [21] study the diagnosis in systems consisting of components and links in which components and links are grouped into clusters. Clusters may be grouped into larger clusters themselves, resulting in a hierarchy of clusters. To reduce diagnosis computations, first local estimates are calculated, and then, these diagnoses are merged using a synchronous product to account for component interactions. For the purpose of communicating information from a local site to the global system or the coordinator, some logic rules or suitable consistency properties are required for online diagnosis. In [38], the proposed approach features a problem-decomposition/solution-composition nature whose core is the on-line progressive reconstruction of the behaviors of a class of distributed systems. However, the complexity of reconstruction is exponential in the number of messages in the cluster observation.

Other researches in decentralized approach includes [47]. Debouk *et al.* [47] study modular diagnosis in which local diagnoses are calculated but not merged to account for component interactions. The resulting diagnoses are, therefore, more conservative in the sense that they include normal or faulty modes that may not be present in a centralized diagnosis.

Hierarchical approach is another way to manage the complexity in large-scale systems. [50, 62, 67] present an approach to fault diagnosis for hierarchical finite-state machines. Taking advantage of system structure, at any point in time, only a submodel of the plant corresponding to the current phase operation, called D-holon, is used in diagnosis. This reduces computer memory requirement. Furthermore, when the common events of system components are observable, a semi-modular approach is presented for diagnosis that reduces the time complexity from exponential to polynomial.

Multi-resolution diagnosis can be regarded as a hierarchial approach, which examines the system at different layers of abstraction (from coarse to fine) to narrow down the diagnosis. In this way, the original diagnosis problem is replaced by a sequence of simpler problems. [60] introduces a hierarchical computational procedure for fault diagnosis. There a hierarchy consisting of several levels of abstraction for the plant is constructed at the design stage. Then, during the online operation of the diagnosis system, (decentralized) diagnosis is performed at successive levels, starting at the highest level of abstraction.

1.2 Thesis Objectives

The objective of this research is the development of a multi-resolution approach to fault diagnosis in discrete-event systems. This approach is used in order to replace the original fault diagnosis problem with a sequence of simpler problems, and hence smaller solutions. Towards this goal, models of the plant at various levels of abstraction (from coarse to fine) are developed. In this thesis we use the Relational Coarsest Partition (RCP) problem to reduce the plant model at various levels and thus construct the models needed for multi-resolution diagnosis. The reason we have chosen model reduction using RCP is that there exist efficient methods for the solution of RCP and model reduction using RCP could be quite effective. To improve the efficiency of model reduction, we aim to use minimal sensor sets. We develop sensor selection methods that are particularly suitable for sensor selection for multi-resolution diagnosis in the sense that the computations performed at the lowest (finest) level of hierarchy of models for sensor selection can be reused (entirely or at least, in part) for sensor selection at higher levels as well.

1.3 Thesis Outline and Contributions

In Chapter 2, we briefly review the background material on discrete event systems and model-based diagnosis, which we will use in the following chapters. The key notations and definitions in this chapter are presented along with simple examples.

In Chapter 3, we present our problem formulation with a simple physical example.

In this thesis, we develop a multi-resolution diagnosis system in a state-based framework [5] and a new method for sensor selection in diagnosis problems, and in particular, multi-resolution diagnosis systems. Sensor selection procedures involve diagnosability tests. In Chapter 4, we develop a test for failure diagnosability in the state-based framework of [5]. This test is polynomial and similar to those in [31, 32] for the event-based problem of [4].

While the development of multi-resolution and sensor selection algorithms are done in this thesis for untimed systems, the procedures are applicable to timed systems as well. Thus in Chapter 4 we also study testing diagnosability in timed DES. In this case, instead of adopting the tests for untimed models, following an approach similar to [23] for diagnoser design, we have developed a test based on the Timed Reachability Transition System (timed RTS) [23]. We show through examples, that if timed RTS is available (for instance, computed for diagnoser design following the procedure in [23]), then testing diagnosability based on timed RTS can be considerably less complex than tests based on untimed methods. [23] mentioned that timed RTS can be easily computed if the transition-time sets (i.e., the set of time in ticks between output changes) are bounded sets. In this thesis, we investigate the cases in which the transition-time sets may be unbounded. We obtain two sets of conditions under which the transition-time sets are unbounded but can be represented in the form of the union of a bounded number of intervals. We will see that in such case timed RTS can be conveniently computed. In this way, we expand the set of cases in which the method of diagnoser design of [23] and the corresponding diagnosability test in Chapter 4 of this thesis, are applicable.

In this thesis, we propose a multi-resolution fault diagnosis that uses reduced models of the system (various levels of abstraction) to perform the diagnosis. To make the model reduction more efficient we choose a minimal sensor set. In Chapter 5, we develop a sensor selection algorithm which is particularly suitable for multi-resolution diagnosis. Specifically, we propose a polynomial algorithm that verifies failure diagnosability by examining the distinguishability of two conditions at a time. Next a polynomial-time procedure is presented that first finds minimal sensor sets for distinguishing one condition from another. We refer to these minimal sets as *minimal distinguishers*. Then the procedure combines these minimal distinguishers to obtain a minimal sensor set for fault detection and isolation. It is shown that taking advantage of the structure of the system, as done in the algorithms proposed in this chapter, reduces the time complexity of testing diagnosability and sensor selection by a factor $\mathcal{O}(p^{2m-1})$ (compared with test for untimed DES in Chapter 4 and the test in [31] for nondeterministic untimed DES); here p is the number of failure modes and m ($m \ll p$) is maximum number of failures that can occur simultaneously. A benefit of using minimal distinguishers is that their computation (thus, the computations of sensor selection) may be speeded up using heuristics and expert knowledge. The proposed method for sensor selection is particularly suitable for multi-resolution diagnosis problem since some of the results of computations performed for sensor selection at the lowest (finest) level of multi-resolution diagnosis can be reused for sensor selection at higher levels.

In Chapter 6, we propose a multi-resolution fault diagnosis system in which fault diagnosis is performed at several steps. First, the occurrence of fault is detected. At the second step, once the occurrence of fault is detected, the fault group is isolated. Each fault group may, for instance, contain the faults in one of the subsystems. Therefore, at the second step the diagnosis is narrowed down. This process of narrowing down the diagnosis is repeated until the failure mode is isolated. In this way, the original problem of fault detection and isolation is replaced with a sequence of simpler problems. For each simpler problem, a coarser (more abstract) model of the system may be enough. In this thesis, we propose to use model reduction using the solution of the Relational Coarsest Partition problem to obtain the coarser models required for multi-resolution diagnosis.

Furthermore, to make the model reduction more effective, we use the sensor selection procedures developed in Chapter 5 to find minimal sensor sets for diagnosis at each step of multi-resolution diagnosis. One of advantages of the methods of Chapter 5 for sensor selection is that following these methods, some of the computations performed for sensor selection at the lowest (finest) level of diagnosis can be used for sensor selections for models at higher levels of abstraction. To our knowledge, the use of model reduction through the Relational Coarsest Partition problem, along with proper sensor selection to improve the efficiency of abstraction to develop models for multi-resolution fault diagnosis has not been previously studied in the DES literature.

As mentioned before, the models used at different steps of a multi-resolution diagnosis are simpler than the original plant model. There are mainly two reasons for this. First in each step of diagnosis, we solve a simpler problem. For instance, the first involves detection of fault (and no isolation is done), and the second step involves the isolation of fault group. For finding these simpler questions, coarser models of plant should be enough. Second, as the diagnosis progresses from step to step, the range of possible faults is narrowed, and we can isolate the subsystem in which the fault has occurred. At this stage, for isolating the failure mode, either the models of the other subsystems can be ignored or a very simplified (coarse) model of the other subsystem may be enough. This significantly reduces the plant model needed for fault diagnosis in the following stages. As an illustrative example, we have applied the proposed method to fault diagnosis in an ozone generator plant.

We first develop our framework for the single-failure scenario and then discuss its extension to the case of simultaneous failures. Furthermore we discuss the issue of online reconfiguration of the multi-resolution fault diagnosis (when the plant is operational). It turns out the sensor selection is the most computationally expensive step of design. We show that the sensor selection algorithms can provide an alternative method for sensor selection that does not involve any diagnosability test online and hence has significantly lower complexity. This comes at the expense of more offline (design stage) computations.

In Chapter 7, we present a summary of our results. At the same time, the main contributions developed in the preceding chapters are presented along with the discussion of further research directions.

Finally, A bibliography lists all of the references made in this thesis.

Chapter 2

PRELIMINARIES

In this chapter, we briefly review the background material on discrete event system and model-based diagnosis, which we will use in the following chapters. The key notations and definitions in this chapter are presented along with simple examples.

2.1 Discrete Event Systems

It is assumed that the system of interest can be modelled as a discrete event system (DES). A DES can be thought of as a dynamic system equipped with a discrete state set and an event driven state transition structure. An event in a DES occurs instantaneously causing transition from one state to another. There are several approaches to model a discrete event system. Examples include Finite State Automata, Petri Nets, Queuing Networks and Pseudo Codes. In this dissertation, we use two well-known DES models, Moore and Mealy automata.

2.1.1 DES Model

A nondeterministic finite-state Moore automaton (generator) is defined as a six-tuple

$$G = (X, \Sigma, \delta, x_0, Y, \lambda) \tag{2.1}$$

where X, Σ and Y are the finite state, event and output sets, respectively. x_0 is the initial state, $\delta : X \times \Sigma \to 2^X$ (2^X denotes the power set of X) the transition function and $\lambda : X \to Y$ the output map.

Obviously, in a Moore automaton, the output is associated with state. An example is given in Figure 2.1, where $X = \{0, 1, 2\}$, $x_0 = 0$, $\Sigma = \{\alpha, \beta, \gamma\}$, $Y = \{a, b\}$, $\lambda(0) = a$, $\lambda(1) = b$, $\lambda(2) = a$.

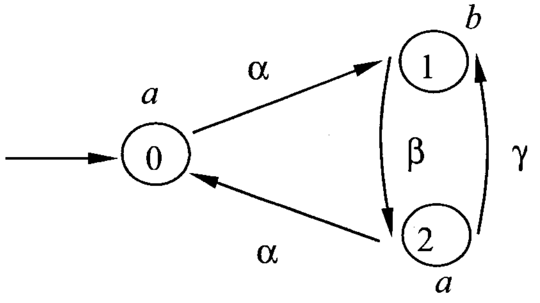


Fig. 2.1. A Moore automaton example.

On the other hand, a nondeterministic finite-state Mealy automaton is defined similar to (2.1), except that its output is associated with state transitions. That is, $\lambda : X \times \Sigma \to Y$ is the output map. Figure 2.2 is an example in which, the "Lift" event represents the lifting of the receiver, and the "Hang-up" event represents putting the receiver back on the hook. "A" is the initial state. O_1, O_2, O_3, $O_4 \in Y$ are output signals sent by the telephone to switch station. For instance, O_3 is a trigger signal (a pulse) from the phone terminal to the switch station, and $\lambda(\text{A, Lifting}) = O_3$.

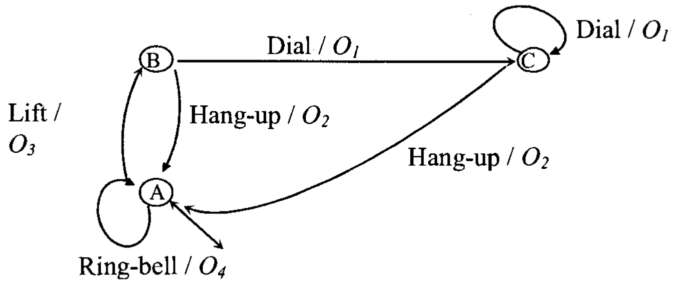


Fig. 2.2. A Mealy automaton representing a telephone unit.

In practice, each of the above models may be suitable for a certain set of applications. However, for every Mealy automaton, there exists an "equivalent" Moore generator (one that, for any given event sequence, generates the same output sequence as that of the Mealy automaton) and vice versa. In other words, Mealy and Moore automaton have the same modelling power. In the following, we assume that the system to be discussed is modelled as a finite state Moore automaton G.

For any $x_i \in X$, $\sigma_i \in \Sigma$ with $i = \{1, 2, \ldots, n\}$ and $n \geq 2$, a path $x_1 \xrightarrow{\sigma_1} \ldots \xrightarrow{\sigma_{n-1}} x_n$, is called a **cycle** if $x_1 = x_n$. The cycle is called a **simple cycle** if for all x_i, x_j $(i \neq j)$ in the cycle, $x_i \neq x_j$ except for $x_1 = x_n$. A path $x_1 \xrightarrow{\sigma_1} \ldots \xrightarrow{\sigma_{n-1}} x_n$, with $n \geq 2$, is called a **traversal trajectory** or **direct trajectory** if for all x_i, x_j in the path with $i \neq j$, $x_i \neq x_j$. For brevity, in this dissertation, we simply refer to **direct trajectory** as **trajectory**.

Definition 2.1.1 *Suppose two states x, x' in G satisfy $\lambda(x) \neq \lambda(x')$ and x' can be reached from x through a path along which the output is equal to $\lambda(x)$ (except at x'); then we say x' is* ***output-adjacent*** *to x and write $x \Rightarrow x'$ [5].* ■

2.1.2 Timed DES

If the descriptions of system components include timing constraints, then timed models can be used to describe the system. In a timed discrete event system (TDES) [23], the sequence of events occurring in the system is described with respect to the ticks of a global clock. Formally, a nondeterministic Timed DES (TDES) is defined as a collection

$$G = (X, \Sigma \cup \{\tau\}, \delta, x_0, Y, \lambda) \tag{2.2}$$

where X, $\Sigma \cup \{\tau\}$, Y are the finite state, event and output sets; x_0 is the initial state; $\delta : X \times \Sigma \cup \{\tau\} \rightarrow 2^X$ is the transition function and $\lambda : X \rightarrow Y$ is the output map. Note that here τ represents the "tick" of the global clock. Without loss of generality, we assume that the event *tick* in timed DES does not change the system output. If the assumption fails, for example, if $x_1 \xrightarrow{\tau} x_2$ and $\lambda(x_1) \neq \lambda(x_2)$, then we add an event σ and a state x_2' with $\lambda(x_2') = \lambda(x_1)$ and replace $x_1 \xrightarrow{\tau} x_2$ with $x_1 \xrightarrow{\tau} x_2' \xrightarrow{\sigma} x_2$. After applying the above change to all "*tick*" transitions that violate the assumption, we obtain a TDES in which tick transitions do not change output. It is assumed that the TDES is **activity-loop-free**, that is, it does not contain a cycle of non-tick events [63].

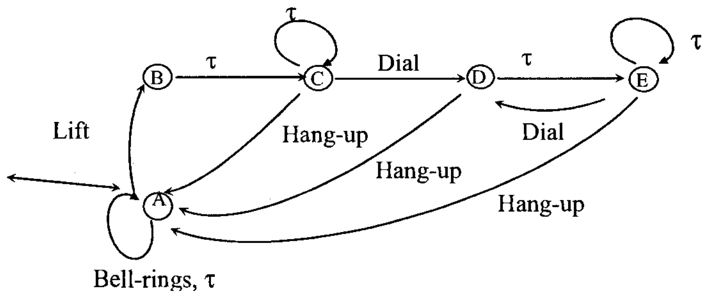


Fig. 2.3. A TDES automaton representing a telephone unit.

Furthermore, we assume G does not have deadlock (terminal) states since the tick event happens periodically and, at each state $x \in X$, if no nontick event is enabled, the tick should be enabled ($\delta(x, \tau) \neq \emptyset$).

A TDES model for the telephone unit in the previous example is shown in Figure 2.3. Note that the outputs are not shown to avoid cluttering the figure.

2.2 Fault Diagnosis

2.2.1 Fault Diagnosis Using DES Models

In this section, we review failure modelling and diagnoser design in a state-based framework [5]. Consider the finite-state Moore automaton G in (2.1). Suppose there are p **failure modes** $F_1, F_2, \ldots, F_p$ in discrete-event system G. The event set can be partitioned into $\Sigma = \Sigma_N \cup \Sigma_f$, where $\Sigma_f = \{f_1, \ldots, f_p\}$ is the set of failure events and Σ_N, the set of non-failure events. As a result of the failure event f_i, the failure mode F_i develops in the plant.

The **condition** set of system will be $\mathcal{K} = \{N, F_1, F_2, \dots, F_p, F_{1,2}, F_{1,3}, \dots, F_{p-1,p}, \dots, F_{1,\dots,p}\}$. It is assumed that X can be partitioned according to system condition. In other words, $X = X_N \cup (\cup_{i=1}^{p} X_{F_i}) \cup \dots \cup X_{F_{1,\dots,p}}$.

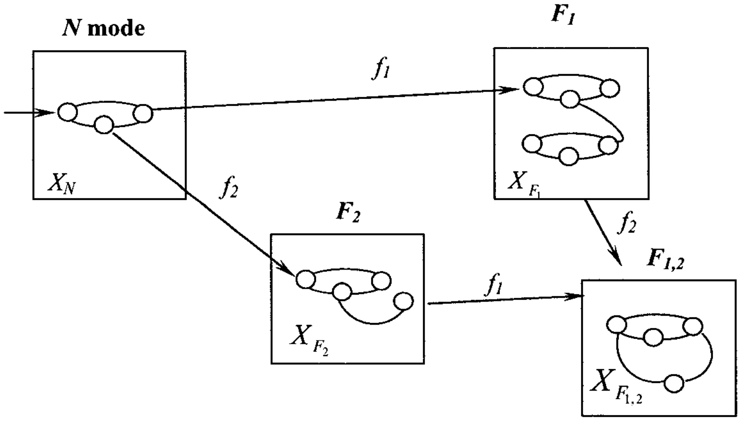


Fig. 2.4. Failure event and failure condition.

Figure 2.4 shows an example to explain the concepts of failure event and failure mode. In this example, the system has two failure events ($\Sigma_f = \{f_1, f_2\}$, $\Sigma_f \subseteq \Sigma$), three failure modes ($F_1$, F_2 and $F_{1,2}$). The F_1 (resp. F_2) subautomaton of the model describes the system behavior when f_1 (resp. f_2) has occurred. Similarly, $F_{1,2}$ describes the system after both f_1 and f_2 have occurred. The state set of the system is $X = (\cup_{i=1}^{2} X_{F_i}) \cup X_N \cup X_{F_{1,2}}$. By definition, the **condition set** of the system is $\mathcal{K} = \{N, F_1, F_2, F_{1,2}\}$.

Define $\kappa : x \rightarrow \mathcal{K}$ such that for every $x \in X$, $\kappa(x)$ is the condition of the system in that state $x : \kappa(x) = N$ if $x \in X_N$, $\kappa(x) = F$ if $x \in X_F$. Also extend the condition map to subsets of X according to $\kappa : 2^X \rightarrow 2^{\mathcal{K}}$ with $\kappa(z) = \{\kappa(x) | x \in z\}$, for any $z \subseteq X$.

We assume, without loss of generality, that all failure events are unobservable. That is, the occurrence of failure does not result in an output change that identifies the failure.

As discussed in the previous chapter, Sampath *et al.* [4] introduced the concept of diagnoser to perform diagnosis. In this event-based approach, the diagnoser can be viewed as an extended observer for G which gives an estimate of the current state of the system and information on potential past occurrences of *failure events.* In the state-based framework [5], however, the diagnoser takes the output sequence of the system $(y_1 y_2 \dots y_k)$ as input and generates at its output an estimate of the system state z_k and thus the *condition* of the system $\kappa_k = \kappa(z_k)$ at the time that y_k was generated (Fig. 2.5).

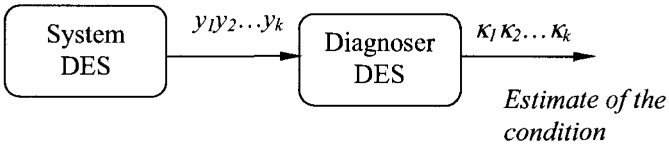


Fig. 2.5. System and Diagnoser.

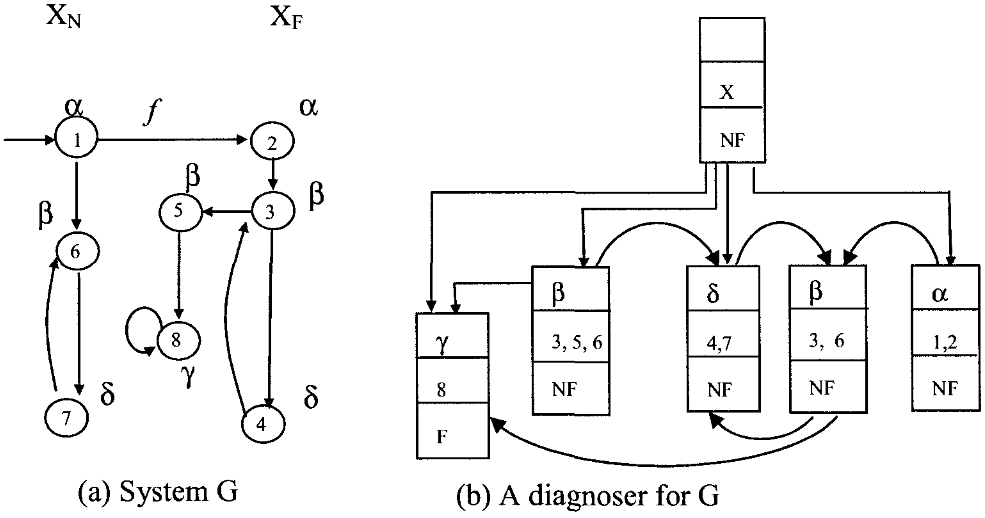
In [5], the diagnoser for the DES G is defined as a DES denoted by D with: $D = (Z \cup \{\underline{z}_0\}, Y, \pi, \underline{z}_0, \mathcal{K}, \kappa)$, where $Z \cup \{\underline{z}_0\}$, Y, $\mathcal{K}$ are the state, event and output sets of D. $\underline{z}_0 := (z_0, 0)$ is the initial set, with $z_0 \in 2^X - \{\emptyset\}$; $Z \subseteq 2^X - \{\emptyset\}$, and $\pi : Z \cup \{\underline{z}_0\} \times Y \rightarrow Z$ is the partial transition function; the output map is

$\kappa : Z \cup \{\underline{z}_0\} \to \mathcal{K}$. Given state estimate z_k, and upon observing y_{k+1}, the state estimate is updated according to:

$$z_1 = z_0 \cap \lambda^{-1}(\{y_1\}) \; (k = 0)$$

$$z_{k+1} = \pi(z_k, y_{k+1}) = \{x | \lambda(x) = y_{k+1} \text{ and } (\exists x' \in z_k : x' \Rightarrow x)\} \; (k \geq 1)$$

Consider the system G in Figure 2.6(a) with a set of failure events $\Sigma_F = \{\sigma_f\}$; the output map is: $\lambda(1) = \lambda(2) = \alpha$, $\lambda(3) = \lambda(5) = \lambda(6) = \beta$, $\lambda(4) = \lambda(7) = \delta$, $\lambda(8) = \gamma$.



(a) System G

(b) A diagnoser for G

Fig. 2.6. A simple example of DES fault diagnosis.

The diagnoser for G is shown in Figure 2.6(b). Initially, the system state is assumed unknown; thus $z_0 = X$. The condition of z_0 is uncertain. If an output "α" is observed, then the system state must be "1" or "2" and the condition of the system will still be uncertain "N, F". Similarly, if "β" is observed first, then the state must be "3", "5" or "6", and the condition of the state is "N, F". If "γ" is observed after "β", then the state must be "8" and the diagnoser indicates that the condition estimate of the system is "F".

This diagnoser has a cycle $\{3, 6\} \rightarrow \{4, 7\} \rightarrow \{3, 6\}$ corresponding to the output sequence $\beta\delta\beta$, which is called a "**F-indeterminate**" cycle because the condition estimate in the diagnoser cycle is "N, F" and hence uncertain. Note that there are two cycles in G with the same output sequence $\beta\delta\beta$: states $3-4-3$, in the failure mode (F), and states $6-7-6$, in the Normal mode (N). The diagnoser cannot distinguish between these two cycles. Therefore, if the system is trapped in the faulty cycle $3-4-3$, the diagnoser will not be able to detect the failure. Fault diagnosability is reviewed in Section 2.3.

Since failure diagnosis relies on the output sequence and output-adjacent states, it is useful to convert the original system G into the **Reachability Transition System** (RTS) [5] which contains information about the system output sequences in a compact form corresponding to G. The RTS corresponding to G is defined to be the transition system $\tilde{G} = (X, R, Y, \lambda)$, which has X, Y and λ as the state set, output set and output map; R is a binary relation $R \subseteq X \times X$ and $(x_1, x_2) \in R$ if and only if x_2 is output-adjacent to x_1 [5]. Figure 2.7 shows the RTS from the original system in Figure 2.6.

From a computational viewpoint, RTS can be computed in $\mathcal{O}(|X|^2 + |X| \cdot |\theta|)$ time because a breadth-first search reachability analysis for each $x \in X$ can be done in $\mathcal{O}(|X| + |\theta|)$ time. Here θ is the set of transition of G, and $|\theta|$ is its cardinality. $\tilde{G}$ can be viewed as a modified version of G which contains the information about transitions among output-adjacent states in a compact form.

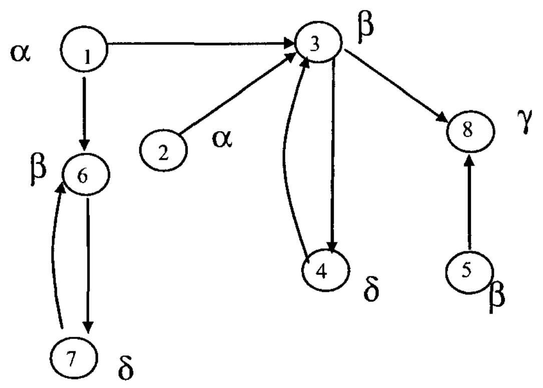


Fig. 2.7. Reachability transition system for the system in Figure 2.6.

The number of the states of diagnoser in the worst case is exponential in the number of plant states $|X|$. Therefore, it might be better to store the reachability transition system in memory and perform diagnostic computations **online**: having z_k and the output y_{k+1}, use RTS $\tilde{G}$ to compute z_{k+1} and then update condition estimate. This method of **online implementation** is particularly useful when initial state estimate z_0 is unknown at the design stage. This is the case in multi-resolution diagnosis system, to be discussed in Chapter 6.

In [5], an equivalence relation on the state set of $\tilde{G}$ based on the Relational Coarsest Partition problem is used to reduce the RTS $\tilde{G}$ and thus the diagnosis system. Specifically, the system state set is partitioned into the blocks such that states in each block contain the same information about the present and future estimations of the system's condition. Then the resulting equivalence relation is used to reduce RTS $\tilde{G}$.

Consider the RTS $\tilde{G} = (X, R, Y, \lambda)$. For every x_1, $x_2 \in X$, let $x_2 \in R(x_1)$ iff $(x_1, x_2) \in R$, i.e., $x_1 \Rightarrow x_2$. Let $\pi = \{B_1, \ldots, B_{|\pi|}\}$ be a partition of X, with B_i denoting the blocks of π. The partition π is said to be compatible with R if and only if whenever x and x' are in the same block B_i, then for any block B_j, $R(x) \cap B_j \neq \emptyset$ iff $R(x') \cap B_j \neq \emptyset$. Define $\prod$ be the set of partitions compatible with R. Let ker refers to the equivalence kernel of the corresponding map and $\wedge$ denotes the meet operation in the lattice of equivalence relations [64]. The set $\{\pi \in \prod, \pi \leq \ker \lambda \wedge \ker \kappa\}$ is closed under $\vee$, the join operation in the lattice of equivalence relations, and therefore has a unique supremal element which is the coarsest partition compatible with R and finer than $\ker \lambda \wedge \ker \kappa$. $\pi^\star$ can be used to find a reduced version of $\tilde{G}$. Let $P : X \rightarrow X/\pi^\star$ be the canonical projection. The **reduced RTS** $\bar{G}$ can be defined [5] as $\bar{G} = (\bar{X}, \bar{R}, Y, \bar{\lambda})$, where $\bar{X} = X/\pi^\star$; for all $\bar{x}_1, \bar{x}_2 \in \bar{X}$, $(\bar{x}_1, \bar{x}_2) \in \bar{R}$ (i.e., $\forall x \in \bar{x}_1, \exists x' \in P^{-1}\bar{x}_2$ and $(x, x') \in R$); and for all $\bar{x} \in X$, $\bar{\lambda}(\bar{x}) = \lambda(x)$ for any $x \in \bar{x}$.

Similarly define $\bar{\kappa} : \bar{X} \rightarrow \mathcal{K}$ according to $\bar{\kappa}(\bar{x}) = \kappa(x)$ for any $x \in \bar{x}$. Since $\pi^\star \leq \ker \lambda \wedge \ker \kappa$, $\bar{\lambda}$ and $\bar{\kappa}$ are well-defined. The canonical projection of a subset $z \subseteq X$ to be $P(z) := \bigcup\{[x] | x \in z\}$. $\bar{G}$ is the reduced version of $\tilde{G}$ and a diagnoser designed based on $\bar{G}$ with initial state estimate $\bar{z}_0 := Pz_0$ will generate the same condition estimates as that designed based on $\tilde{G}$ [5].

2.2.2 Fault Diagnosis Using Timed DES Models

Let $y_1 y_2 \ldots y_k$ be the output sequence generated by TDES G and t_j, the number of ticks occurred between y_{j-1} and y_j $(2 \leq j \leq k)$. A (standard) diagnoser [23] based on the output sequence $y_1 y_2 \ldots y_k$ and the timing sequence $t_2 \ldots t_k$, generates an estimate of the system state $z_k(t) \in 2^X - \{\emptyset\}$ and the condition of the system $\kappa(z_k(t))$, where t denotes the number of clock ticks occurred after the last output symbol y_k was generated. Every time a clock tick occurs or a new output is generated, the state and condition estimates are updated.

Since failure diagnosis relies on the output sequences and output-adjacent states, it is useful to convert the original system G into a **Timed Reachability Transition System** (Timed RTS) [23] which contains information about the system output sequences and the corresponding timing information in a compact form. The timed RTS corresponding to G is defined to be the transition system $\tilde{G} = (X, Y, T, \lambda)$, where X, Y, λ are the state set, output set and output map; $T : X \times X \rightarrow 2^{\mathbb{N}}$ ($\mathbb{N} = \{0, 1, 2, \ldots\}$) is the transition-time function. For output-adjacent states x and x'', $T(x, x'')$ is the set of times (in ticks) that it takes on the paths from x to x'' with constant output (until x'' is reached). If x and x'' are not output-adjacent, then $T(x, x'') = \emptyset$ by definition. $\tilde{G}$ can be viewed as a modified version of G which contains the information about transitions among output-adjacent states in a compact form.

2.3 Fault Diagnosability

The example in Section 2.2.1 illustrates the important issues in DES fault diagnosis and the notion of diagnosability. Diagnosability is defined as follows [5].

Definition 2.3.1 *[5] A permanent failure F of system G is said to be **diagnosable** if there exist an integer $N \geq 0$ such that following both of the occurrence of failure and the start of diagnosis, the failure can be detected and isolated after the occurrence of at most N events in the system.* ■

Definition 2.3.2 *[5] If the occurrence of a failure mode F_i can be directly concluded from the generation of an output symbol $y \in Y$, then y is called F_i-**indicative**. In single-failure scenario, y is F_i-indicative if $\lambda^{-1}(\{y\}) \subseteq X_{F_i}$.* ■

A state z of the diagnoser corresponding to a state estimate for the plant is called F_i **-certain** if $\kappa(z)$, the corresponding estimate of the system's condition, indicates that the failure has occurred. A state z of the diagnoser is defined as F_i **-uncertain** if $\kappa(z)$, the corresponding estimate of the system's condition is consistent with the occurrence of F_i but, doesn't conclusively indicate that the failure has occurred. In single-failure scenario, z is F_i -certain if $\kappa(z) = \{F_i\}$, and F_i-uncertain if $\{F_i\} \subseteq \kappa(z)$ but $\kappa(z) \neq \{F_i\}$.

Definition 2.3.3 *[5] Suppose $z^1, \dots, z^m$ is a cycle of F_i -uncertain states of the diagnoser. The cycle is called F_i **-indeterminate** if there exist $l \geq 1$ and $x_1^j, x_2^j, \dots, x_l^j \in z^j$, for all $1 \leq j \leq m$ such that $x_k^j \in X_{F_i}$ for all $1 \leq j \leq m$, $1 \leq k \leq l$ and $x_1^1, x_1^2, \dots, x_1^m, x_2^1, \dots, x_2^m, \dots, x_l^1, \dots, x_l^m$ form a cycle in the RTS.* ■

Intuitively, a failure would not be diagnosable if after it occurs, the system stops generating new output symbols (unless the last output symbol is **failure indicative**). Also, a failure would be undiagnosable if after its occurrence, the system can generate a periodic output sequence that throws the diagnoser into a cycle of failure uncertain states, (i.e., failure indeterminate cycles). For instance, in the example in Section 2.2.1, $\{3,6\} \rightarrow \{4,7\} \rightarrow \{3,6\}$ is an F-indeterminate cycle and therefore F is undiagnosable. Theorem 2.3.1 provides necessary and sufficient conditions for diagnosability for permanent failures in single-failure scenarios, assuming $z_0 = X$.

Theorem 2.3.1 *[5]*

Assume single-failure scenario and $z_0 = X$. A permanent failure F_i is diagnosable if and only if

1. *From every $x \in X_{F_i}$, there is (at least) one transition to another state in X_{F_i} unless $\lambda(x)$ is F_i-indicative.*

2. *There is no cycle in X_{F_i} consisting of states having the same output symbol unless the output symbol is F_i-indicative.*

3. *There are no F_i-indeterminate cycles in the diagnoser.* ■

As for the fault diagnosis in the timed DES, we have the following definition.

Definition 2.3.4 *A permanent failure mode F_i is **time-diagnosable** if there exists an integer $T_i \geq 0$ such that following both the occurrence of the failure and initialization of the diagnoser, F_i can be detected and isolated after the occurrence of at most T_i ticks.* ■

Similar necessary and sufficient conditions for time-diagnosability are given in [23] which involve F-indeterminate cycles of diagnosers. Since the conditions for diagnosability involve diagnosers, their verification requires exponential computational complexity in the number of system states.

Chapter 3

PROBLEM FORMULATION

In this chapter, we provide a brief description of the problem and outline of discussion in the rest of the thesis. A running example involving an ozone generator plant is also introduced.

Suppose that the plant can be modelled as a nondeterministic finite state Moore automaton (**FSMA**) $G = (X, \Sigma, \delta, x_0, Y, \lambda)$ as described in Equation 2.1. This model describes the behavior of the system in both normal and faulty situations.

We assume that the failure modes are **permanent**; in other words, after the occurrence of a failure, the failure mode remains in the plant indefinitely. Note that simultaneous failures are assumed possible. For example, in a plant with two failure modes F_1 and F_2, the plant can be in one of four conditions: N (normal), $F_1, F_2, F_{1,2}$, where $F_{1,2}$ refers to the simultaneous occurrence of both failures.

Fig. 3.1 shows the state transition graph of a plant with permanent failure modes. Each circle corresponds to a block in the partition of the plant state set X based on plant condition. We observe that the transition graph of a plant with permanent failure modes has a tree structure with N (normal) condition as the root. One of the consequences of this structure is that G cannot have a cycle with states in more than a single condition. For instance, there is no cycle whose states are in X_{F_1} and $X_{F_{1,2}}$. As a result, the set of cycles of G is the union of cycles of individual conditions (i.e., individual blocks in Fig. 3.1).

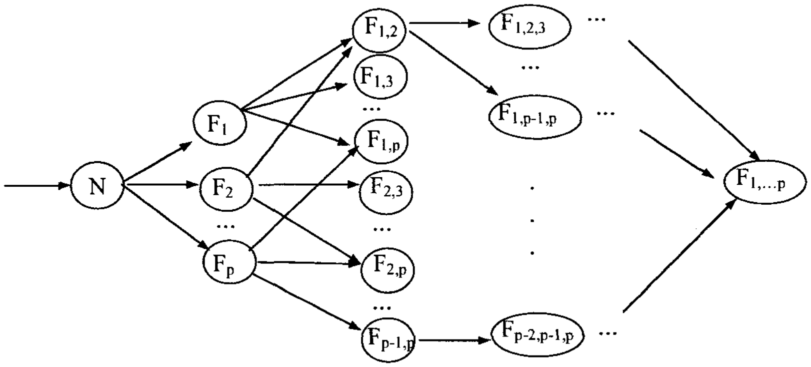


Fig. 3.1. System Structure with Permanent and Simultaneous Failures.

Remark 3.0.1 *We assume the failures are permanent for simplicity. In the case of* ***non-permanent*** *failures, since the plant may not dwell in the faulty condition long enough, the detection of non-permanent failures can be done using the event-based approach [4] to detect the occurrence of non-permanent failures. Alternatively, following Remark 1 of [5], we can convert the problem of detection of occurrence of failure event to a problem of detecting the presence of a* ***permanent*** *fault. Therefore, in this dissertation, without loss of generality, we assume that all failure modes are permanent.* ■

Now in order to detect and isolate the failures in such dynamic systems and manage the computational complexity, we introduce an algorithm which uses a sequence of models of the plant, with increasing resolutions, to narrow down the range of possible diagnosis step by step and to finally isolate the failure. In this way, the original problem of failure diagnosis is broken down into a sequence of simpler problems. This approach is similar to Branch and Bound techniques used in Operations Research. In this chapter, we describe the proposed **Multi-Resolution Diagnosis** techniques and some design issues briefly. The details will be provided in future chapters.

Let us assume that the failure conditions are grouped into l failure groups $\mathcal{F}^{(1)}, \mathcal{F}^{(2)}$, ..., $\mathcal{F}^{(l)}$. Failure conditions can be grouped, for instance, based on the subsystems that the failures occur. In other words, $\mathcal{F}^{(1)}$ contains the faulty conditions that may develop in subsystem 1 (and so on). Failure grouping will be discussed in more detail in Chapter 6. Let $F_1^{(i)}, \ldots F_{p_i}^{(i)}$ denote the faulty conditions in group $\mathcal{F}^{(i)}$: $\mathcal{F}^{(i)} = \{F_1^{(i)}, \ldots, F_{p_i}^{(i)}\}$ $(1 \leq i \leq l)$. Therefore, the set of failure conditions and groups can be represented in the form of a hierarchy shown in Fig. 3.2. Here $\mathcal{F} = \cup_{i=1}^{l} \mathcal{F}^{(i)} = \{F_1^{(1)}, \ldots, F_{p_1}^{(1)}, F_1^{(2)}, \ldots, F_{p_l}^{(l)}\}$ is the set of faulty conditions. In Fig. 3.2, $\mathcal{N} = \{N\}$, where N is the normal condition. The multi-resolution diagnosis proposed in this thesis is designed based on grouping of conditions in Fig. 3.2. Note that the three-level failure hierarchy in Fig. 3.2 can be replaced with hierarchies having more than three levels. In this thesis, for simplicity, we present our results based on the three-level hierarchy in Fig. 3.2.

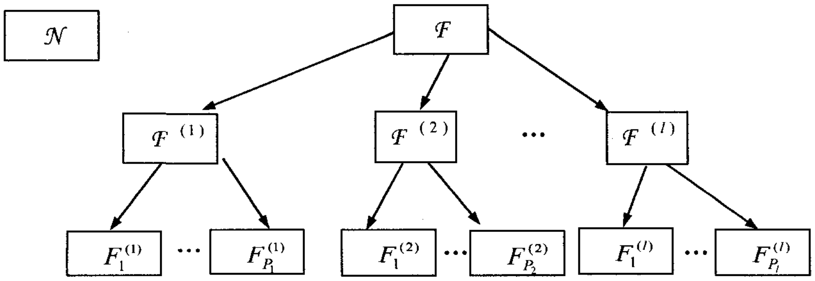


Fig. 3.2. A multi-resolution diagnosis example.

In the multi-resolution diagnosis system proposed in this dissertation, a diagnoser is designed to detect faulty operation. In other words, the diagnoser determines whether the condition of plant is $\mathcal{N}$ or $\mathcal{F}$. Let $\mathbb{K}_1 = \{\mathcal{N}, \mathcal{F}\}$ denote the **level-one condition set**. Once a faulty behavior is detected, another diagnoser is used to identify the faulty group the plant's condition belongs to. We shall refer to this diagnoser and the set $\mathbb{K}_2 = \{\mathcal{F}^{(1)}, \dots, \mathcal{F}^{(l)}\}$ as the second-level diagnoser and condition set. Once the faulty group is identified as, say $\mathcal{F}^{(i)}$, then a third-level diagnoser is invoked to isolate the faulty condition of the plant. Thus, $\mathbb{K}_3 = \mathcal{F}^{(i)}$ $(1 \leq i \leq l)$ will be the third-level condition set.

The proposed diagnosis method requires plant models at various resolution. We obtain these models using model reduction based on the solution to the Relational Coarsest Partition problem. For each diagnosis step, a minimal sensor set is chosen to have a coarser output map and hence to improve the efficiency of model reduction. For sensor selection, in turn, we need efficient algorithms for testing diagnosability. Therefore, in this thesis, we start our work by developing efficient algorithms for testing diagnosability and sensor selection. Then we will examine multi-resolution fault diagnosis.

We use the following as a running example in this thesis [1].

Example 3.0.1 *Consider the ozone generation plant in Fig. 3.3 in which oxygen is converted to ozone using a high-voltage power supplied by a Power Supply Unit (PSU). A cooling system is used to cool the generator since ozone decomposes at high temperatures. The system also contains an oxygen inlet valve (V_1), ozone outlet*

[1] This physical system is adopted from [62] with some modifications to be explained later.

valve (V_2) and cooling water valve (V_3). The set of sensors consists of flow sensors c_{f1} and c_{f2}, pressure sensors c_{p1} and c_{p2}, and the ozone concentration analyzer c_c.

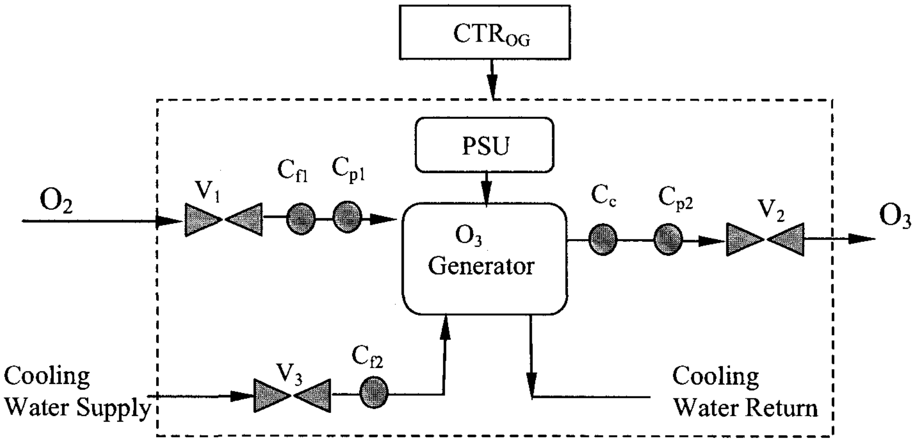


Fig. 3.3. An Ozone generator plant.

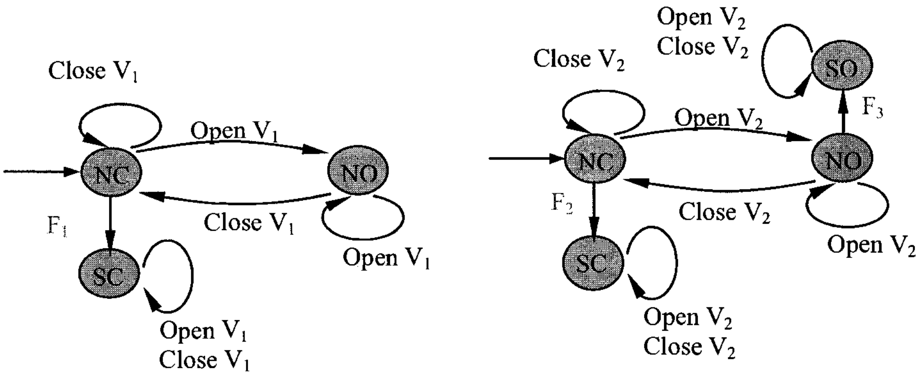


Fig. 3.4. Valve 1 and 2.

The power supply unit (Fig. 3.5) has two operation events: "Run PSU" and "Stop PSU"; one failure "PSU fail" and three corresponding states: "NS"(normal stop), "NR" (normal run) and "F" (failed). The ozone generator is modelled with two states "O_3L" (ozone low) and "O_3N" (ozone normal) (Fig. 3.5).

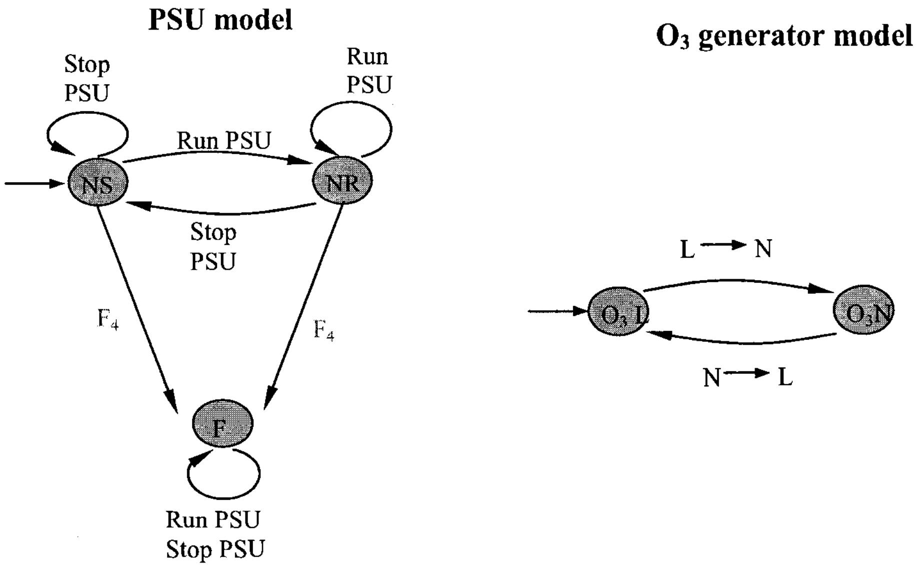


Fig. 3.5. Power supply unit and ozone generator models.

We consider the following six failure modes in this water treatment system: V_1 stuck-closed (F_1), V_2 stuck-closed (F_2), V_2 stuck-open (F_3), power supply unit (PSU) failed (F_4) and V_3 stuck-closed (F_5), V_3 stuck-open(F_6). For brevity, we assume single-failure scenario.

The model of V_1 and V_2 are shown in Fig. 3.4. "NC", "NO", "SC", "SO" stand for Normal-Closed, Normal-Open, Stuck-Closed and Stuck-Open. The model of V_3 is similar to V_2.

The system has a controller CTR_{OG} to manage the operations. The controller CTR_{OG} sends the start up and shut down commands to rotate the ozone generation system in and out of service on a periodic basis. The start up sequence is : open $V_3 \rightarrow$ open $V_1 \rightarrow$ open $V_2 \rightarrow$ Run PSU. After a certain amount of time, the PSU is stopped *and once the ozone concentration becomes low, the valves are closed in the order V_2, V_1 and V_3 (Fig. 3.6).*

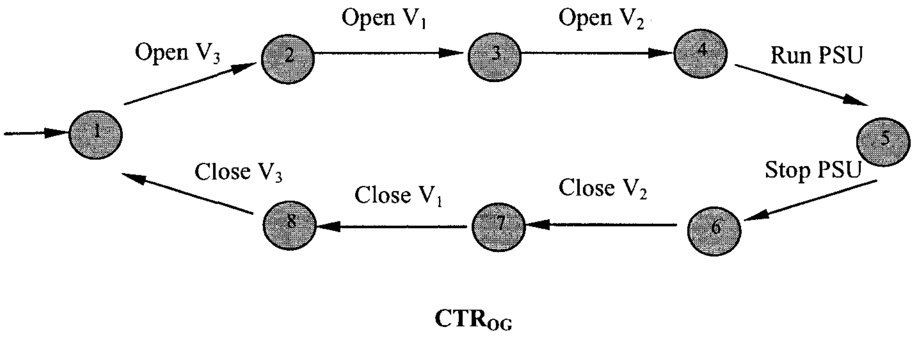


Fig. 3.6. Controller model.

Further details about this system can be obtained in [62]. The condition set of this plant is $\mathcal{K} = \{N, F_1, F_2, F_3, F_4, F_5, F_6\}$. Now we can demonstrate our problem through this physical example: Our objective is to first detect the occurrence of failures. If a failure occurs, then we attempt to narrow down the possible range of possible failures. For instance, we may wish to determine if the failure is in cooling water pipeline, in oxygen supply pipeline or in power supply unit. Once we find this failure group, we finally isolate the failure mode within the group.

■

Chapter 4

DISCRETE EVENT SYSTEM MODEL AND DIAGNOSABILITY

In this chapter, we present two polynomial-time algorithms for testing failure diagnosability in untimed and timed discrete-event systems in a state-based framework. The test for timed discrete-event systems, in particular, uses the information about the timing of events (represented in the timed transition graph of the timed system) gathered in the set of transition-time sets of the timed Reachability Transition System (timed RTS) to verify diagnosability. Compared with other polynomial diagnosability tests developed for untimed systems and adapted for timed systems, this new test does not reduce the worst-case computational complexity; however, as shown using examples, if time RTS has been computed, say for diagnoser design following [5], the test may considerably reduce the computations of testing diagnosability. Time RTS can be efficiently computed when the transition-time sets can be represented as the union of a bounded number of intervals. Sufficient conditions are provided under which the transition-time sets have the aforementioned bounded representation.

4.1 Testing Diagnosability In Untimed DES

In this section, we present a new set of necessary and sufficient conditions for diagnosability of permanent faults in fault scenarios for untimed DES. Based on these conditions, we also construct a test for diagnosability which has polynomial time complexity.

Assume that the system to be diagnosed can be modelled as a nondeterministic finite-state Moore automaton $G = (X, \Sigma, \delta, x_0, Y, \lambda)$ where X, Σ and Y are the finite state set, event set and output set, respectively. x_0 is the initial state, $\delta : X \times \Sigma \rightarrow 2^X$ the transition function and $\lambda : X \rightarrow Y$ the output map. This model describes the behavior of the system in both normal (N mode) and faulty situations (F modes). We assume that the plant has n **failure modes** $F_1, F_2, \ldots, F_n$. The event set can be partitioned into $\Sigma = \Sigma_N \cup \Sigma_f$, where $\Sigma_f = \{f_1, \ldots, f_n\}$ is the set of failure events and Σ_N, the set of non-failure events. As a result of failure event f_i, failure mode F_i develops in the plant. Simultaneous failures are assumed possible. For example, in a plant with two failure modes F_1 and F_2, the plant can be in one of four **conditions**: N(normal), $F_1, F_2, F_{1,2}$, where $F_{1,2}$ refers to the simultaneous occurrence of both failures. Let $\mathcal{K}$ denote the **condition set** and $\mathcal{F}$ the set of **faulty conditions**. Furthermore, let $\mathcal{F}_i$ denote the set of faulty conditions in which failure mode F_i is present and $\bar{\mathcal{F}_i} = \mathcal{F} - \mathcal{F}_i$. Therefore, for instance in a plant with two failure modes F_1 and F_2, we have $\mathcal{K} = \{N, F_1, F_2, F_{1,2}\}$, $\mathcal{F} = \{F_1, F_2, F_{1,2}\}$, $\mathcal{F}_1 = \{F_1, F_{1,2}\}$, and $\bar{\mathcal{F}_1} = \{F_2\}$. It is assumed that the finite state set can be partitioned according to system condition. For example, for the case of two failure modes ($n = 2$), $X = X_N \cup X_{F_1} \cup X_{F_2} \cup X_{F_{1,2}}$. The set of states corresponding to condition set

$\mathcal{F}$ (resp. $\mathcal{F}_i$) is denoted by $X_{\mathcal{F}}$ (resp. $X_{\mathcal{F}_i}$). The **condition map** $\kappa : X \to \mathcal{K}$ returns the condition of each state. This map can be extended to subsets of X: for $z \subseteq X, \kappa(z) = \cup\{\kappa(x)|x \in z\}$.

To develop a test for diagnosability, we begin with introducing a few definitions. Let $\mathcal{F} = \{F_i \mid i = 1, 2, \ldots, p\}$ be the set of failure modes. For a failure mode F_i, $X_{\bar{F}_i} := \bigcup_{\substack{j=1 \\ j\neq i}}^{p} X_{F_j}$. $\bar{F}_i$ represents all failure modes except F_i in the system.

Definition 4.1.1 *The* ***output language*** $L_o(G, x)$ *generated by* G *from the state* $x \in X$ *is defined as*

$$L_o(G,x) := \{y_1 y_2 \ldots y_m \subseteq Y^+ \mid y_1 = \lambda(x),\ \exists x_1, x_2, \ldots, x_m \in X : x_1 = x,$$
$$\forall\, 2 \leq i \leq m : x_{i-1} \Rightarrow x_i,\ \mathit{and}\ y_i = \lambda(x_i)\}$$

■

$L_o(G, x)$ represents the set of all possible (finite) output sequences generated by G starting from state x. $L_o(\tilde{G}, x)$ is defined similarly. Obviously, $L_o(G, x) = L_o(\tilde{G}, x)$.

By Definition 2.3.2 in Section 2.3, in a single failure scenario, y is F_i-indicative iff $\lambda^{-1}(\{y\}) \subseteq X_{F_i}$.

Let $\tilde{G}_{F_i}$ denote the sub-generator of $\tilde{G}$ consisting of the states in X_{F_i} only. Similarly, let $\tilde{G}_N$ and $\tilde{G}_{N,\bar{F}_i}$ be the sub-generators of $\tilde{G}$ corresponding to the states in X_N and $X_{\bar{F}_i} \cup X_N$ (The initial states of $\tilde{G}_{F_i}$, $\tilde{G}_N$ and $\tilde{G}_{N,\bar{F}_i}$ are left undefined.). The following theorem provides necessary and sufficient conditions for diagnosability that, as we will see later, can be verified in polynomial time. The diagnosability test resulting from this theorem is essentially the equivalent of the test in [31] for the state-based framework of [5].

Theorem 4.1.1 *Assume single failure scenario and $z_0 = X$. A permanent fault F_i is diagnosable if and only if*

1. *For any $x \in X_{F_i}$, there is at least one transition to another state in X_{F_i}unless $\lambda(x)$ is F_i-indicative;*

2. *There is no cycle in X_{F_i} consisting of states having the same output unless the output symbol is F_i-indicative;*

3. *For any $x \in X_{F_i}$, and $x' \in X_N \cup (\cup_{j \neq i} X_{F_j})$ satisfying $\lambda(x) = \lambda(x')$, we have*

$$\{s | s \in L_o(\tilde{G}_{N,\bar{F}_i}, x') \cap L_o(\tilde{G}_{F_i}, x), |s| \geq |X|^2\} = \emptyset$$

■

Condition (1) states that there should be no deadlock state in X_{F_i} with no transition out of the state unless the output in that state can be generated only when F_i has occurred. In [5] such output is called F_i-indicative. Similarly, condition (2) states that there should be no cycles with constant output in X_{F_i} unless the constant output is F_i-indicative. Finally, condition (3) states that there should be no common output cycle in X_{F_i} and $X - X_{F_i} = X_N \cup X_{\bar{F}_i}$ (otherwise F_i cannot be distinguished and hence will be undiagnosable).

Proof

(Necessity)

Conditions (1) and (2) guarantee that after F_i occurs, if the output sequence terminates, it will do so in an F_i-indicative symbol. If one of these conditions does not hold, the output sequence can end in a non-F_i-indicative output. If the fault diagnosis system is initialized after this non-F_i-indicative symbol is generated and afterwards no new output is generated, then the failure will not be detected and isolated. This shows the necessity of conditions (1) and (2).

Suppose conditions (1) and (2) hold. We show that if condition (3) does not hold, the diagnoser will have an F_i- indeterminate cycle, and therefore, by Theorem 2.3.1, F_i is undiagnosable. If condition (3) does not hold, then for some $x \in X_{F_i}$ and $x' \in X_N \cup (\cup_{j \neq i} X_{F_j})$ with $\lambda(x) = \lambda(x')$, there exist $s = y_1 y_2 \dots y_q$ $(q = |X|^2)$, with $s \in L_o(\tilde{G}_{N,\bar{F}_i}, x') \cap L_o(\tilde{G}_{F_i}, x)$. Thus, there exist x_i, x'_i $(i = 1, \dots, q)$ such that $x_i \in X_{F_i}$, $x'_i \in X_N \cup X_{\bar{F}_i}$, with $x_1 = x$, $x'_1 = x'$, $\lambda(x_i) = \lambda(x'_i) = y_i$ $(1 \leq i \leq q)$, and $x_i \Rightarrow x_{i+1}$, $x'_i \Rightarrow x'_{i+1}$ $(1 \leq i \leq q-1)$. The corresponding state transitions in $\tilde{G}_{F_i}$ and $\tilde{G}_{N,\bar{F}_i}$ can be represented as follows:

$$\begin{array}{llll}
\tilde{G}_{F_i} & x = x_1 \Rightarrow & x_2 \Rightarrow \dots & x_q \\
\tilde{G}_{N,\bar{F}_i} & x' = x'_1 \Rightarrow & x'_2 \Rightarrow \dots & x'_q \\
\text{Output} & (y_1) & (y_2) & (y_q)
\end{array}$$

Since $|X_{F_i}| \times |X_N \cup X_{\bar{F}_i}| \leq lcm(|X_{F_i}|, |X_N \cup X_{\bar{F}_i}|) \leq |X|^2$, there exist m, k, with $1 \leq m < k \leq |X|^2$ such that $x_m = x_k$, and $x'_m = x'_k$. Therefore, there exist two cycles, one in $\tilde{G}_{F_i}$ and the other in $\tilde{G}_{N,\bar{F}_i}$ having the same output $y_m \dots y_{k-1}$. Thus, there are two cycles in G: (1) C_1 in X_{F_i} (starting and ending in x_m), and (2) C_2 in $X_N \cup X_{\bar{F}_i}$ (starting and ending in x'_m), such that if the system G evolves on these cycles, it will generate the same output sequence $y_m \dots y_{k-1}$. Now if the system evolves on C_1 and when the system is in x_m, the diagnoser is started, then for all j, $1 \leq j \leq k - m + 1$, z_j (the state estimate after observing y_j) will include both x_{j+m-1} and x'_{j+m-1} $(z_j \supseteq \{x_j, x'_j\})$. And more generally, for $j \geq 1$, z_j will include

x_{j^*+m-1} and x'_{j^*+m-1} with $j^* = j \bmod (k-m)$. This shows that the uncertainty in determining the mode of system (between F_i and the rest of modes N, F_j $(j \neq i)$) will never be resolved. More formally, the diagnoser states reachable through the output sequence $y_m \dots y_{k-1}$ and its repetitions will be F_i-uncertain. Therefore, the diagnoser has an F_i-indeterminate cycle and F_i is not diagnosable.

(**Sufficiency**)

Let z_1 denote: a) the state of the diagnoser after the diagnoser is initialized and the first output is read if the diagnoser is started after the failure; or b) the state of the diagnoser immediately after the failure event if the diagnoser was started some time before the occurrence of the failure F_i. One of the following cases will happen.

1. No new output is generated and the diagnoser remains in z_1 indefinitely. Then by conditions (1) and (2), $y_1 = \lambda(x)$ (with $x \in z_1$) is F_i -indicative, and thus z_1 is F_i-certain.

2. The system generates a finite number of output symbols and then stops generating new outputs. Similar to case (a), by conditions (1) and (2), the last output must be F_i-indicative resulting in F_i being detected and isolated. Note that as shown in [5], if a failure can be diagnosed, it is diagnosed in a bounded number of events.

3. The system generates an infinite output sequence. It follows from condition (3) that for $k \geq |X|^2$, z_k includes states from X_{F_i} only and therefore is F_i-certain. Thus, the failure F_i will be diagnosable. Note that this implies that the diagnoser can not have F_i-indeterminate cycles.

■

Conditions (1) and (2) ensure that in the faulty mode F_i, if the output sequence terminates, it will do so in an F_i-indicative output symbol. Condition (3) states that there should be no cycles with identical output sequences in sub-generators $\tilde{G}_{F_i}$ and $\tilde{G}_{N,\bar{F}_i}$.

Through the proof of Theorem 4.1.1, it easily shows that the conditions in Theorem 4.1.1 are equivalent to the conditions in Theorem 2.3.1. However, Theorem 4.1.1 presents a direct polynomial procedure and avoids constructing the diagnoser.

Next we examine the verification of the conditions in Theorem 4.1.1 and show that it can be done in polynomial time. This yields a test for diagnosability which, similar to those in [31] and [32], has polynomial complexity.

It can be seen that verifying conditions (1) and (2) involve finding cycles, and therefore has a time complexity of $\mathcal{O}(|X|+|\theta|)$, where θ is the set of transitions of G. Since in the nondeterministic automaton, $|\theta| \leq |\Sigma| \cdot |X|^2$, the time complexity will be $\mathcal{O}(|\Sigma| \cdot |X|^2)$.

Next we examine verifying condition (3). Suppose x in $\tilde{G}_{F_i}$ is fixed, and x' could be any state in $\tilde{G}_{N,\bar{F}_i}$ with $\lambda(x) = \lambda(x')$. Take x and x' as the initial states of $\tilde{G}_{F_i}$ and $\tilde{G}_{N,\bar{F}_i}$, respectively. We need to find output cycles that are common to $\tilde{G}_{F_i}$ and $\tilde{G}_{N,\bar{F}_i}$. One way to find common output sequences, and thus cycles, is to first convert $\tilde{G}$ to an equivalent nondeterministic generator $M\_\tilde{G}$ in which outputs changes in $\tilde{G}$ are represented as transitions. Formally, for the RTS $\tilde{G} = (X, R, Y, \lambda)$, we construct a non-deterministic generator $M\_\tilde{G} = (X \cup X', Y, \eta, x'_0)$, where $X \cup X', Y, \eta, x'_0$ are the state set, event set, transition function, and initial state. The initial state $x'_0 \notin X$

and $x'_0 \xrightarrow{\lambda(x_0)} x_0$ is the only transition out of x'_0, representing the generation of output $\lambda(x_0)$ in $\tilde{G}$. For any state $x \in X$ that is not reachable in $\tilde{G}$ from any state (i.e., $R^{-1}(x) = \emptyset$), we define a new state $x' \in X'$ and add the transition $x' \xrightarrow{\lambda(x)} x$ to the transitions of $M\_\tilde{G}$. X' consists of all such x' and x'_0. The set of transitions between the states of $M\_\tilde{G}$ consists of those mentioned above from states in X' to the corresponding states in X, and the transition of $\tilde{G}$. Thus, the transition function $\eta : (X \cup X') \times Y \rightarrow 2^{X \cup X'}$ satisfies: $\eta(x, y) = \{x' | x \Rightarrow x', \lambda(x') = y\}$ for $x \in X$.

Let $X'_{F_i} = \{x' \in X' | \exists x \in X_{F_i}, \eta(x', \lambda(x)) = \{x\}\}$ and $X'_{N,\bar{F}_i} = \{x' \in X' | \exists x \in X_{N,\bar{F}_i}, \eta(x', \lambda(x)) = \{x\}\}$. In other words, X'_{F_i} and $X'_{N,\bar{F}_i}$ are those states in X' from which there are transitions to X_{F_i} and $X_{N,\bar{F}_i}$. Now let $M\_\tilde{G}_{F_i}$ and $M\_\tilde{G}_{N,\bar{F}_i}$ be the sub-generators of $M\_\tilde{G}$ corresponding to states $X_{F_i} \cup X'_{F_i}$ and $X_{N,\bar{F}_i} \cup X'_{N,\bar{F}_i}$.

For now, we leave the initial states of $M\_\tilde{G}_{F_i}$ and $M\_\tilde{G}_{N,\bar{F}_i}$ undefined. $M\_\tilde{G}_{F_i}$ and $M\_\tilde{G}_{N,\bar{F}_i}$ represent output changes in $\tilde{G}_{F_i}$ and $\tilde{G}_{N,\bar{F}_i}$, and we use them to verify condition (3) in Theorem 4.1.1. Let $x \in X_{F_i}$ and $x' \in X_N \cup (\cup_{j \neq i} X_{F_j})$ with $\lambda(x) = \lambda(x')$, as in condition (3). Furthermore, let $M\_\tilde{G}_{F_i}(x)$ and $M\_\tilde{G}_{N,\bar{F}_i}(x')$ denote the reachable sub-generators of $M\_\tilde{G}_{F_i}$ and $M\_\tilde{G}_{N,\bar{F}_i}$ with x and x' as their respective initial states. Now, condition (3) is satisfied if and only if there are no cycles with identical output sequences in $\tilde{G}_{F_i}$ and $\tilde{G}_{N,\bar{F}_i}$, which, in turn, is equivalent to the absence of cycles in the product $M\_\tilde{G}_{F_i}(x) \times M\_\tilde{G}_{N,\bar{F}_i}(x')$.

Alternatively, one can compute $M\_\tilde{G}_{F_i} \times M\_\tilde{G}_{N,\bar{F}_i}$ and then obtain $M\_\tilde{G}_{F_i}(x) \times M\_\tilde{G}_{N,\bar{F}_i}(x')$ as the sub-generator of $M\_\tilde{G}_{F_i} \times M\_\tilde{G}_{N,\bar{F}_i}$ reachable from the state (x, x').

As mentioned in Chapter 2, $\tilde{G}$ and therefore, $M\_\tilde{G}_{F_i}$ and $M\_\tilde{G}_{N,\bar{F}_i}$ can be computed in $\mathcal{O}(|X|^2 + |X| \cdot |\theta|)$ time. $\tilde{G}$ has $|R|$ transitions, and thus $M\_\tilde{G}_{F_i} \times M\_\tilde{G}_{N,\bar{F}_i}$ can be computed in $\mathcal{O}(|X|^2 + |R|^2)$ time. $M\_\tilde{G}_{F_i}(x) \times M\_\tilde{G}_{N,\bar{F}_i}(x')$ can be computed in $\mathcal{O}(|X|^2 + |R|^2)$ (since $M\_\tilde{G}_{F_i}$ and $M\_\tilde{G}_{N,\bar{F}_i}$ each has $\mathcal{O}(|X|)$ states and $\mathcal{O}(|R|)$ transitions). $M\_\tilde{G}_{F_i} \times M\_\tilde{G}_{N,\bar{F}_i}$ will have $\mathcal{O}(|X|^2)$ states and $\mathcal{O}(|R|^2)$ transitions. To verify condition (3), we have to find the cycles of $M\_\tilde{G}_{F_i} \times M\_\tilde{G}_{N,\bar{F}_i}$ which can be done in $\mathcal{O}(|X|^2 + |R|^2)$ time. Therefore, the total complexity of verifying condition (3) is $\mathcal{O}((|X|^2 + |X| \cdot |\theta| + |R|^2)$. Since $|R| \leq |X|(|X| - 1)$ and $\theta| \leq |\Sigma| \cdot |X|^2$, we obtain $\mathcal{O}(|X|^4 + |\Sigma| \cdot |X|^3)$ as the complexity of verifying diagnosability of F_i.

Remark 4.1.1 *Compared with the tests proposed in [5], our procedure is more computationally economical because verifying diagnosability using diagnosers in the worst case has exponential time complexity in the number of system states (that is, $\mathcal{O}(2^{|X|})$). [31] and [32] propose polynomial algorithms for testing diagnosability with complexities $\mathcal{O}(|X|^4)$ and $\mathcal{O}(|X|^2)$ in the event-based framework of [4]. The test in [31] is applicable to nondeterministic systems. The test proposed here in Theorem 4.1.1 is essentially the equivalent of the test in [31] for the state-based framework of [5]. Both tests have complexities of $\mathcal{O}(|X|^4)$. The test in Theorem 4.1.1 forms the basis for a new test for fault diagnosability in timed DES which will be discussed in the next section.* ■

Theorem 4.1.1 can be easily extended to the cases involving simultaneous failures. Let us consider two simultaneous failures. Suppose we have p failure modes $F_1, \ldots, F_p$. Let F_{jk} denote simultaneous occurrence of failures F_j and F_k and let $\mathcal{F} = \{F_i | i = 1, 2, \ldots p\} \cup \{F_{jk} | j, k = 1, 2, \ldots p, j < k\}$. The order of indices in F_{jk} is not important.

For simplicity we have assumed the indices are in increasing order ($j < k$). Thus the condition map is $\mathcal{K} = \{N\} \cup \{F_i | i = 1, 2, \ldots p\} \cup \{F_{jk} | 1 \leq j \leq p, 1 \leq k \leq p, j < k\}$. Consider a failure mode F_i, let $\tilde{G}_{F_i}$ and $\tilde{G}_{N,\bar{F}_i}$ be the sub-generators corresponding to the states in $X_{F_i} \cup ((\cup_{i<j\leq P} X_{F_{ij}}) \cup (\cup_{1\leq j<i} X_{F_{ji}}))$ and the rest of system states, respectively. With the above notation, the conditions of Theorem 4.1.1 for diagnosability of F_i remain the same for this case. The theorem can be similarly extended to the cases when we have more than two simultaneous failures.

Remark 4.1.2 *Theorem 4.1.1 holds if instead of the RTS $\tilde{G}$, the reduced RTS $\tilde{G}_R$ is used.* ■

Example 4.1.1 *A system G with two single faults F_1, F_2 is shown in Figure 4.1. The output set is $Y = \{a, b, c, d\}$. The failure events are shown by dashed lines.*

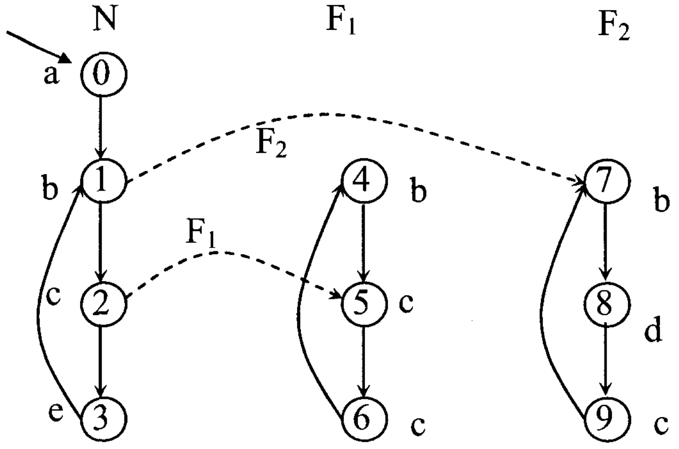


Fig. 4.1. A system G with two single failures F_1, F_2.

Suppose that we want to check the diagnosability of the fault mode F_1. Obviously, condition (1) and (2) of Theorem 4.1.1 are satisfied.

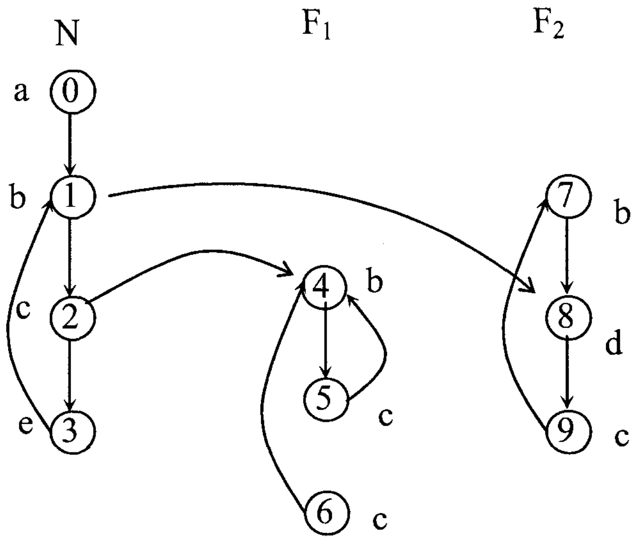


Fig. 4.2. $\tilde{G}$(Reachability Transition System (RTS) of G).

The Reachability Transition System (RTS) $\tilde{G}$ *is shown in Figure 4.2.* $X_{F_1} = \{4,5,6\}$ *and* $X_{N,\bar{F}_1} = X - X_{F_1} = \{0,1,2,3,7,8,9\}$. *Define sub-generators* $\tilde{G}_{F_1}$ *and* $\tilde{G}_{N,\bar{F}_1}$ *of* $\tilde{G}$ *corresponding to the states in* X_{F_1} *and* $X_{N,\bar{F}_1}$. *Then the output cycles in* $\tilde{G}_{F_1}$ *are: bcbc..., whereas the output cycle in* $\tilde{G}_{N,\bar{F}_1}$ *is bcebce... and bdcbdc.... Therefore, there is no common output cycle in these two sub-generators and condition(3) of Theorem 4.1.1 holds.*

A more efficient way of verifying condition (3) of Theorem 4.1.1 is as follows. We can convert $\tilde{G}$ *into an equivalent generator* $M\_\tilde{G}$ *representing output changes as events (Figure 4.3). The sub-generators* $M\_\tilde{G}_{F_1}$ *and* $M\_\tilde{G}_{N,\bar{F}_1}$ *are also shown in Figure 4.3. Note that in the resulting finite state machine, we have added an extra state* $0'$ *before initial state* 0 *because we should convert the first output signal to event. State* $6'$ *has been added for a similar reason.*

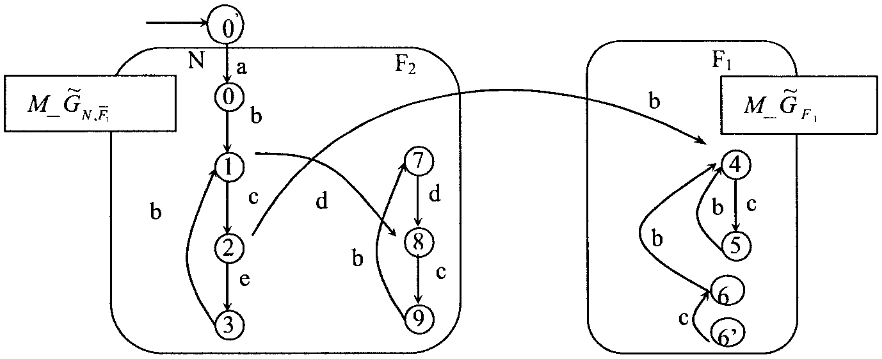


Fig. 4.3. $M_{\tilde{G}}, M_{\tilde{G}_{F_1}}$ and $M_{\tilde{G}_{N,\bar{F}_1}}$.

Let us verify condition (3) for states $x = 4$, and $x' = 1$. As discussed before, we have to form the parallel product of two M_generators $M\_\tilde{G}_{F_1}(4)$ and $M\_\tilde{G}_{N,\bar{F}_1}(1)$. If there is at least one cycle in the resulting system, condition (3) is not satisfied. For this specific example, the result which is shown in Figure 4.4 contains no cycles. We can repeat this procedure and verify that condition (3) is satisfied. Since conditions (1) and (2) are also satisfied, F_1 is diagnosable.

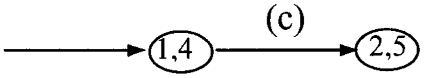


Fig. 4.4. Parallel product of $M_{\tilde{G}_{F_1}}(4)$ and $M_{\tilde{G}_{N,\bar{F}_1}}(1)$.

■

4.2 Testing Diagnosability In Timed DES

In this section, we discuss a polynomial test for time-diagnosability. Let us consider a non-deterministic **Timed DES (TDES)** [11] $G = (X, \Sigma \cup \{\tau\}, \delta, x_0, Y, \lambda)$

where $X, \Sigma \cup \{\tau\}, Y$ are the finite state, event and output sets, x_0 is the initial state; $\delta : X \times \Sigma \cup \{\tau\} \rightarrow 2^X$ is the transition function and $\lambda : X \rightarrow Y$ is the output map [1]. It is assumed that the TDES is **activity-loop-free**, that is, it does not contain a cycle of non-tick events [11]. Assume p failure modes, $F_1, \ldots, F_p$, and $X = X_N \dot{\cup} (\dot{\cup}_{i=1}^{p} X_{F_i})$.

While it is possible to adapt the procedures for verifying diagnosability in untimed DES, such as those given in [31,32] or the procedure in Theorem 4.1.1 in the previous section, here however, we develop an alternative test for diagnosability specifically for timed DES based on timed RTS $\tilde{G}$. Timed RTS may be considered as a modified version of the TDES in which the timing information represented in timed transition graph of the TDES has been gathered and compiled in the transition-time function $T(x, x'')$ (The transition-time function was defined in Section 2.1.2). In [23], it is shown that using timed RTS to construct the diagnoser may significantly reduce the size of the diagnoser. In this dissertation, we show that using timed RTS may also significantly reduce the computations of verifying diagnosability.

The analysis of computational complexity is included in this chapter. We show that the proposed test becomes particularly efficient (compared with existing methods for untimed DES) when the transition-time sets are either bounded or can be represented as the union of a bounded number of intervals.

4.2.1 Diagnosability Test

We start by defining time sequences associated with output sequences.

[1] Without loss of generality, we assume the event "tick" in timed DES does not change the system output. In general, if the assumption fails, we replace the tick transition with two consecutive transitions, a new state and event. For example, if $x_1 \xrightarrow{\tau} x_2$ and $\lambda(x_1) \neq \lambda(x_2)$, we introduce an event σ and a state x_2' such as $x_1 \xrightarrow{\tau} x_2'$, $\lambda(x_1) = \lambda(x_2')$, $x_2' \xrightarrow{\sigma} x_2$.

Definition 4.2.1 *Consider $x \in X$ and an output sequence $s = y_1 \dots y_n$ $(n \geq 2)$, with $s \in L_0(\tilde{G}, x)$. The set of **time sequences** $TS(s, x, \tilde{G})$ is defined according to*

$$TS(s,x,\tilde{G}) := \{\hat{t}_1\hat{t}_2 \dots \hat{t}_{n-1} \mid \exists x_i \in X \ (1 \leq i \leq n) : x_1 = x, \ x_i \Rightarrow x_{i+1},$$

$$\hat{t}_i \in T(x_i, x_{i+1}) \ (1 \leq i \leq n-1), \ \textit{and}\, y_i = \lambda(x_i) \ (1 \leq i \leq n)\} \quad \blacksquare$$

Let G_{F_i} denote the subgenerator of G consisting of the states in X_{F_i} only. Similarly, let G_N and $G_{N,\bar{F}_i}$ be the subgenerators of G corresponding to the states in X_N and $X_N \cup X_{\bar{F}_i}$ (The initial states of G_{F_i}, G_N and $G_{N,\bar{F}_i}$ are left undefined.) The subgenerators of $\tilde{G}$, $\tilde{G}_{F_i}$, $\tilde{G}_N$ and $\tilde{G}_{N,\bar{F}_i}$, are defined similarly. Output languages and time sequences for G_N, G_{F_i}, $G_{N,\bar{F}_i}$, $\tilde{G}_N$, $\tilde{G}_{F_i}$ and $\tilde{G}_{N,\bar{F}_i}$ are defined similar to those of G.

Define $Cyc(X_N)$, $Cyc(X_{F_i})$ and $Cyc(X_{\bar{F}_i})$ as the cycles (including the self-loops) of constant output in G_N, G_{F_i} and $G_{\bar{F}_i}$ and

$$\lambda(Cyc(X_N)) := \{y \in Y \mid \exists x : x \text{ is a state in a cycle in}$$

$$Cyc(X_N) \text{ and } y = \lambda(x)\}$$

$\lambda(Cyc(X_{F_i}))$ and $\lambda(Cyc(X_{\bar{F}_i}))$ are also defined similarly.

Furthermore, let

$$T_{max} := \max\nolimits_{x,\, x'' \in X}\{\max T(x,x'') \mid T(x,x'') \neq \emptyset \text{ and } \sup T(x,x'') < \infty\}$$

T_{max} is the longest transition-time an output change can take between any two states x, x'' with a *finite* transition-time set $T(x, x'')$.

Remark 4.2.1 *Consider a state $x \in X$. Suppose there are no cycles with constant output $y = \lambda(x)$ that are reachable from x using a path with constant output $y = \lambda(x)$. Then (i) for some $x' \in X$, $x \Rightarrow x'$ and (ii) for all $x'' \in X$ with $x \Rightarrow x''$, we have* $\sup T(x, x'') < \infty$. *Now if at some point, the plant G is in state x, then the next output symbol will be generated before $T_{max} + 1$ ticks of clock. The above interpretation of T_{max} will be used in the proof of Theorem 4.2.1.* ■

Theorem 4.2.1 provides necessary and sufficient conditions for diagnosability. These conditions form the basis of the diagnosability test (for timed DES) presented in this thesis.

Theorem 4.2.1 *Assume single-failure scenario and $z_0 = X$. A permanent failure F_i is diagnosable if and only if we have the following:*

1. $\lambda(Cyc(X_{F_i})) \cap (\lambda(Cyc(X_N)) \cup \lambda(Cyc(X_{\bar{F}_i}))) = \emptyset$

2. *If $x \in X_{F_i}$, and $x' \in X_N \cup (\cup_{j \neq i} X_{F_j})$ with $\lambda(x) = \lambda(x')$, then for any $s \in L_o(\tilde{G}_{N,\bar{F}_i}, x') \cap L_o(\tilde{G}_{F_i}, x)$ with $|s| \geq |X|^2$, we have*

$$TS(s, x', \tilde{G}_{N,\bar{F}_i}) \cap TS(s, x, \tilde{G}_{F_i}) = \emptyset.$$

Proof

Proof (Necessity)

If condition (1) does not hold, then there exist two cycles, one in $Cyc(X_{F_i})$ (say C) and another in $Cyc(X_N) \cup Cyc(X_{\bar{F}_i})$ (say C') having the same output. If the fault diagnosis system is initialized when the system evolves on C, then no new output will be generated and the system state estimate will include states in C and C' and as a result, the fault F_i will be undiagnosable. This shows the necessity of condition (1).

Suppose condition (1) holds. If condition (2) does not hold, then for some $x \in X_{F_i}$ and $x' \in X_N \cup (\cup_{j \neq i} X_{F_j})$ with $\lambda(x) = \lambda(x')$, there exists $s = y_1 y_2 \ldots y_q$ ($q \geq |X|^2$), with $s \in L_o(\tilde{G}_{N,\bar{F}_i}, x') \cap L_o(\tilde{G}_{F_i}, x)$ such that $TS(s, x', \tilde{G}_{N,\bar{F}_i}) \cap TS(s, x, \tilde{G}_{F_i}) \neq \emptyset$. Thus, there exist x_i, x'_i ($i = 1, \ldots, q$) such that $x_i \in X_{F_i}$, $x'_i \in X_N \cup X_{\bar{F}_i}$, $x_1 = x$, $x'_1 = x'$, $\lambda(x_i) = \lambda(x'_i) = y_i$ ($1 \leq i \leq q$), and $x_i \Rightarrow x_{i+1}$, $x'_i \Rightarrow x'_{i+1}$, $T(x_i, x_{i+1}) \cap T(x'_i, x'_{i+1}) \neq \emptyset$ ($1 \leq i \leq q-1$). The corresponding state transitions in $\tilde{G}_{F_i}$ and $\tilde{G}_{N,\bar{F}_i}$ can be represented as follows:

$$\begin{array}{llll} \tilde{G}_{F_i}: & x = x_1 \Rightarrow & x_2 \Rightarrow \ldots & x_q \\ \tilde{G}_{N,\bar{F}_i}: & x' = x'_1 \Rightarrow & x'_2 \Rightarrow \ldots & x'_q \\ \textit{Output}: & (y_1) & (y_2) & (y_q) \end{array}$$

Since $|X_{F_i}| \times |X_N \cup X_{\bar{F}_i}| < |X|^2$, there exist m, k, with $1 \leq m < k \leq |X|^2$, such that $x_m = x_k$ and $x'_m = x'_k$. Therefore, there exist two cycles, one in $\tilde{G}_{F_i}$ and the other in $\tilde{G}_{N,\bar{F}_i}$ having the same output $y_m \ldots y_{k-1}$ and the same time sequences $\hat{t}_m, \ldots \hat{t}_{k-1}$ ($\hat{t}_j \in T(x_j, x_{j+1}) \cap T(x'_j, x'_{j+1})$, $m \leq j \leq k-1$). Thus, there are two cycles in G: (1) $\mathcal{C}_1$ in X_{F_i} (starting and ending in x_m), and (2) $\mathcal{C}_2$ in $X_N \cup X_{\bar{F}_i}$ (starting and ending in x'_m) such that on these cycles, the system can generate the same output sequence $y_m \ldots y_{k-1}$ with the same time sequence $\hat{t}_m, \ldots \hat{t}_{k-1}$. Now if when the system is in x_m, the diagnoser is started and after this, the system remains on $\mathcal{C}_1$, then for all $1 \leq j \leq k - m + 1$, z_j (the state estimate after observing y_j) will include both x_{j+m-1} and x'_{j+m-1}. And more generally, for $j \geq 1$, z_j will include x_{j^*+m-1} and x'_{j^*+m-1} with $j^* = j \bmod (k-m)$. This shows that the uncertainty between F_i and the rest of modes N and F_j's ($j \neq i$) will never be resolved. In other words, the states of the diagnoser that are reachable through the output sequence $y_m \ldots y_{k-1}$ and its repetitions will be F_i-uncertain. Therefore, F_i will not be diagnosable.

(Sufficiency)

Let z_1 denote: (i) the state of the diagnoser after the diagnoser is initialized and the first output is read if the diagnoser is started after the failure F_i, or (ii) the state of the diagnoser immediately after the failure event F_i if the diagnoser is started some time before the occurrence of the failure F_i. Let $y_1y_2y_3\ldots$.be the output sequence generated by the system. As for the output sequence, one of the following cases happens.

a) No new output is generated and the output remains $y_1 = \lambda(z_1)$. If z_1 is F_i -certain, then the diagnoser state estimate will remain F_i -certain for the future. Suppose z_1 is not F_i -certain. Since no new output is generated, then G has a cycle in X_{F_i} with output y_1. By condition (1), there is no cycle in $X_N \cup (\cup_{j \neq i} X_{F_j})$ which generates the same output. Therefore, by Remark 4.2.1, after the maximum of $T_{max} + 1$ ticks, the state estimate provided by the diagnoser, say z', will not contain states from $X_N \cup (\cup_{j \neq i} X_{F_j})$ and will be F_i -certain; in other words, $\kappa(z') = \{F_i\}$.

b) The system generates an infinite number of output symbols $y_1y_2\ldots$. If z_1 is F_i-certain, then future diagnoser state estimates will be F_i-certain and the fault is diagnosed. Suppose z_1 is not F_i-certain. Let z_2' be the state estimate after y_2 is generated. If z_2' is F_i- certain, then the fault is diagnosed. If z_2' is not F_i-certain, then there exists $x \in X_{F_i}$ and $x' \in X_N \cup (\cup_{j \neq i} X_{F_j})$ with $x,\ x' \in z_2'$ (Note that z_1 does not necessarily include a state from X_{F_i}, even if z_1 is not F_i- certain.) Let $s = y_2 \ldots y_k$ and z_k' be the state estimate immediately after y_k is generated with $k = |X|^2 + 1$. Then, we have the following cases:

1. $s \notin L_o(\tilde{G}_{N,\bar{F}_i}, x') \cap L_o(\tilde{G}_{F_i}, x)$. It follows that z'_k will not include any state $x' \in X_N \cup (\cup_{j\neq i} X_{F_j})$, and therefore it is F_i-certain; i.e., the failure F_i will be diagnosed.

2. $s \in L_o(\tilde{G}_{N,\bar{F}_i}, x') \cap L_o(\tilde{G}_{F_i}, x)$. Then it follows from the condition (2), that the time sequence sets $TS(s, x', \tilde{G}_{N,\bar{F}_i})$ and $TS(s, x, \tilde{G}_{F_i})$ have no common element. That is, z'_k will not include any $x' \in X_N \cup (\cup_{j\neq i} X_{F_j})$ and therefore it is F_i-certain, and the failure F_i will be diagnosed.

c) The system generates a finite number of output symbols $y_1 y_2 \dots y_{n_i}$ $(n_i \geq 2)$ and then stops generating new outputs. Let $t_{k,k+1}$ be the number of ticks generated between y_k and y_{k+1} $(1 \leq k \leq n_i - 1)$.

1. If $t_{k,k+1} \leq T_{max}$, $(1 \leq k \leq n_i - 1)$, then similar to case a), the diagnoser state estimate generated $T_{max} + 1$ ticks after y_{n_i} is generated will be F_i-certain, and the diagnoser state estimate will remain F_i-certain afterwards. Therefore, after a maximum of $n_i T_{max} + 1$ ticks, the state estimate provided by the diagnoser will be F_i-certain. This shows F_i can be diagnosed in finite time. As shown in [23], if a fault can be diagnosed, it will be diagnosed in a bound number of ticks.

2. Assume for some $1 \leq k \leq n_i - 1$, $t_{k,k+1} > T_{max}$. Let k_0 be the smallest such k. Let z'_{k_0} be the diagnoser state estimate after $y_{k_0}$is generated. If z'_{k_0} is F_i-certain, the fault will be diagnosable. Suppose z'_{k_0} is not F_i-certain. Then since $t_{k,\,k+1} > T_{max}$, X_{F_i} contains a cycle with output $\lambda(z'_{k_0})$. Then by condition (1), $X_N \cup (\cup_{j\neq i} X_{F_j})$ can not have a cycle with constant output $\lambda(z'_{k_0})$. Thus, after

$T_{max} + 1$ ticks, the diagnoser state estimate will be F_i-certain. Therefore, the failure F_i will be diagnosed in at most $k_0 T_{max} + 1$ ticks.

■

Condition (1) in Theorem 4.2.1 means that there are no two constant-output cycles in X_{F_i} and $X_N \cup (\cup_{j \neq i} X_{F_j})$ with the same output. Condition (2) states that there are no cycles in $\tilde{G}_{F_i}$ and $\tilde{G}_{N,\bar{F}_i}$ having the same output sequence and a common time sequence. Next we discuss how conditions (1) and (2) can be verified.

Verifying condition (1) involves finding cycles in G with constant output, in X_{F_i} and $X_N \cup (\cup_{j \neq i} X_{F_j})$. For condition (2), we need to find output sequences (with their corresponding time sequences) that are common in $\tilde{G}_{F_i}$ and $\tilde{G}_{N,\bar{F}_i}$. One way to find common output sequences, and thus cycles, is first to convert timed RTS $\tilde{G} = (X, Y, T, \lambda)$ to a nondeterministic generator $M\_\tilde{G}$ in which output changes in $\tilde{G}$ are represented as transitions. Formally, for the timed RTS $\tilde{G}$, we construct a nondeterministic generator $M\_\tilde{G} = (X \cup X', Y, \eta, x_0', T)$, where $X \cup X'$, Y, η, x_0' and T are the state set, event set, transition function, initial state and transition-time function. The initial state $x_0' \notin X$ and (by definition) $x_0' \xrightarrow{\lambda(x_0)} x_0$ is the only transition out of x_0', representing the generation of output $\lambda(x_0)$ in $\tilde{G}$. For any state $x \in X$ that is not reachable in $\tilde{G}$ from any state, we define a new state $x' \in X'$ and add the transition $x' \xrightarrow{\lambda(x)} x$ to the transitions of $M\_\tilde{G}$. X' consists of all such x' and the initial state x_0'. The set of transitions between the states of $M\_\tilde{G}$ consists of those mentioned above from the states in X' to the corresponding states in X, and the transitions of $\tilde{G}$. Thus, the transition function $\eta : (X \cup X') \times Y \to 2^{X \cup X'}$ satisfies: $\eta(x, y) = \{x' \mid x \Rightarrow x',\ \lambda(x') = y\}$ for $x \in X$. The definition of T (as a function on

$X \times X$) is extended to a function on $(X \cup X') \times (X \cup X')$ as follows: $T(x, x')$ is the same as before if $x,\ x' \in X$, and if $x \in X'$ or $x' \in X'$, then $T(x, x') = \{0\}$.

Let $X'_{F_i} = \{x' \in X' \mid \exists x \in X_{F_i},\ \eta(x', \lambda(x)) = \{x\}\}$ and $X'_{N,\bar{F}_i} = \{x' \in X' \mid \exists x \in X_{N,\bar{F}_i}, \eta(x', \lambda(x)) = \{x\}\}$. In other words, X'_{F_i} and $X'_{N,\bar{F}_i}$ are those states in X' from which there are transitions to X_{F_i} and $X_{N,\bar{F}_i}$. Now let $M\_\tilde{G}_{F_i}$ and $M\_\tilde{G}_{N,\bar{F}_i}$ be the subgenerators of $M\_\tilde{G}$ corresponding to states $X_{F_i} \cup X'_{F_i}$ and $X_{N,\bar{F}_i} \cup X'_{N,\bar{F}_i}$.

For now, we leave the initial states of $M\_\tilde{G}_{F_i}$ and $M\_\tilde{G}_{N,\bar{F}_i}$ undefined. $M\_\tilde{G}_{F_i}$ and $M\_\tilde{G}_{N,\bar{F}_i}$ represent output changes in $\tilde{G}_{F_i}$ and $\tilde{G}_{N,\bar{F}_i}$ in the form of transitions, and we use them to verify condition (2) in Theorem 4.2.1. Let $x \in X_{F_i}$ and $x' \in X_N \cup (\cup_{j \neq i} X_{F_j})$ with $\lambda(x) = \lambda(x')$, as in condition (2). Furthermore, let $M\_\tilde{G}_{F_i}(x)$ and $M\_\tilde{G}_{N,\bar{F}_i}(x')$ denote the reachable subgenerators of $M\_\tilde{G}_{F_i}$ and $M\_\tilde{G}_{N,\bar{F}_i}$ with x and x' as the initial states. Now, condition (2) is satisfied if and only if there are no cycles with identical output sequences having a common timing sequence in $\tilde{G}_{F_i}$ and $\tilde{G}_{N,\bar{F}_i}$, which, in turn, is equivalent to the absence of cycles in the product $M\_\tilde{G}_{F_i}(x) \times M\_\tilde{G}_{N,\bar{F}_i}(x')$. Note that in the computation of the product of the generators, we have to take into account the timing sequences of the output sequences as well. In the following, we define the **timed product** of generators that takes the timing of transitions into account in the product operation.

Definition 4.2.2 *Consider two timed finite-state generators $M\_\tilde{G}_1 = (X_1, Y,\ \eta_1,\ x'_{0,1},\ T_1)$ and $M\_\tilde{G}_2 = (X_2, Y, \eta_2, x'_{0,2}, T_2)$. Define the* ***timed product*** *of these two finite-state generators $M\_\tilde{G}_1 \times M\_\tilde{G}_2$ as the reachable sub-generator of $M\_\tilde{G} =$*

(X, Y, η, x'_0, T), where $X = X_1 \times X_2$, $\eta : X \times Y \rightarrow 2^X$, $x'_0 = (x'_{0,1}, x'_{0,2})$ and T is the transition-time function. The functions η and T are given by:

$$\begin{aligned} \eta((x_1,x_2),y) =& \{(x'_1,x'_2) \mid x'_1 \in \eta_1(x_1,y),\ x'_2 \in \eta_2(x_2,y) \ \mathit{and} \\ & T_1(x_1,x'_1) \cap T_2(x_2,x'_2) \neq \emptyset\} \\ T((x_1,x_2),\ (x'_1,x'_2)) =& T_1(x_1,x'_1) \cap T_2(x_2,x'_2) \end{aligned}$$

■

We observe that the above procedure of constructing timed products can be used to look for common output sequences having common timing sequences in two generators. Suppose $x \in X_{F_i}$ and $x' \in X_N \cup (\cup_{j \neq i} X_{F_j})$ with $\lambda(x) = \lambda(x')$. Let $M\_\tilde{G}_{F_i}(x)$ and $M\_\tilde{G}_{N,\bar{F}_i}(x')$ denote the subgenerators of $M\_\tilde{G}_{F_i}$ and $M\_\tilde{G}_{N,\bar{F}_i}$ reachable from x and x'. If the timed product of $M\_\tilde{G}_{F_i}(x) \times M\_\tilde{G}_{N,\bar{F}_i}(x')$ does not contain any cycles, then condition (2) of Theorem 4.2.1 is satisfied, and vice versa. To verify condition (2), this procedure has to be repeated for all $x \in X_{F_i}$ and $x' \in X_N \cup (\cup_{j \neq i} X_{F_j})$ that have the same output. The following example demonstrates the procedure.

Example 4.2.1 *Figure 4.5 shows a timed DES G where $\Sigma = \{\alpha, \beta, f_1, f_2\}$, $Y = \{a, b\}$, $F = \{F_1, F_2\}$ and $X = X_N \cup X_{F_1} \cup X_{F_2}$ with $X_N = \{0, 1, 2, 3, 4,$ 5.1, 5.2, 5.3, 5.4, 6$\}$, $X_{F_1} = \{7\}$, $X_{F_2} = \{8, 9, 10, 11.1, 11.2, 11.3, 11.4\}$.*

We apply Theorem 4.2.1 to see whether F_2 is diagnosable. First, there is a self-loop in state 7 and $\lambda(Cyc(X_{F_1})) = \{a\}$. In contrast, there is no cycle with constant output in X_N and X_{F_2}. Therefore, $Cyc(X_N) = \emptyset$ and $Cyc(X_{F_2}) = \emptyset$, and condition (1) is satisfied. To verify condition (2), we examine the output sequences and their time sequences. For example, states $9 \in X_{F_2}$ and $0 \in X_N$ have the same output 'a'.

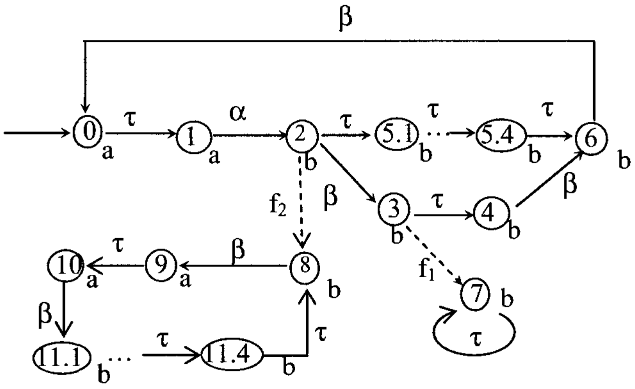


Fig. 4.5. A timed DES example.

The output sequence starting from these states is 'ababab...'; therefore, $L_o(\tilde{G}_{N,\bar{F}_2},0)\cap L_o(\tilde{G}_{F_2},9) = a(ba)^*$. *We can see that the time sequences for* $s=(ab)^n$ $(n \geq |X|^2/2 = 18^2/2$ *are* $TS(s,0,\tilde{G}_{N,\bar{F}_2}) = [\tau_1(\tau_5+\tau_1)\tau_0]^{n-1}\tau_1$ *and* $TS(s,9,\tilde{G}_{F_2}) = \tau_1(\tau_4\tau_1)^{n-1}$ *(τ_i represents a duration of i ticks.) Therefore,* $TS(s,0,\tilde{G}_{N,\bar{F}_2}) \cap TS(s,9,\tilde{G}_{F_2}) = \emptyset$. *It means that their time sequences have no common element. Similarly, for output sequences* $s = a(ba)^n$, $T(s,0,\tilde{G}_{N,\bar{F}_2}) \cap T(s,9,\tilde{G}_{F_2}) = \emptyset$. *Therefore, condition (2) is satisfied for* $x = 9$ *and* $x' = 0$. *Condition (2) can be verified for all* $x \in X_{F_2}$ *and* $x' \in X_N \cup X_{F_1}$ *. Therefore,* F_2 *is diagnosable. Similarly, we can verify that* F_1 *is diagnosable.*

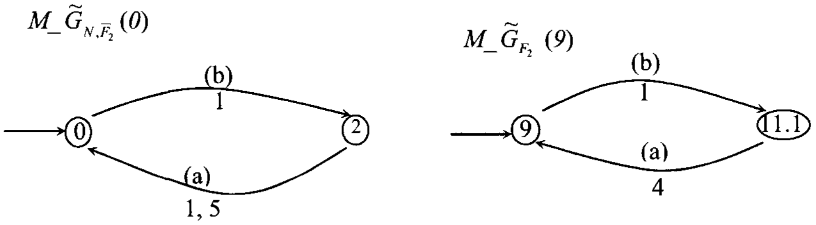


Fig. 4.6. Example 4.2.1: $M_{\tilde{G}_{N,\bar{F}_2}}(0)$ and $M_{\tilde{G}_{F_2}(9)}$.

Table 4.1
Timed RTS of Example 4.2.1

State	Output	Output-adjacent state(time)	State	Output	Output-adjacent state(time)
0	a	2(1)	6	(b)	0(0)
1	a	2(0)	7	(a)	—
2	b	0({ 1, 5})/9(0)	8	(b)	9(0)
3	b	0(1)	9	(a)	11.1(1)
4	b	0(0)	10	(a)	11.1(0)
5.i	b	0(5-i)	11.i	(b)	9(5-i)

Alternatively, to verify condition (2) of Theorem 4.2.1, we can follow the procedure described before the example which is more suitable for computer implementation. The timed RTS $\tilde{G}$ is shown in Table 4.1. To verify condition (2) for $x = 9$ and $x' = 0$, we convert $\tilde{G}$ into $M\_\tilde{G}$ and then obtain subgenerators $M\_\tilde{G}_{N,\bar{F}_2}(0)$ and $M\_\tilde{G}_{F_2}(9)$ (Fig. 4.6), which take 0 and 9 as their respective initial states. Then we form the timed product of $M\_\tilde{G}_{N,\bar{F}_2}(0) \times M\_\tilde{G}_{F_2}(9)$ as discussed in Definition 4.2.2.

The result is shown in Fig. 4.7. We observe that there are no cycles in the product generator, and thus, condition (2) is satisfied for $x = 9$ and $x' = 0$. This procedure has to be repeated for all $x \in X_{F_2}$ and $x' \in X_N \cup X_{F_1}$ having the same output. ■

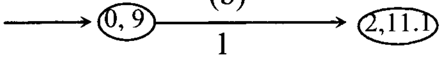


Fig. 4.7. Example 4.2.1: Timed product of $M_{\tilde{G}_{N,\bar{F}_2}(0)}$ and $M_{\tilde{G}_{F_2}(9)}$.

Theorem 4.2.1 can be easily extended to the cases involving simultaneous failures. Let us consider two simultaneous failures. Suppose we have p failure modes F_1, …,F_p. Let F_{jk} denote simultaneous occurrence of failures F_j and F_k and let $\mathcal{F} = \{F_i \mid i = 1,2,\ldots p\} \cup \{F_{jk} \mid j,\ k = 1,2,\ldots p,\ j < k\}$. The order of indices in F_{jk} is not important. For simplicity we have assumed the indices are in increasing order ($j < k$). Thus the condition map is $\mathcal{K} = \{N\} \cup \{F_i \mid i = 1,2,\ldots p\} \cup \{F_{jk} \mid 1 \leq j \leq p,\ 1 \leq j < k \leq p\}$. Consider a failure mode F_i. Let $\tilde{G}_{F_i}$ and $\tilde{G}_{N,\bar{F}_i}$ be the subgenerators corresponding to the states in $X_{F_i} \cup ((\cup_{i<j\leq p} X_{F_{ij}}) \cup (\cup_{1\leq j<i} X_{F_{ji}}))$ and the rest of system states, respectively. With the above notation, the conditions of Theorem 4.2.1 for diagnosability of F_i remain the same for this case. The theorem can be similarly extended to the cases when we have more than two simultaneous failures.

Remark 4.2.2 *In order to verify diagnosability in timed DES, we can also use the tests developed for untimed DES. [31] and [32] provide diagnosability tests in the event-based framework of [4]. Theorem 4.1.1 proposes a test in the state-based framework of [5].*

In order to use the test in Theorem 4.1.1, first the information about the clock tick (an observable event) must be transferred and included in the output map. [2] *Let G' be the resulting TDES and Y' the new (extended) output set. Furthermore, let $\tilde{G}'$ denote the RTS corresponding to G'. To verify the diagnosability of a failure*

[2] For this, we can replace the TDES G with another TDES $G' = (X', \Sigma' \cup \{\tau\}, \delta', x_0, Y', \lambda')$. TDES G' is obtained by replacing every τ transition $x_1 \xrightarrow{\tau} x_2$ in G with transitions $x_1 \xrightarrow{\tau} x_1' \xrightarrow{\sigma} x_2$ in G', where x_1' and σ are new state and event. In G', $\Sigma' = \Sigma \cup \{\sigma\}$ and $Y' = Y \times \{0,1\}$. The output map $\lambda' : X' \to Y'$ is defined according to: $\lambda'(x) = (\lambda(x), 0)$ if $x \in X$, and $\lambda'(x) = (\lambda(x), 1)$ if $x \in X' - X$. Thus in the two back-to-back transitions $x_1 \xrightarrow{\tau} x_1' \xrightarrow{\sigma} x_2$, the output changes from $(\lambda(x_1), 0)$ to $(\lambda(x_1'), 1)$ and to $(\lambda(x_2), 0)$.

mode F_i, the diagnosability test for untimed DES in Theorem 4.1.1 can be applied to G'. This test looks for common output cycles in $\tilde{G}'_{F_i}$ and $\tilde{G}'_{N,\bar{F}_i}$ (the subgenerators corresponding to states X'_{F_i} and $X'_N \cup (\cup_{j\neq i} X'_{F_j})$). To this end, for any $x \in X'_{F_i}$ and $x' \in X'_N \cup (\cup_{j\neq i} X'_{F_j})$ with $\lambda'(x) = \lambda'(x')$, M_generators, say, $M\_\tilde{G}'_{F_i}(x)$ and $M\_\tilde{G}'_{N,\bar{F}_i}(x')$ are constructed in which output changes in $\tilde{G}'_{F_i}$ and $\tilde{G}'_{N,\bar{F}_i}$ (starting from x and x') are represented as transitions (For instance Fig. 4.8 shows $M\_\tilde{G}'_{F_2}(9)$ and $M\_\tilde{G}'_{N,\bar{F}_2}(0)$ for Example 4.2.1.) Next, $M\_\tilde{G}'_{F_i}(x) \times M\_\tilde{G}'_{N,\bar{F}_i}(x')$ is constructed and examined for the existence of cycles.

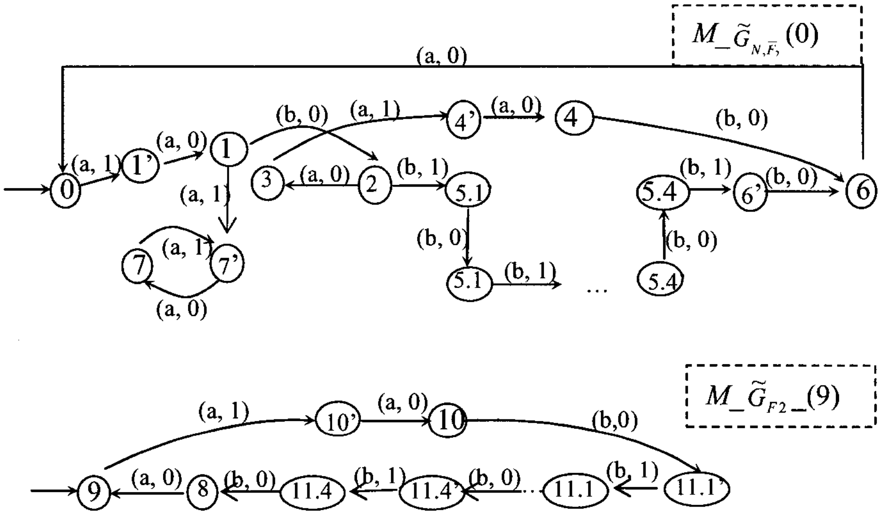


Fig. 4.8. Example 4.2.1: M_generators with tick treated as an extra output signal.

The main difference between the above method (based on tests for untimed DES) and the one proposed in this section is that in the method proposed here, the information about the timing of events in the TDES is gathered and summarized in the transition-time function $T(x, x')$ and is subsequently used in diagnosability test, specif-

ically, in the timed product in Definition 4.2.2. In the following section, we discuss cases in which the transition-time sets can be computed efficiently and the intersection operation of time product (Definition 4.2.2) can be performed easily by comparing integers. In the tests based on untimed models, however, the timing information is retained in the transition graphs of the DES model and the timing properties have to be investigated using operations on graphs.

In the method proposed here, the timing information is gathered and summarized in the M_generators and thus $M\_\tilde{G}_{F_i}(x)$ and $M\_\tilde{G}_{N,\bar{F}_i}(x')$, and $M\_\tilde{G}_{F_i}(x) \times M\_\tilde{G}_{N,\bar{F}_i}(x')$ have typically fewer states than $M\_\tilde{G}'_{F_i}(x)$, $M\_\tilde{G}'_{N,\bar{F}_i}(x')$ and $M\_\tilde{G}'_{F_i}(x) \times M\_\tilde{G}'_{N,\bar{F}_i}(x')$ in the untimed methods.

Table 4.2
State size of DES (Example 4.2.1) required for testing diagnosability

	Proposed Method for Timed DES	Adopting Untimed Methods
$M\_\tilde{G}'_{N,\bar{F}_2}(0)$	2 states	$2m_1 + 8$ states
$M\_\tilde{G}_{F_2}(9)$	2 states	$2m_2 + 4$ states

For instance, in Example 4.2.1, if in the TDES G (Figure 4.5), the transition from state 2 to state 6 takes m_1 ticks, and the transition from state 11.1 to state 8 takes m_2 ticks, then $M\_\tilde{G}_{N,\bar{F}_2}(0)$ and $M\_\tilde{G}_{F_2}(9)$ in Fig.4.6 (constructed based on the timed RTS) have 2 and 2 states whereas the corresponding M_generators in the untimed approach, $M\_\tilde{G}'_{N,\bar{F}_2}(0)$ and $M\_\tilde{G}'_{F_2}(9)$, will have $2m_1$+8 and $2m_2$+4 states (Table 4.2). Therefore, we observe that when the time bounds of events are large, the

M_generators required for verifying diagnosability using Theorem 4.2.1 have significantly fewer states than those required using an untimed method. ■

In summary, as shown in Example 4.2.1 and Remark 4.2.2, if the timed RTS has been already computed, say for diagnostic design following [5], then the test proposed in Theorem 4.2.1 can be performed with significantly fewer computations than tests developed for untimed DES (such as Theorem 4.1.1).

In the next section, we discuss three cases in which the timed RTS can be computed efficiently.

4.2.2 Computation of timed RTS and Analysis of Computational Complexity

In this section, we discuss the issue of computational complexity and show that the diagnosability test proposed in the previous section can be performed in polynomial time. The test becomes particularly efficient (compared with existing methods for untimed DES) when the transition-time sets are either bounded or can be represented as the union of a bounded number of intervals. We will provide conditions under which the transition-time sets have the above representation.

Condition (1) in Theorem 4.2.1 which involves finding cycles can be verified in $\mathcal{O}(|X|+|\theta|)$, where θ is the set of transitions of G[3]. For the nondeterministic generator G, $|\theta| < |\Sigma \cup \{\tau\}| \cdot |X|^2$ and thus we shall assume $\mathcal{O}(|\theta|) = \mathcal{O}(|\Sigma| \cdot |X|^2)$. The computation of timed RTS $\tilde{G}$ involves finding transition-times $T(x, x'')$. Suppose for

[3] The set of states in G that belong to a cycle can be obtained using a breadth-first search (BFS). A BFS takes $\mathcal{O}(|X| + |\theta|)$ time.

$x \in X$, the complexity of finding $T(x, x'')$ for all $x'' \in X$, $x'' \neq x$, is $\mathcal{O}(T_1(|X|, |\Sigma|))$. Later in this section, we discuss three cases in which T_1 is a polynomial function. For each $x \in X$, the set of output-adjacent states can be calculated using a breadth-first search in $\mathcal{O}(|X| + |\theta|)$. The corresponding transition-time sets can be obtained in $\mathcal{O}(T_1)$. Thus, the entire timed RTS, and as a result, $M\_\tilde{G}_{F_i}$ and $M\_\tilde{G}_{N,\bar{F}_i}$, can be computed in $\mathcal{O}(|X|(|X| + |\theta| + T_1))$. Let R denote the set of transitions of $\tilde{G}$, and suppose the transition-time intersection operation in Definition 4.2.2 can be done in $\mathcal{O}(T_2(|X|))$ time. With this notation, $M\_\tilde{G}_{F_i} \times M\_\tilde{G}_{N,\bar{F}_i}$ can be computed in $\mathcal{O}(|X|^2 + |R|^2 + |R|^2 \cdot T_2)$ time since $M\_\tilde{G}_{F_i}$ and $M\_\tilde{G}_{N,\bar{F}_i}$ have each $\mathcal{O}(|X|)$ states and $\mathcal{O}(|R|)$ transitions. [4] $M\_\tilde{G}_{F_i} \times M\_\tilde{G}_{N,\bar{F}_i}$ will have $\mathcal{O}(|X|^2)$ states and $\mathcal{O}(|R|^2)$ transitions. To verify condition (2) in Theorem 4.2.1 we have to find the cycles of $M\_\tilde{G}_{F_i} \times M\_\tilde{G}_{N,\bar{F}_i}$ which can be done in $\mathcal{O}(|X|^2 + |R|^2)$ time. As a result, the total complexity of verifying condition (2) is $\mathcal{O}(|X|^2 + |X| \cdot |\theta| + |X| \cdot T_1 + |R|^2 \cdot T_2)$. Since $|R| \leq |X|(|X|-1)$ and $\mathcal{O}(|\theta|) = \mathcal{O}(|\Sigma| \cdot |X|^2)$, we get $\mathcal{O}(|X| \cdot |\theta| + |X| \cdot T_1 + |X|^4 \cdot T_2) = \mathcal{O}(|\Sigma| \cdot |X|^3 + |X| \cdot T_1 + |X|^4 \cdot T_2)$ as the complexity of verifying condition (2) and diagnosability of F_i.

Later in this section, we will show that the test can be performed in polynomial time. For now, we would like to point out that in general, the transition-time $T(x, x'')$ needed in the computation of timed RTS $\tilde{G}$ may be finite or unbounded sets.

In the following, we discuss cases in which the transition-time sets are either bounded or can be represented as finite unions of intervals: $T(x, x'') = (\cup_{k=1}^{n}[t_l^k, t_u^k]) \cup [t_l^{n+1}, \infty)$, with n bounded by a polynomial function of $|X|$. In these cases, as discussed

[4] Suppose two generators G_1 and G_2 have n_1 and n_2 states, and m_1 and m_2 transitions. $G_1 \times G_2$ can be computed in $\mathcal{O}(n_1 n_2 + m_1 m_2)$ time.

in Remark 4.2.2, the algorithm proposed in this dissertation can be more efficient than the existing methods.

We need a few definitions for further discussion. Let $URch(x)$ denote the set of states of the TDES G that have output $y = \lambda(x)$ and are reachable from a state x using a path along which the output is $y = \lambda(x)$:

$$URch(x) := \{x\} \cup \{x' \in X \mid \exists l \geq 1,\ \exists x_1, \ldots x_{l+1},\ \exists \sigma_1, \ldots \sigma_l, : x_1 = x,$$
$$x_{l+1} = x',\ \text{and}\ (\lambda(x_k) = \lambda(x),\ x_{k+1} \in \delta(x_k, \sigma_k)\ (1 \leq k \leq l))\}$$

For the output-adjacent states x and x'' with $x \Rightarrow x''$, denote the set of states of the TDES G that are reachable from x and co-reachable to x'' by $URch(x, x'')$. In other words,

$$URch(x, x'') := \{x' \mid x' \in URch(x)\ \text{ and }\ x' \Rightarrow x''\}.$$

Furthermore, let $Cyc(x, x'')$ denote the set of cycles in $URch(x, x'')$. Also, define $Traj(x, x'')$ as the trajectories in TDES G from x to x'' through $URch(x, x'')$:

$$Traj(x, x'') := \{\{x_1, \ldots, x_{l+1}\} \mid l \geq 1,\ x_1 = x, x_{l+1} = x'',\ x_k \in$$
$$URch(x, x'')\ (1 \leq k \leq l+1),\ x_i \neq x_j\ (1 \leq i,\ j \leq l,\ j \neq i)$$
$$\text{and}\ (\exists \sigma_1, \ldots \sigma_l : x_{k+1} \in \delta(x_k, \sigma_k)\ (1 \leq k \leq l))\}$$

Lemmas 4.2.2, 4.2.3 and 4.2.4 provide sufficient conditions under which $T(x, x'')$ can be represented as the union of a bounded number of intervals.

Lemma 4.2.2 *For output-adjacent states x, $x'' \in X$, $T(x, x'')$ is a bounded set if and only if $Cyc(x, x'') = \emptyset$.*

Proof $Cyc(x, x'') = \emptyset$ means none of the paths connecting x to x'' through $URch(x, x'')$ contains a cycle; therefore the length of each of these paths is less than or equal to $|X| - 1$ events and hence $\max T(x, x'') < |X|$. As a result, if $Cyc(x, x'') = \emptyset$, then $T(x, x'')$ is bounded. Conversely, $T(x, x'')$ bounded implies $URch(x, x'')$ contains no cycles, i.e., $Cyc(x, x'') = \emptyset$. ■

It should be noted that complicated or unpredictable temporal behavior (such as cycles of unobservable events) is considered undesirable in real-time control. Therefore, we can expect the above lemma to be applicable to a very useful range of control problems.

Under the circumstances of Lemma 4.2.2 and assuming $Cyc(x, x'') = \emptyset$ for all output-adjacent states x and x'', the computation of transition-times from a state can be done using a breadth-first search [5] and takes $\mathcal{O}(T_1(|X|, |\Sigma|)) = \mathcal{O}(|X| + |\theta|) = \mathcal{O}(|\Sigma| \cdot |X|^2)$. Furthermore, the intersections of transition-time sets can be computed in $\mathcal{O}(T_2(|X|)) = \mathcal{O}(|X|)$ since each $T(x, x'')$ can be written in the form of $\cup_{k=1}^{n}[t_l^k, t_u^k]$ with $n < |X|$. [6] As a result, the complexity of verifying time-diagnosability using Theorem 4.2.1 is $\mathcal{O}(|\Sigma| \cdot |X|^3 + |X| \cdot T_1 + |X|^4 \cdot T_2) = \mathcal{O}(|X|^5 + |\Sigma| \cdot |X|^3)$.

Remark 4.2.3 *As shown earlier, the complexity of verifying diagnosability of a TDES using Theorem 4.2.1 is* $\mathcal{O}(|X|^3 \cdot |\Sigma| + |X| \cdot T_1 + |X|^4 \cdot T_2)$*. The complexity of testing diagnosability of a TDES using the test (for untimed DES) in Theorem 4.1.1 is*

[5] The transition-time set $T(x, x'')$ can be found by first constructing the reachability tree for the language generated by the subgenerator of G corresponding to the states $URch(x, x'') \cup \{x''\}$. Each node of the tree corresponds to a unique sequence and thus to a specific state (reached using that sequence) and a transition time (of the sequence). The transition-time set $T(x, x'')$ is the set of transition times of the nodes that correspond to state x''. Note that since by assumption, $Cyc(x, x'') = \emptyset$, the reachability tree will be finite.

[6] The expression $\sum_{k=1}^{n_1}[a_l^k, a_u^k] \cap \sum_{k=1}^{n_2}[b_l^k, b_u^k] = \emptyset$, with n_1, n_2, $a_u^{n_1}$, $b_u^{n_2} < |X|$ can be verified in $\mathcal{O}(|X|)$ since the verification can be done by sorting the numbers a_l^k, a_u^k, b_l^k and b_u^k.

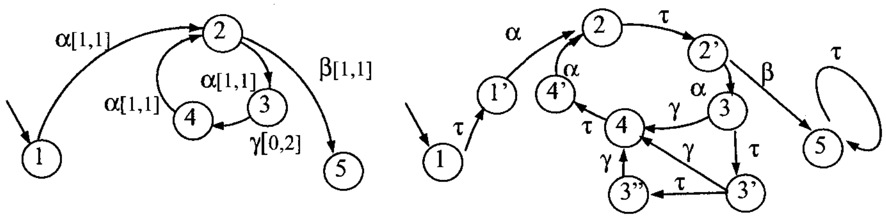
$\mathcal{O}(|X|^4 + |X|^3 \cdot |\Sigma|)$. Under the circumstances of Lemma 4.2.2 (and assuming $|X|$ grows faster than $|\Sigma|$), the complexity for Theorem 4.2.1 and Theorem 4.1.1 become $\mathcal{O}(|X|^5)$ and $\mathcal{O}(|X|^4)$ respectively. So it appears the complexity of the scheme proposed in this section is slightly higher. However, it should be noted that in the above analysis, we had $\mathcal{O}(T_2(|X|)) = \mathcal{O}(|X|)$ which is very conservative. In many practical cases, the transition-time sets $T(x, x'')$ can be either a single interval or the union of a few intervals, and the intersection of time-transition sets in Definition 4.2.2 can be performed easily and fast. As a result, we do not expect the complexity of testing diagnosability using Theorem 4.2.1 to increase faster than $\mathcal{O}(|X|^4)$. Furthermore, while the size of state set $|X|$ in the worst-case increases exponentially with the number of system components, the magnitude of transition-times, however, depend on the length of tick (as the unit of time) and the dynamics of the underlying process, and not on the number of system components. This again indicates that $\mathcal{O}(T_2(|X|)) = \mathcal{O}(|X|)$ is very conservative and points to $\mathcal{O}(|X|^4)$ as the likely complexity of using Theorem 4.2.1. As a matter of fact, as shown in Remark 4.2.2, the procedure proposed in this thesis can be more efficient than the untimed version since it gathers and summarizes the information about the timing of events and subsequently uses the information in diagnosability test.

Finally, it should be noted that while the complexity of verifying diagnosability using Theorem 4.1.1 and (as explained above) Theorem 4.2.1, is $\mathcal{O}(|X|^4)$ and $\mathcal{O}(|X|^5)$, the complexity of verifying dignosability given in [32] is $\mathcal{O}(|X|^2)$. The reason for the difference is that, unlike [32], in this thesis the plant G is assumed to be nondeterministic. ■

Lemmas 4.2.3 and 4.2.4 consider the case of $Cyc(x, x'') \neq \emptyset$ and provide sufficient conditions under which $T(x, x'')$ can be represented as a finite union of intervals.

Lemma 4.2.3 *Suppose $Cyc(x, x'') \neq \emptyset$. Assume that there exist two simple cycles $\mathcal{C}_1$ and $\mathcal{C}_2$ in $Cyc(x, x'')$ and a trajectory $\mathcal{T}$ in $Traj(x, x'')$ such that (i) $\mathcal{C}_1$ and $\mathcal{C}_2$ intersect with $\mathcal{T}$ (i.e., each shares a state with $\mathcal{T}$) and (ii) the greatest common divisor of the transition times of the cycles (in ticks) is equal to 1. Then $T(x, x'')$ can be represented as $T(x, x'') = (\cup_{k=1}^{n}[t_l^k, t_u^k]) \cup [t_l^{n+1}, \infty)$ with $n < |X|^2 + |X|$.* ■

Before proving Lemma 4.2.3, let us examine its implications. Suppose there are two cycles in $Cyc(x, x'')$ with transition-times t_c and t_c+1. Since any two consecutive integers n and $n+1$ ($n \geq 1$) are relatively prime, $gcd(t_c, t_c+1) = 1$. Assuming condition (i) in Lemma 4.2.3 is satisfied, $T(x, x'')$ can be represented by a finite union of intervals. The above case (two cycles with transition-times that are one tick apart) may occur in cases where the TDES is obtained from an activity transition graph (ATG) [63] in which one (or more) unobservable event in a cycle in the ATG has different lower and upper time bounds. In other words, the transition-time of the cycles varies over a range of numbers. An example is discussed in Example 4.2.2.



(a) Activity Transition Graph

(b) Timed DES

Fig. 4.9. An example to convert DES to Timed DES.

Example 4.2.2 *Fig. 4.9(a) depicts an ATG and Fig.4.9(b) shows the corresponding TDES. We assume in the TDES, states 1 and 5 are output-adjacent ($1 \Rightarrow 5$). The outputs in all states are assumed to be the same except state* 5. *In the ATG cycle* $2-3-4$, *the* γ *event has lower and upper time bounds of* 0 *and* 2. *In the TDES, the ATG cycle has produced three cycles. Specifically,* $URch(1,5)$ *is the set of all states except* 5, $Traj(1,5) = \{\{1,1',2,2',5\}\}$, $Cyc(1,5) = \{\{2,2',3,4,4'\}$,$\{2,2',3,3',4,4'\}$, $\{2,2',3,3',3'',4,4'\}\}$. *The transition times of the cycles are* $2,3$ *and* 4, *and* $gcd(2,3) = gcd(3,4) = 1$. *All three simple cycles have a common state with the trajectory* $1 \rightarrow 1' \rightarrow 2 \rightarrow 2' \rightarrow 5$. *In this example,* $T(1,5)= \{2,4,5,6,\ldots\}$ $=\{2\} \cup [4,\infty)$. ■

Proof of Lemma 4.2.3.

Let l_1, l_2, and t_a denote the transition times of $\mathcal{C}_1$ and $\mathcal{C}_2$ and $\mathcal{T}$. By assumption, l_1 and l_2 are relatively prime ($gcd(l_1, l_2) = 1$). According to [64] (Chapter 4, Section 12), for two positive integers l_1 and l_2, the set of all $k_1 l_1 + k_2 l_2$, with positive integers k_1 and k_2, includes all multiples of $gcd(l_1, l_2)$ larger than $l_1 l_2$. In our case, $gcd(l_1, l_2) = 1$ and therefore

$$[l_1 l_2 + 1, \infty) \subseteq \{k_1 l_1 + k_2 l_2 \mid k_1, k_2 \in \{1, 2, \ldots\}\}$$

As the result, $T(x, x'') \supseteq \{t_a + k_1 l_1 + k_2 l_2 \mid k_1,\ k_2 \geq 0\} \supseteq [t_a + l_1 l_2 + 1, \infty)$. This shows that $T(x, x'')$ can be represented as $T(x, x'') = (\cup_{k=1}^{n} [t_l^k, t_u^k]) \cup [t_a + l_1 l_2 + 1, \infty)$ with $n \leq t_a + l_1 l_2 < |X| + |X|^2$. ■

In Example 4.2.2, $t_a = 2$. If we take $\mathcal{C}_1$ and $\mathcal{C}_2$ to be $\{2,2',3,4,4'\}$ and $\{2,2',3,3',4,4'\}$, then we have $l_1 = 2$ and $l_2 = 3$. Now, $T(1,5) = \{2\} \cup [4,\infty)$ which includes $[t_a + l_1 l_2 + 1, \infty) = [9, \infty)$.

Assuming the conditions in Lemma 4.2.3 hold, $T(x, x'')$ can be written in the form $T(x, x'') = T_f(x, x'') \cup [|X|^2 + |X|, \infty)$ where $T_f(x, x'') \subseteq [0, |X|^2 + |X| - 1]$. Now fix state x. To determine the sets $T_f(x, x'')$ (and thus $T(x, x'')$), one may examine event sequences in the TDES containing up to $m := |X|^2 + |X| - 1$ clock ticks. For this, let X_x^{OA} be the set of states that are output-adjacent to x ($X_x^{OA} = \{x'' | \; x \Rightarrow x''\}$), and G_x^{OA} be the subgenerator of G containing the states $X_x^{OA} \cup (\cup\{URch(x, x''), x'' \in X_x^{OA}\})$ (i.e., all states x'' that are output-adjacent to x, and the corresponding states in $URch(x, x'')$). The transitions out of states X_x^{OA} are removed in G_x^{OA}, and the initial state of G_x^{OA} is taken to be x. Next, we can form the product of G_x^{OA} with the generator G_c in Figure 4.10 which counts the ticks up to m. The states of $G_x^{OA} \times G_c$ are of the form (x, k) where x is a state of G_x^{OA} and $0 \leq k \leq m$. Finally, we have $T_f(x, x'') = \{k | \; (x'', k) \text{ is a reachable state of } G_x^{OA} \times G_c\}$. G_x^{OA} can be constructed in $\mathcal{O}(|X| + |\theta|) = \mathcal{O}(|\Sigma| \cdot |X|^2)$ time using the breadth-first search. $G_x^{OA} \times G_c$ can be computed in $\mathcal{O}(|X|^3 + |\Sigma|^2 \cdot |X|^4)$ since G_x^{OA} and G_c have $\mathcal{O}(|X|)$ and $\mathcal{O}(|X|^2)$ states, and $\mathcal{O}(|\theta|) = \mathcal{O}(|\Sigma| \cdot |X|^2)$ and $\mathcal{O}(|\Sigma| \cdot |X|^2)$ transitions. Thus the $T_f(x, x'')$'s (and thus $T(x, x'')$'s) can be computed in $\mathcal{O}(|X|^3 + |\Sigma|^2 \cdot |X|^4)$. Therefore, $T_1(|X|, |\theta|) = |X|^3 + |\Sigma|^2 \cdot |X|^4$. The transition-time set intersection can be performed in $\mathcal{O}(n) = \mathcal{O}(|X|^2)$ time (n is the parameter in Lemma 4.2.3), and hence $T_2(|X|) = |X|^2$. This gives an overall computational complexity of $\mathcal{O}(|X|^6 + |\Sigma|^2 \cdot |X|^5)$ for verifying time-diagnosability. It should be noted that to verify the assumptions of Lemma 4.2.3, one needs to find the simple cycles in the $Cyc(x, x'')$'s and for each simple cycle, determine the set of reachable simple cycles. This can be done in $\mathcal{O}(|X| + |\theta|) = \mathcal{O}(|\Sigma| \cdot |X|^2)$ time.

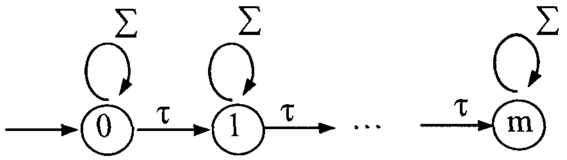
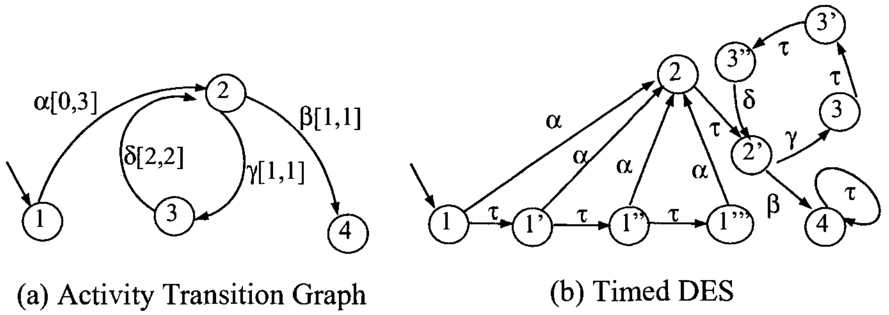


Fig. 4.10. Generator G_c.

Lemma 4.2.3 implies that if the TDES includes cycles of unobservable events, the transition-time sets will have representations as finite union of intervals as long as the duration of cycles vary over a range of numbers. Variation in the duration of cycles is not an unreasonable assumption. Other sufficient conditions may be obtained so as to have representations in the form of finite union of intervals. Lemma 4.2.4 provides one set of such conditions. While Lemma 4.2.3 provides conditions involving one trajectory and multiple cycles, Lemma 4.2.4 provides another set of sufficient conditions involving one cycle and multiple trajectories.

Lemma 4.2.4 *Suppose $Cyc(x, x'') \neq \emptyset$. Assume that there exist a simple cycle $\mathcal{C}$ in $Cyc(x, x'')$ with the transition time of t_c and a set of trajectories such that (i) all trajectories intersect with $\mathcal{C}$ (each shares at least a state with $\mathcal{C}$), (ii) the set of transition times of these trajectories form an interval $[t_l, t_u]$ (i.e., for any $t \in [t_l, t_u]$, there is a trajectory in the above-mentioned set with transition time t), and (iii) $t_c \leq t_u - t_l$. Then $T(x, x'')$ can be represented as $T(x, x'') = (\cup_{k=1}^{n}[t_l^k, t_u^k]) \cup [t_{n+1}, \infty)$ with $n < |X|$.* ■

Before proving Lemma 4.2.4, let us examine its implications. The lemma considers output-adjacent states connected with a set of trajectories intersecting a cycle, with variation in the transition times of the trajectories being equal to or larger than the transition time of the cycle. This case may be encountered in the TDES obtained from an activity transition graph (ATG) in which one (or more) unobservable event in a trajectory in the ATG has different lower and upper time bounds, and the trajectory intersects an unobservable cycle with a transition time less than the difference between the upper and lower time bounds of the trajectory. Figure 4.11 shows an example of such an ATG and the corresponding Timed DES.



(a) Activity Transition Graph

(b) Timed DES

Fig. 4.11. Example for Lemma 4.2.4.

In this example, by assumption, $1 \Rightarrow 4$ and the outputs of all states except 4 are the same. The ATG trajectory $1 \xrightarrow{\alpha} 2 \xrightarrow{\beta} 4$ has produced four trajectories in the TDES with the transition-times of 1, 2, 3, 4 and all trajectories intersect with the cycle $2 \rightarrow 2' \rightarrow 3 \rightarrow 3' \rightarrow 3''$. The transition time of the cycle is 2, which is less than the variation in the transition-times of the trajectories $(4 - 1 = 3)$. Therefore, all of the conditions of Lemma 4.2.4 are satisfied. We observe that $T(1, 4) = [1, \infty)$.

Proof of Lemma 4.2.4.

Let $\mathcal{P}$ be a path from x to x'' in $URch(x, x'')$ consisting of a trajectory $\mathcal{T}$ from the set of trajectories mentioned in the lemma and k repetitions of the cycle $\mathcal{C}$. Then the transition-time of $\mathcal{P}$ is $t + kt_c$ where t is the transition-time of $\mathcal{T}$ $(t_l \leq t \leq t_u)$.

Therefore, $T(x,x'') \supseteq [t_l,t_u] \cup [t_l+t_c,t_u+t_c] \cup [t_l+2t_c,t_u+2t_c] \cup \ldots = \cup_{k=0}^{\infty}[t_l+kt_c,t_u+kt_c]$. Since $t_l+t_c \leq t_u$, $[t_l,t_u] \cup [t_l+t_c,t_u+t_c] = [t_l,t_u+t_c]$, and generally $[t_l+kt_c,t_u+kt_c] \cup [t_l+(k+1)t_c,t_u+(k+1)t_c] = [t_l+kt_c,t_u+(k+1)t_c]$ for $k \geq 0$. As a result, $T(x,x'') \supseteq [t_l,\infty)$, and thus $T(x,x'') = (\cup_{k=1}^{n}[t_l^k,t_u^k]) \cup [t_l,\infty)$, with $n < t_l < |X|$. ■

We can show (similar to Lemma 4.2.3) that if the assumptions of Lemma 4.2.4 are true, $T_1(|X|,|\theta|) = |X|^2 + |X|^3 \cdot |\Sigma|^2$, $T_2(|X|) = |X|$ and time-diagnosability can be verified in $\mathcal{O}(|X|^5 + |\Sigma|^2 \cdot |X|^4)$.

Lemmas 4.2.2, 4.2.3 and 4.2.4 provide conditions under which the timing information of events can be gathered and be used for testing diagnosability efficiently. The test proposed in this section can still be performed with polynomial complexity to verify diagnosability even if the conditions in the above lemmas do not hold. However, as we will see, the computations involved in testing are similar to those in tests based on untimed models and therefore, the proposed test will not have any advantage over the existing methods. The crucial step is the computation of $M\_\tilde{G}_{F_i} \times M\_\tilde{G}_{N,\bar{F}_i}$, where the timing information plays an important role. Note that to perform the timed product (Definition 4.2.2) the sets $T_1(x_1,x_1')$ and $T_2(x_2,x_2')$ are not necessarily required. We only need to know whether or not $T_1(x_1,x_1') \cap T_2(x_2,x_2') \neq \emptyset$. This can be verified as follows. Form subgenerators $G_{x_1}^{OA}$ and $G_{x_2}^{OA}$ (G_x^{OA} was defined after the proof of Lemma 4.2.3). In $G_{x_1}^{OA}$ (resp. $G_{x_2}^{OA}$), replace all non-tick events with a new event e_1 (resp. e_2). Form the synchronous product $G_{x_1}^{OA}||G_{x_2}^{OA}$. It can be seen that $T_1(x_1,x_1') \cap T_2(x_2,x_2') \neq \emptyset$ if and only if (x_1',x_2') is reachable in $G_{x_1}^{OA}||G_{x_2}^{OA}$. The above operations can be done in polynomial time and thus the entire test for diagnosability has polynomial complexity [7]. However, it can be seen that the computations in testing the timing (i.e., $T_1(x_1, x_1') \cap T_2(x_2, x_2') \neq \emptyset$) are similar to the computations used in the existing methods for untimed DES.

4.3 Summary

In this chapter, we present alternative polynomial algorithms for failure diagnosability testing in nondeterministic untimed and timed discrete-event systems in a state-based framework. In timed discrete-event failure diagnosis, the key issue regarding computational complexity is whether the transition time sets can be represented as the union of a bounded number of intervals. We have discussed three cases in which transition times have such representations and as a result, the proposed method for testing diagnosability can be performed efficiently. These cases cover a good range of problems. In cases where Lemmas 4.2.2, 4.2.3 and 4.2.4 do not apply, the proposed test can still be done in polynomial time but the computations will be more or less similar to those of the methods based on untimed DES, and in these cases, our method will not have any particular computational advantage over the existing ones (developed for untimed DES).

[7] It can be shown that the complexity of the test is $\mathcal{O}(|X|^4)$ (same as the tests based on untimed models).

Chapter 5

SENSOR SELECTION IN DISCRETE-EVENT SYSTEMS FOR FAULT DIAGNOSIS

In Chapter 4, we presented algorithms with polynomial-time complexity for testing diagnosability in untimed and timed discrete-event systems in a state-based framework. These algorithms are used in this chapter to develop procedures for sensor selection.

We consider the problem of sensor selection, that is, the problem of sufficient observation for guaranteeing the detection and isolation of failures in discrete event dynamic systems. We propose the concept of "minimal distinguisher", which is a minimal sensor set for distinguishing one system condition from another, and then develop procedures for computing and combining minimal distinguishers to obtain a minimal sensor set for failure detection and isolation.

It is shown in this chapter that taking advantage of the structure of the system, as done in the proposed algorithms, reduces the time and space complexity of testing diagnosablity and sensor selection. A benefit of using minimal distinguishers is that their computation (thus, the computations for sensor selection) may be speeded up using heuristics and expert knowledge. In addition, in Chapter 6, we will show that minimal distinguishers can be used to reduce the computations required for reconfiguring a fault diagnosis system online (as it operates).

5.1 Preliminaries

5.1.1 Plant Model

Assume that the system to be diagnosed can be modelled as a nondeterministic finite-state Moore automaton $G = (X, \Sigma, \delta, x_0, Y, \lambda)$ where X, Σ and Y are the finite state set, event set and output set, respectively. x_0 is the initial state, $\delta : X \times \Sigma \rightarrow 2^X$ the transition function and $\lambda : X \rightarrow Y$ the output map. This model describes the behavior of the system in both normal (N mode) and faulty situations (F modes). We assume that the plant has p **failure modes** $F_1, F_2, \ldots, F_p$. The event set can be partitioned into $\Sigma = \Sigma_N \cup \Sigma_f$, where $\Sigma_f = \{f_1, \ldots, f_p\}$ is the set of failure events and Σ_N, the set of non-failure events. As a result of failure event f_i, failure mode F_i develops in the plant. Simultaneous failures are assumed possible. For example, in a plant with two failure modes F_1 and F_2, the plant can be in one of four **conditions**: N(normal), $F_1, F_2, F_{1,2}$, where $F_{1,2}$ refers to the simultaneous occurrence of both failures. Let $\mathcal{K}$ denote the **condition set** and $\mathcal{F}$ the set of **faulty conditions**. Furthermore, let $\mathcal{F}_i$ denote the set of faulty conditions in which failure mode F_i is present and $\bar{\mathcal{F}}_i = \mathcal{F} - \mathcal{F}_i$. Therefore, for instance in a plant with two failure modes F_1 and F_2, we have $\mathcal{K} = \{N, F_1, F_2, F_{1,2}\}$, $\mathcal{F} = \{F_1, F_2, F_{1,2}\}$, $\mathcal{F}_1 =$

$\{F_1, F_{1,2}\}$, and $\bar{\mathcal{F}}_1 = \{F_2\}$. It is assumed that the finite state set can be partitioned according to system condition. For example, for the case of two failure modes ($n = 2$), $X = X_N \cup X_{F_1} \cup X_{F_2} \cup X_{F_{1,2}}$. The set of states corresponding to condition set $\mathcal{F}$ (resp. $\mathcal{F}_i$) is denoted by $X_{\mathcal{F}}$ (resp. $X_{\mathcal{F}_i}$). The **condition map** $\kappa : X \to \mathcal{K}$ returns the condition of each state. This map can be extended to subsets of X: for $z \subseteq X, \kappa(z) = \cup\{\kappa(x) | x \in z\}$.

We assume that the failure modes are **permanent**; in other words, after the occurrence of a failure, the failure mode remains in the plant indefinitely. Fig. 3.1 shows the state transition graph of a plant with permanent failure modes. Each circle corresponds to a block in the partition of the plant state set X based on plant condition. We observe that the transition graph of a plant with permanent failure modes has a tree structure with N (normal) condition as the root. One of the consequences of this structure is that G cannot have a cycle with states in more than a single condition. For instance, there is no cycle whose states are in X_{F_1} and $X_{F_{1,2}}$. As a result, the set of cycles of G is the union of cycles of individual conditions (i.e., individual blocks in Fig. 3.1).

For $F \in \mathcal{F}$, define $\tilde{G}_F$ as the sub-generator of $\tilde{G}$ consisting of the states in X_F only. Similarly, define $\tilde{G}_N$, $\tilde{G}_{\mathcal{F}}$, $\tilde{G}_{\mathcal{F}_i}$ and $\tilde{G}_{N,\bar{\mathcal{F}}_i}$ as the sub-generators of $\tilde{G}$ corresponding to the states in X_N, $X_{\mathcal{F}}$, $X_{\mathcal{F}_i}$ and $X_N \cup X_{\bar{\mathcal{F}}_i}$ (The initial states of these sub-generators are left undefined.) Note that similar to G, the RTS $\tilde{G}$ also has a tree structure as in Fig. 3.1).

The **output language** $L_o(G, x)$ generated by G from the state $x \in X$ is defined as $L_o(G, x) := \{y_1 y_2 \ldots y_m \subseteq Y^+ \mid y_1 = \lambda(x), \exists x_i \in X \ (1 \le i \le m) : x_1 = x, \, x_{i-1} \Rightarrow x_i,$

$y_i = \lambda(x_i), 2 \leq i \leq m\}$. $L_o(\tilde{G}_{\mathcal{F}_i}, x)$, $L_o(\tilde{G}_{N,\bar{\mathcal{F}}_i}, x)$, $L_o(\tilde{G}_N, x)$ and $L_o(\tilde{G}_{\mathcal{F}}, x)$ are defined similarly.

5.1.2 Diagnosability

The objective in fault diagnosis is to detect and isolate failure modes. The faulty behavior is considered **detectable** if whenever it occurs, it can be distinguished from normal behavior (and hence detected) with finite delay.

Definition 5.1.1 *The faulty behavior is* ***detectable*** *if there exists an integer $N \geq 0$ such that following both the occurrence of failure and the start of diagnosis, the faulty behaviors can be distinguished from normal behavior (i.e., the diagnoser enters a (fault-certain) state z with $\kappa(z) \subseteq \mathcal{F}$), after the occurrence of at most N events in the system.* ■

A failure mode is **diagnosable** if within a finite time delay, it can be detected and isolated. More precisely, diagnosability is defined as Definition 2.3.1.

A DES is said to be **diagnosable** if all of its failure modes are diagnosable. Theorem 4.1.1 provides necessary and sufficient conditions for failure diagnosability (in the state-based framework of [23]) which results in a test for diagnosability with polynomial complexity. The aforementioned theorem considers the case of single-failure scenario. The simple extension to the case of simultaneous failures is as follows.

Theorem 5.1.1 *Assume $z_0 = X$. The faulty behavior F_i is diagnosable if and only if*

1. *For any $x \in X_{\mathcal{F}_i}$, if there is no transition out of x, then $\lambda^{-1}(\lambda(x)) \cap (X - X_{\mathcal{F}_i}) = \emptyset$;*

2. *There is no cycle in $X_{\mathcal{F}_i}$ consisting of states having the same output, say y, unless $\lambda^{-1}(y) \cap (X - X_{\mathcal{F}_i}) = \emptyset$;*

3. *For any $x \in X_{\mathcal{F}_i}$, and $x' \in X_N \cup X_{\bar{\mathcal{F}}_i}$ satisfying $\lambda(x) = \lambda(x')$, we have*

$$\{s | s \in L_o(\tilde{G}_{\mathcal{F}_i}, x) \cap L_o(\tilde{G}_{N,\bar{\mathcal{F}}_i}, x'), |s| \geq |X|^2\} = \emptyset$$

■

Condition (1) states that there should be no deadlock state in $X_{\mathcal{F}_i}$ with no transition out of the state unless the output in that state can be generated only when F_i has occurred. Such an output is called **F_i-indicative**. Similarly, condition (2) states that there should be no cycles with constant output in $X_{\mathcal{F}_i}$ unless the constant output is F_i-indicative. Finally, condition (3) states that there should be no common output cycle in $X_{\mathcal{F}_i}$ and $X - X_{\mathcal{F}_i} = X_N \cup X_{\bar{\mathcal{F}}_i}$ (otherwise F_i cannot be distinguished and hence will be undiagnosable).

Using the results in Theorem 5.1.1, we can obtain necessary and sufficient conditions for fault detectability by simply replacing condition set $\mathcal{F}_i$ with $\mathcal{F}$.

Theorem 5.1.2 *Assume $z_0 = X$. The faulty behavior is detectable if and only if*

1. *For any $x \in X_{\mathcal{F}}$, if there is no transition out of x, then $\lambda^{-1}(\lambda(x)) \cap X_N = \emptyset$;*

2. *There is no cycle in $X_{\mathcal{F}}$ consisting of states having the same output, say y, unless $\lambda^{-1}(y) \cap X_N = \emptyset$;*

3. For any $x \in X_{\mathcal{F}}$, and $x' \in X_N$ satisfying $\lambda(x) = \lambda(x')$, we have

$$\{s | s \in L_o(\tilde{G}_{\mathcal{F}}, x) \cap L_o(\tilde{G}_N, x'), |s| \geq |X|^2\} = \emptyset$$

■

5.2 Problem Formulation

Failure detection and isolation procedure uses sensor observations, which means that failure diagnosability depends in part on the set of sensors used. Given a set of sensors, some of the failure modes may not be diagnosable. On the other hand, some of the sensors may provide redundant information and thus not be necessary for failure diagnosis. Through the above consideration, a problem arises: Is there a minimal sensor set for the diagnosability of a given set of failure modes in the plant?

In this chapter, we consider the minimal sensor selection problem. In this problem, we are given a dynamic system with failure modes and a set of sensors. Suppose we use the same system model as defined in Section 5.1 and are given a sensor set $C_{tot} = \{c_1, \ldots, c_{n_s}\}$. For any sensor $c_j \in C_{tot}$, the output map is $\lambda_{(j)} : X \rightarrow Y_{(j)}$. The output map of the system is then $\lambda : X \rightarrow Y$, where $\lambda(X) = (\lambda_{(1)}(X), \ldots, \lambda_{(n_s)}(X))$, and $Y = Y_{(1)} \times \ldots \times Y_{(n_s)}$. Suppose we want to restrict output tracking from the complete sensor set to a subset of sensors. For a given subset of sensors $C_J = \{c_{j_1}, \ldots, c_{j_{|J|}}\}$, where $J \subseteq \{1, \ldots, n_s\}$, $j_1 < j_2 < \ldots < j_{|J|}$, and $C_J \subseteq C_{tot}$, we can represent the output map as $\lambda_J : X \rightarrow Y_J$, where $\lambda_J(x) = (\lambda_{(j_1)}(x), \ldots, \lambda_{(j_{|J|})}(x))$, and $Y = Y_{(j_1)} \times \ldots \times Y_{(j_{|J|})}$.

We consider sensor selection for failure detection and failure diagnosis (i.e., detection and isolation). The necessary and sufficient conditions for solvability of fault detection are given in Theorem 5.1.2. As mentioned before, whether these conditions hold or not depends in part on the sensor set. Let $\mathcal{D}$ denote the set of sensor sets for which the fault detection problem is solvable: $\mathcal{D} = \{C | C \subseteq C_{tot}$ and fault detection using sensors in C is solvable$\}$.

The conditions for failure diagnosability are also given in Theorem 5.1.1. If these conditions hold for all failure modes, then the failure detection and isolation is solvable. Define $\mathcal{DI} = \{C | C \subseteq C_{tot}$ and failure detection and isolation using sensors in C is solvable$\}$.

In this chapter, we study the problems of finding minimal sensor sets of $\mathcal{D}$ and $\mathcal{DI}$. Note that C is a minimal element of $\mathcal{D}$ if $C \in \mathcal{D}$ and for any $C' \subset C$ $(C' \neq C)$, we have $C' \notin \mathcal{D}$. Minimal elements of $\mathcal{DI}$ are defined similarly.

Assume that using the entire set C_{tot}, the problems of failure detection, and failure detection and isolation are both solvable. This implies that $\mathcal{D} \neq \emptyset$ and $\mathcal{DI} \neq \emptyset$ (otherwise searching for minimal elements would be unnecessary.).

A straightforward solution (used in literature) for finding minimal elements of $\mathcal{D}$ and $\mathcal{DI}$ is to use a top-down approach starting from C_{tot}. Procedure 5.1 is a top-down solution for minimal sensor selection in failure detection. The procedure starts from C_{tot} and in each step it removes one sensor from the set until the sensor set becomes minimal. Note that in step 2a, the test for solvability of failure detection problem is performed (which amounts to the verification of the conditions in Theorem 5.1.2).

Procedure 5.1: Given a sensor set $C_{tot} = \{c_1, \ldots, c_{n_s}\}$ and condition set $\mathcal{K}$.

```
1. Initialization: $C_{\mathcal{D}} := C_{tot}$
2. For all $c \in C_{\mathcal{D}}$,
           Compute $\tilde{G}$
2a.        If $C_{\mathcal{D}} - \{c\} \notin \mathcal{D}$
                 Go to 2b
           End (If)
        $C_{\mathcal{D}} := C_{\mathcal{D}} - \{c\}$
2b. Continue
    End (For)
```

* $C_{\mathcal{D}}$ is a minimal sensor set for failure detection

Procedure 5.1: A top-down solution for sensor selection for failure detection.

If in Step $2a$, $\mathcal{D}$ is replaced with $\mathcal{DI}$, a procedure for finding minimal sensor selection for the failure detection and isolation problem will be obtained. Similar bottom-up procedures for finding minimal sensor sets can be developed.

Later in this chapter, we introduce an algorithm for testing failure diagnosability that takes advantage of the tree structure of the plant (shown in Figure 3.1) to reduce time and space complexity of the test. This also leads to algorithms for finding minimal sensor sets that take advantage of the tree structure of the plant. Specifically, instead of searching for sensor sets that solve failure detection and failure detection and isolation problems, the proposed algorithms look for sensor sets that allow us to *distinguish* one condition $A \in \mathcal{K}$ from another condition $A' \in \mathcal{K}$. These sensor sets are referred to as *distinguishers.* Following the proposed algorithms, sensor sets for failure detection, and failure detection and isolation are obtained by combining distinguishers. In other words, instead of solving the entire problem of sensor selection, we break the problem into a set of smaller problems of finding distinguishers.

5.3 Minimal Distinguishers

In this section, we propose a new concept of *minimal distinguisher* and provide the procedure to calculate it formally. According to Theorem 5.1.1, failure mode F_i is diagnosable if and only if the outputs of deadlock states (condition 1) and cycles with constant outputs in the set of conditions $\mathcal{F}_i$ (condition in which F_i has occurred) can be distinguished from the outputs generated in other conditions, and that the periodic output cycles in conditions $\mathcal{F}_i$ are distinguishable from those in other conditions. This motivates the following definition of distinguishability of conditions.

Definition 5.3.1 *Let $\mathcal{A}$ and $\mathcal{A}'$ be two nonempty disjoint subsets of the condition set $\mathcal{K}$ ($\emptyset \neq \mathcal{A}, \mathcal{A}' \subseteq \mathcal{K}$, $\mathcal{A} \cap \mathcal{A}' = \emptyset$) and $X_{\mathcal{A}}$ and $X_{\mathcal{A}'}$ the corresponding states. Condition set $\mathcal{A}$ is* ***distinguishable*** *from $\mathcal{A}'$ if: (1) for any $x \in X_{\mathcal{A}}$, if there is no transition out of x, then $\lambda^{-1}(\lambda(x)) \cap X_{\mathcal{A}'} = \emptyset$; (2) there is no cycle in $X_{\mathcal{A}}$ consisting of states having the same output, say y, unless the output symbol y satisfies $\lambda^{-1}(y) \cap X_{\mathcal{A}'} = \emptyset$; (3) for any $x \in X_{\mathcal{A}}$ and for any $x' \in X_{\mathcal{A}'}$ satisfying $\lambda(x) = \lambda(x')$, we have $\{s|\ s \in L_o(\tilde{G}_{\mathcal{A}}, x) \cap L_o(\tilde{G}_{\mathcal{A}'}, x'),\ |s| \geq |X|^2\} = \emptyset$.* ■

Next we define distinguisher sensor sets.

Definition 5.3.2 *Let C be a non-empty subset of sensors ($C \subseteq C_{tot}$, $C \neq \emptyset$). For two disjoint condition sets $\mathcal{A}$ and $\mathcal{A}'$, if $\mathcal{A}$ is distinguishable from $\mathcal{A}'$ based on the outputs from the sensors in C, then the sensor set C is called an $\mathcal{A}|\mathcal{A}'$-**distinguisher**.*■

The set of $\mathcal{A}|\mathcal{A}'$-distinguishers is denoted by $\mathbf{SD}(\mathcal{A}|\mathcal{A}')$.

Definition 5.3.3 *Let the sensor set C be an $\mathcal{A}|\mathcal{A}'$ -distinguisher: $C \in$ **SD**$(\mathcal{A}|\mathcal{A}')$. C is called a **minimal distinguisher** if none of the proper subsets of C is an $\mathcal{A}|\mathcal{A}'$ -distinguisher.* ■

The set of minimal $\mathcal{A}|\mathcal{A}'$- distinguishers is denoted by SMD$(\mathcal{A}|\mathcal{A}')$.

Example 5.3.1 *Consider the system in Figure 5.1 which has two failure modes F_1 and F_2. Simultaneous failures are assumed possible and thus $\mathcal{F} = \{F_1, F_2, F_{1,2}\}$. The sensor set $C_{tot} = \{c_1, c_2, c_3\}$. The output sets of c_1, c_2, c_3 are $Y_{(1)} = \{\alpha, \beta, \gamma, \delta\}$, $Y_{(2)} = \{l, h\}$ and $Y_{(3)} = \{e, d\}$, respectively. In Figure 5.1, the output at each state is written next to it.*

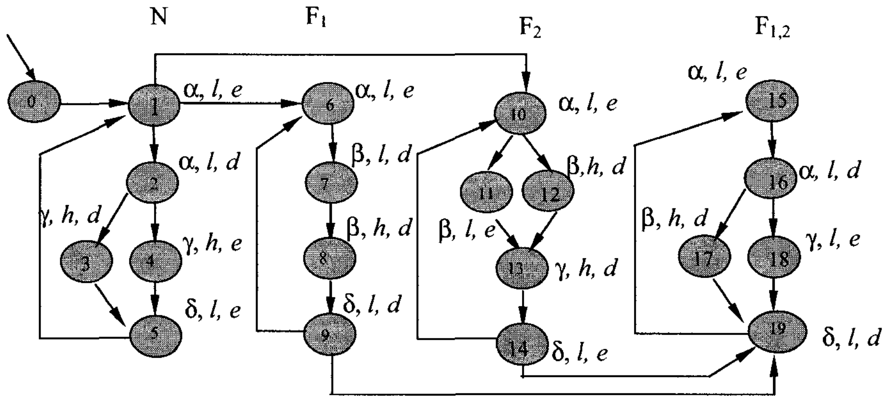


Fig. 5.1. Example 5.3.1: An Example for Sensor Selection.

*We can see that condition F_1 is distinguishable from N (normal) based on the output sequence from sensor c_1 alone or from sensors c_2 and c_3 together. Thus, $\{c_1\}$, $\{c_2, c_3\} \in$ **SD**$(\{F_1\}|\{N\})$. In the case of c_1, for instance, the output sequence generated by c_1 in N and F_1 conditions are periodic and different. Note that both $\{c_1\}$, $\{c_2, c_3\}$ are minimal $\{F_1\}|\{N\}$-distinguishers. For instance, in the case of $\{c_2, c_3\}$,* *suppose c_2 is removed from the sensor set. c_3 can generate the output sequence $(ed)^*$ in both N and F_1 conditions. Thus c_3 is not an $\{F_1\}|\{N\}$-distinguisher. The same is true if c_3 is removed. Thus $\{c_2, c_3\}$ is a minimal $\{F_1\}|\{N\}$-distinguisher.* ■

From now on, in $\mathbf{SD}(\mathcal{A}|\mathcal{A}')$ if $\mathcal{A}$ (or $\mathcal{A}'$) is singleton, we drop the curly braces in $\mathcal{A}$ ($\mathcal{A}'$). For example, instead of $\mathbf{SD}(\{F_1\}|\{N\})$, we write $\mathbf{SD}(F_1|N)$.

It follows from Definition 5.3.1 that for a given sensor set, failure detection is possible if and only if $\mathcal{F}$ is distinguishable from N. Thus, $\mathcal{D} = \mathbf{SD}(\mathcal{F}|N)$. The following theorem shows that given a sensor set, failure detection is possible if and only if the sensors can be used to distinguish every faulty condition from normal.

Theorem 5.3.1 $\mathcal{D} = \cap_{F \in \mathcal{F}}$ ***SD***$(F|N)$.

Proof We prove this theorem for the case of two failure modes, with $\mathcal{F} = \{F_1, F_2, F_{1,2}\}$. The extension to the general case will be similar. Suppose

$$C \in \mathbf{SD}(F_1|N) \cap \mathbf{SD}(F_2|N) \cap \mathbf{SD}(F_{1,2}|N) \tag{5.1}$$

For any deadlock state $x \in X_{F_1}$, $C \in \mathbf{SD}(F_1|N)$ implies $\lambda^{-1}(\lambda(x)) \cap X_N = \emptyset$ and thus $\lambda^{-1}(\lambda(x)) \subseteq X_{F_1} \cup X_{F_2} \cup X_{F_{1,2}}$. Similarly, for any deadlock state x in X_{F_2} or $X_{F_{1,2}}$, $\lambda^{-1}(\lambda(x)) \subseteq X_{F_1} \cup X_{F_2} \cup X_{F_{1,2}}$. Thus condition (1) in Theorem 5.3.1 is true. Since by assumption, the failure modes are permanent and thus, G has a tree structure (Figure 3.1), the cycles in $X_{F_1} \cup X_{F_2} \cup X_{F_{1,2}}$ are the union of cycles in X_{F_1}, X_{F_2} and $X_{F_{1,2}}$. For any cycle in X_{F_1} (or X_{F_2} or $X_{F_{1,2}}$) with constant output y, by (5.1), $\lambda^{-1}(y) \cap X_N = \emptyset$ and thus $\lambda^{-1}(y) \subseteq X_{F_1} \cup X_{F_2} \cup X_{F_{1,2}}$. Thus condition (2) of Theorem 5.3.1 holds. Condition (3) of Theorem 5.3.1 holds similarly. As a result, $C \in \mathcal{D}$.

Conversely, suppose $C \in \mathcal{D}$. If x is a deadlock state in X_{F_1}, then by condition (1) of Theorem 5.3.1, $\lambda^{-1}(\lambda(x)) \cap X_N = \emptyset$. Therefore, condition (1) in Definition 5.3.1 for distinguishability of F_1 from N is satisfied. Similarly for any cycle in X_{F_1} with constant output y, $C \in \mathcal{D}$ and condition (2) of Theorem 5.3.1 implies $\lambda^{-1}(y) \cap X_N = \emptyset$ (thus condition (2) for distinguishability of F_1 from N holds). Furthermore, condition (3) of Theorem 5.3.1 implies condition (3) for distinguishability of F_1 from N. Thus, by Definition 5.3.2, $C \in \mathbf{SD}(F_1|N)$. Similarly, we can show $C \in \mathbf{SD}(F_2|N)$ and $C \in \mathbf{SD}(F_{1,2}|N)$. ■

Example 5.3.1 (Continued): We can verify that $\mathbf{SD}\ (F_1|N)$=$\{\{c_1\}, \{c_1, c_2\}, \{c_1, c_3\}, \{c_2, c_3\}, \{c_1, c_2, c_3\}\}$, $\mathbf{SD}(F_2|N)$=$\{\{c_1\}, \{c_1, c_2\}, \{c_1, c_3\}, \{c_2, c_3\}, \{c_1, c_2, c_3\}\}$, $\mathbf{SD}\ (F_{1,2}|N)$=$\{\{c_1, c_2\}, \{c_1, c_3\}, \{c_1, c_2, c_3\}\}$. Thus by Theorem 5.3.1, $\mathcal{D} = \mathbf{SD}(F_1|N) \cap \mathbf{SD}(F_2|N) \cap \mathbf{SD}(F_{1,2}|N) = \{\{c_1, c_2\}, \{c_1, c_3\}, \{c_1, c_2, c_3\}\}$. ■

Since $\mathcal{D} = \mathbf{SD}(\mathcal{F}|N)$, Theorem 5.3.1 can be expressed as $\mathbf{SD}(\mathcal{F}|N) = \cap_{F \in \mathcal{F}} \mathbf{SD}(F|N)$. A generalization of this result is as follows.

Theorem 5.3.2 *Let $\emptyset \neq \mathcal{A}$, $\mathcal{A}' \subseteq \mathcal{K}$ and $\mathcal{A} \cap \mathcal{A}' = \emptyset$. Then,*

$\boldsymbol{SD}(\mathcal{A}|\mathcal{A}')$=$\cap_{F \in \mathcal{A}} \cap_{F' \in \mathcal{A}'}$ $\boldsymbol{SD}(F|F')$

Proof Similar to Theorem 5.3.1 and omitted for brevity. ■

The following theorem characterizes the sensor sets for which the system will be diagnosable in terms of distinguishers.

Theorem 5.3.3 $\mathcal{DI} = \cap_{F \in \mathcal{F}} \boldsymbol{SD}(F|N) \cap (\cap\{\boldsymbol{SD}(F|F') \mid F, F' \in \mathcal{F} \textit{ and } F \neq F'\})$.

Proof For brevity, we prove the theorem when the system has two failure modes: $\mathcal{K} = \{N, F_1, F_2, F_{1,2}\}$. Extension to the general case will be similar.

The system is diagnosable if and only if F_1 and F_2 are diagnosable. It follows from Theorem 5.1.1, that F_1 (resp. F_2) is diagnosable if and only if $\{F_1, F_{1,2}\}$ (resp. $\{F_2, F_{1,2}\}$) is distinguishable from $\{N, F_2\}$ (resp. $\{N, F_1\}$). Therefore,

$$\mathcal{DI} = \mathbf{SD}(\{F_1, F_{1,2}\}|\{N, F_2\}) \cap \mathbf{SD}(\{F_2, F_{1,2}\}|\{N, F_1\}) \tag{5.2}$$

By Theorem 5.3.2, $\mathbf{SD}(\{F_1, F_{1,2}\}|\{N, F_2\}) = \mathbf{SD}(F_1|N) \cap \mathbf{SD}(F_1|F_2) \cap \mathbf{SD}(F_{1,2}|N) \cap \mathbf{SD}(F_{1,2}|F_2)$ and $\mathbf{SD}(\{F_2, F_{1,2}\}|\ \{N, F_1\}) = \mathbf{SD}(F_2|N) \cap \mathbf{SD}(F_2|F_1) \cap \mathbf{SD}(F_{1,2}|N) \cap \mathbf{SD}$ $(F_{1,2}|F_1)$. Substituting from these two equations in (5.2), we get the desired result. ■

In the following two sections, we will use the results of Theorems 5.3.1 and 5.3.3 to find minimal elements of $\mathcal{D}$ (for the failure detection problem) and $\mathcal{DI}$ (for the failure detection and isolation problem). The proposed algorithms use minimal distinguishers. Minimal distinguishers can be obtained using a top-down or a bottom-up approach. Procedure 5.2 provides a top-down approach for finding an element of $\mathbf{SMD}(A_1|A_2)$, denoted in Procedure 5.2 by $C_{A_1|A_2}$, where A_1, A_2 are two distinct conditions (A_1, $A_2 \in \mathcal{K}$, $A_1 \neq A_2$). Procedure 5.2 starts with $C_{A_1|A_2} = C_{tot}$ and removes sensors from it until $C_{A_1|A_2}$ becomes a minimal distinguisher of A_1 from A_2.

Next we find the computational complexity of verifying distinguishability (i.e., conditions in Definition 5.3.1) and from that, we obtain the computational complexity of Procedure 5.2. Suppose $C \subseteq C_{tot}$ and we want to verify whether $C \in \mathbf{SD}(A_1|A_2)$ for $A_1, A_2 \in \mathcal{K}$ with $A_1 \neq A_2$. We assume that each condition $A \in \mathcal{K}$ (each block in

Figure 3.1) has $\mathcal{O}(|X_A|) = M$ states and $\mathcal{O}(|\theta_A|) = n_e M^2$ transitions where $n_e = |\Sigma|$, the size of the event set of G.

The deadlock states of X_{A_1} in condition (1) of Definition 5.3.1 can be computed and their outputs compared with states in X_{A_2} in $\mathcal{O}(|X_{A_1}| + |X_{A_2}|) = \mathcal{O}(M)$. The cycles with constant output in condition (2) can be computed in $\mathcal{O}(|X_A| + |\theta_A|) = \mathcal{O}(M + n_e M^2) = \mathcal{O}(n_e M^2)$. Thus condition (2) can be verified in $\mathcal{O}(n_e M^2)$.

```
Procedure 5.2 Given a sensor set $C_{tot} = \{c_1, ..., c_{n_s}\}$ and condition set $\mathcal{K}$,
$A_1, A_2 \in \mathcal{K}$ and $A_1 \neq A_2$.
1. Initialization: $C_{A_1|A_2} := C_{tot}$
2. For all $c \in C_{A_1|A_2}$
        Compute $\tilde{G}_{A_1}$ and $\tilde{G}_{A_2}$
        If $C_{A_1|A_2} - \{c\} \notin$ SD($A_1| A_2$)
            go to 2a
        End (If)
      $C_{A_1|A_2} := C_{A_1|A_2} - \{c\}$
2a    Continue
   End (For)
```

Procedure 5.2: $C_{A_1|A_2}$ is a minimal $A_1| A_2$-distinguisher.

As shown in Section 4.1, condition (3) can be verified by finding the cycles of $M\_\tilde{G}_{A_1} \times M\_\tilde{G}_{A_2}$, where $M\_\tilde{G}_{A_1}$ and $M\_\tilde{G}_{A_2}$ are automata obtained from $\tilde{G}_{A_1}$ and $\tilde{G}_{A_2}$ through a process similar to the conversion of Moore machines to Mealy machines. Thus, the sizes of state and transition sets of $M\_\tilde{G}_{A_1} \times M\_\tilde{G}_{A_2}$ and $\tilde{G}_{A_1} \times \tilde{G}_{A_2}$ are of the same order. The number of states and transitions of automata $\tilde{G}_{A_1}$ (and $\tilde{G}_{A_2}$) is $\mathcal{O}(M)$ and $\mathcal{O}(M^2)$. Therefore, the complexity of computing $M\_\tilde{G}_{A_1} \times M\_\tilde{G}_{A_2}$ is $\mathcal{O}(M^4)$. $M\_\tilde{G}_{A_1} \times M\_\tilde{G}_{A_2}$ will have $\mathcal{O}(M^2)$ states and $\mathcal{O}(M^4)$ transitions. As a result, verifying conditions (3) which involves finding the cycles of $M\_\tilde{G}_{A_1} \times M\_\tilde{G}_{A_2}$

will have $\mathcal{O}(M^2+M^4) = \mathcal{O}(M^4)$ complexity. Thus the test $C \in \mathbf{SD}(A_1|A_2)$ (verifying distinguishability) can be done in $\mathcal{O}(M^4 + n_e M^2)$ time.

Now we find the complexity of Procedure 5.2. The main loop has to be repeated n_s times. Each time we compute $\tilde{G}_{A_1}$ and $\tilde{G}_{A_2}$ which takes $\mathcal{O}(|X_A|(|X_A| + |\theta_A|)) = \mathcal{O}(M(M + n_e M^2)) = \mathcal{O}(n_e M^3)$ and test $C_{A_1|A_2} - \{c\} \in \mathbf{SD}(A_1|A_2)$ (verify the distinguishability of A_1 from A_2 with sensors $C_{A_1|A_2} - \{c\}$) which takes $\mathcal{O}(M^4 + n_e M^2)$ time. Therefore, the complexity of Procedure 5.2 is $\mathcal{O}(n_s M^4 + n_s n_e M^3)$.

5.4 Sensor Selection For Fault Detection and Isolation

In this section, we present two algorithms for finding a minimal set of sensors to allow failure detection (i. e., a minimal element of $\mathcal{D}$), and failure detection and isolation, respectively.

5.4.1 Sensor Selection for Failure Detection

A proposed algorithm is given in Procedure 5.3 and can be regarded as a bottom-up algorithm for solving the problem. It is assumed that for every faulty condition $F \in \mathcal{F}$, a minimal $F|N$-distinguisher has been previously obtained using Procedure 5.2.

The desired minimal solution for failure detection is denoted by $C_{\mathcal{F}|\mathcal{N}}$. Step 1 initializes $C_{\mathcal{F}|\mathcal{N}}$ to the empty set. Then step 2 forms the union of all available $C_{\mathcal{F}|\mathcal{N}}$. Therefore, at the end of step 2, $C_{\mathcal{F}|\mathcal{N}} \in \mathbf{SD}(F|N)$ for all $F \in \mathcal{F}$ and thus by Theorem 5.3.2, $C_{\mathcal{F}|\mathcal{N}} \in \mathbf{SD}(\mathcal{F}|N)$. In step 3, we reduce the sensors in $C_{\mathcal{F}|\mathcal{N}}$ to make $C_{\mathcal{F}|\mathcal{N}}$ a minimal element of $\mathbf{SD}(\mathcal{F}|N)$=$\mathcal{D}$. This is done by removing one sensor, c, and

```
Procedure 5.3: Given a sensor set $C_{tot} = \{c_1, \ldots, c_{n_s}\}$ and minimal
$F|N$-distinguishers $C_{F|N}$, for every $F \in \mathcal{F}$.
1. Initialization: $C_{\mathcal{F}|\mathcal{N}} := \varnothing$
2. For all $F \in \mathcal{F}$
        $C_{\mathcal{F}|\mathcal{N}} := C_{\mathcal{F}|\mathcal{N}} \cup C_{F|N}$
   End (For)
3. For all $c \in C_{\mathcal{F}|\mathcal{N}}$
        Compute $\tilde{G}_N$
        For all $F \in \mathcal{F}$
                Compute $\tilde{G}_F$
                If $C_{\mathcal{F}|\mathcal{N}} - \{c\} \notin$ SD($F|N$)
                    go to 3a
                End (If)
        End (For)
        $C_{\mathcal{F}|\mathcal{N}} := C_{\mathcal{F}|\mathcal{N}} - \{c\}$
3a      Continue
   End (For)
```

Procedure 5.3: $C_{\mathcal{F}|\mathcal{N}}$ is a minimal $\mathcal{F}|\mathcal{N}$-distinguisher (a minimal sensor set for failure detection).

then testing whether $C_{\mathcal{F}|\mathcal{N}} - \{c\}$ is an $\mathcal{F}|N$-distinguisher or not. To test whether $C_{\mathcal{F}|\mathcal{N}} - \{c\}$ is an $\mathcal{F}|N$-distinguisher, we have used Theorem 5.3.2, and test whether $C_{\mathcal{F}|\mathcal{N}} - \{c\}$ is $F|N$-distinguisher for all $F \in \mathcal{F}$. In summary, after the termination of the algorithm, $C_{\mathcal{F}|\mathcal{N}}$ will be a minimal element of SD($\mathcal{F}|N$)=$\mathcal{D}$. Thus we have the following.

Theorem 5.4.1 *Procedure 5.3 generates a minimal sensor set for failure detection, i.e., $C_{\mathcal{F}|\mathcal{N}}$ is a minimal element of $\mathcal{D}$.* ■

The proof is straightforward and is given in the preceding paragraph.

Example 5.4.1 *(Example 5.3.1 continued): Starting Procedure 5.2 with $C_{F_1|N} = \{c_1\}$, $C_{F_2|N} = \{c_2, c_3\}$ and $C_{F_{1,2}|N} = \{c_1, c_3\}$, after step 2, $C_{\mathcal{F}|\mathcal{N}} = \{c_1, c_2, c_3\}$. Next* *in step 3, first we remove* c_1. $C_{\mathcal{F}|\mathcal{N}} - \{c_1\} = \{c_2, c_3\}$, *however, is not an* $F_{1,2}|N$-*distinguisher. So* c_1 *stays in* $C_{\mathcal{F}|\mathcal{N}}$. *Next we consider* $C_{\mathcal{F}|\mathcal{N}} - \{c_2\} = \{c_1, c_3\}$ *which will be* $F_1|N$, $F_2|N$, *and* $F_{1,2}|N$-*distinguisher. Therefore,* $C_{\mathcal{F}|\mathcal{N}}$ *is reduced to* $\{c_1, c_3\}$. *Next we verify that* $C_{\mathcal{F}|\mathcal{N}} - \{c_3\} = \{c_1\}$ *is not an* $F_{1,2}|N$-*distinguisher. Thus the algorithm terminates with* $C_{\mathcal{F}|\mathcal{N}} = \{c_1, c_3\}$ *as the final result.* ■

Next let us find the computational complexity of Procedure 5.3. In previous section, we showed that the complexity of testing $C \in \mathbf{SD}(A_1|A_2)$ is $\mathcal{O}(M^4 + n_e M^2)$. First let us assume that simultaneous occurrence of up to only two failures is possible. Therefore, the condition set $\mathcal{K} = \{N, F_1, \ldots, F_p, F_{1,2}, \ldots, F_{p-1,p}\}$. Suppose the minimal distinguishers to start Procedure 5.3 are calculated using Procedure 5.2. Since there are $\mathcal{O}(p^2)$ conditions, the computational complexity of finding the required minimal distinguishers is $\mathcal{O}(n_s p^2 M^4 + n_e n_s p^2 M^3)$. Step 2 of Procedure 5.3 can be performed in $\mathcal{O}(p^2 n_s)$, and step 3 in $\mathcal{O}(n_s(n_e M^3 + p^2(n_e M^3 + M^4 + n_e M^2)))$. Therefore, the computational complexity of Procedure 5.3 is $\mathcal{O}(n_s p^2 M^4 + n_e n_s p^2 M^3)$. Next, let us extend the results to the simultaneous occurrence of up to m failures (with $m << p$). In this case, we have $\mathcal{O}(|\mathcal{F}|) = \mathcal{O}(p^m)$ failure conditions and Procedure 5.3 for finding $C_{\mathcal{F}|N}$ requires $\mathcal{O}(n_s p^m M^4 + n_e n_s p^m M^3)$ time. (Simultaneous occurrence of a large number of failure modes is unlikely and therefore not considered here.)

The number of events, n_e, typically increases linearly with the number of system components whereas M, the number of states in each condition, increase exponentially with the number of system components. Therefore, M increases much faster than n_e.

With the above assumption, the complexity of sensor selection for fault detection will be $\mathcal{O}(n_s p^m M^4)$.

The proposed procedure for finding minimal sensor set for failure detection takes advantage of the tree structure of the plant G (Figure 3.1) and its consequence as expressed in Theorem 5.3.1 to reduce the random-access memory requirement (i.e., less space complexity) in two ways. First it constructs an element of $\mathcal{D}$ using minimal $F|N$-distinguishers (step 2 of Procedure 5.3). For constructing minimal $A_1|A_2$-distinguishers, only the sub-generators of two conditions A_1 and A_2 (G_{A_1} and G_{A_2}) are used (not the entire plant G). Second, the trimming of $C_{\mathcal{F}|\mathcal{N}}$ (calculated in step 2) to a minimal element of $\mathcal{D}$ in step 3 is done by performing a set of distinguishability tests for only two conditions at a time. This again is a consequence of Theorem 5.3.1 and thus, the tree structure of G.

Now we will further discuss the benefits of using minimal distinguishers when we discuss sensor selection for failure detection and isolation.

5.4.2 Sensor Selection for Failure Detection and Isolation

The proposed algorithm is given in Procedure 5.4 and can be regarded as a bottom-up algorithm for solving the problem. It is assumed that minimal distinguishers $C_{F|N}$ for every $F \in \mathcal{F}$ and minimal distinguishers $C_{F|F'}$ for every $F, F' \in \mathcal{F}$ with $F \neq F'$ have been obtained (using, say, Procedure 5.2). The desired minimal solution for failure detection and isolation is denoted by C_{iso}. Step 1 initializes C_{iso} to the empty set. Then step 2 forms the union of all minimal distinguishers. Therefore, at the end of step 2, $C_{iso} \in \mathbf{SD}(F|N)$ for all $F \in \mathcal{F}$, and $C_{iso} \in \mathbf{SD}(F|F')$ for all $F, F' \in \mathcal{F}$ (with

$F \neq F'$), and thus by Theorem 5.3.3, $C_{iso} \in \mathcal{DI}$. In step 3, we reduce the sensors in C_{iso} to make C_{iso} a minimal element of $\mathcal{DI}$. This is done by removing one sensor, c, and then testing whether $C_{iso} - \{c\} \in \mathcal{DI}$. To test whether $C_{iso} - \{c\} \in \mathcal{DI}$, we have used Theorem 5.3.3, and test whether $C_{iso} - \{c\}$ is $F|N$-distinguisher for all $F \in \mathcal{F}$ and $F|F'$-distinguisher for all $F, F' \in \mathcal{F}$, with $F \neq F'$. In summary, after the termination of the algorithm, C_{iso} will be a minimal element of $\mathcal{DI}$. Thus we have the following.

Theorem 5.4.2 *Procedure 5.4 generates a minimal sensor set for failure detection and isolation, i.e., C_{iso} is a minimal element of $\mathcal{DI}$.*

Procedure 5.4: Given a sensor set $C_{tot} = \{c_1, \ldots, c_{n_s}\}$ and minimal distinguishers $C_{F|N}$ for every $F \in \mathcal{F}$ and minimal distinguishers $C_{F|F'}$ for every $F, F' \in \mathcal{F}$ with $F \neq F'$.

```
1. Initialization: $C_{iso} := \varnothing$
2. $C_{iso} = \cup_{F\in\mathcal{F}} C_{F|N} \cup (\cup\{C_{F|F'} \mid F, F' \in \mathcal{F} \text{ and } F \neq F'\})$
3. For all $c \in C_{iso}$
        Compute $\tilde{G}_N$ and all $\tilde{G}_F$ $(F \in \mathcal{F})$
        For all $F \in \mathcal{F}$
            If $C_{iso}$ -{c}$\notin$ SD($F|N$)
                go to 3a
            End (If)
            For all $F' \in \mathcal{F}$, $F' \neq F$.
                If $C_{iso}$ -{c}$\notin$ SD($F|F'$)
                    go to 3a
                End (If)
            End (For)
        End (For)
        $C_{iso} := C_{iso} - \{c\}$
3a.  Continue
   End (For)
```

Procedure 5.4: C_{iso} is a minimal sensor set for failure detection and isolation

Example 5.4.2 *(Example 5.3.1 continued): Suppose we start Procedure 5.4 with the following minimal distinguishers:* $C_{F_1|N} = C_{F_2|N} = \{c_1\}$, $C_{F_{1,2}|N} = \{c_1, c_2\}$, $C_{F_1|F_2} = C_{F_2|F_1} = \{c_1\}$, $C_{F_1|F_{1,2}} = C_{F_{1,2}|F_1} = \{c_1, c_3\}$, $C_{F_2|F_{1,2}} = C_{F_{1,2}|F_2} = \{c_1\}$. *After step 2,* $C_{iso}= \{c_1, c_2, c_3\}$. *Next in step 3, first we remove* c_1. *In this case* $C_{iso} - \{c_1\} = \{c_2, c_3\} \notin$ ***SD****(*$F_{1,2}|N$*). So we keep* c_1. *Next we remove* c_2 *from* C_{iso} *and observe that* $C_{iso}-\{c_2\} = \{c_1, c_3\}$ *can be used to distinguish every faulty condition from normal, and every faulty condition from every other faulty condition. Hence,* $\{c_1, c_3\} \in \mathcal{DI}$. *Finally, removing* c_3 *from* $C_{iso} = \{c_1, c_3\}$ *results in* $\{c_1\}$ *which is not an* $F_{1,2}|N$*-distinguisher and as a result,* $C_{iso} = \{c_1, c_3\}$ *is a minimal element of* $\mathcal{DI}$.■

Similar to Procedure 5.3, in Procedure 5.4, in the computation of minimal distinguishers required to start the procedure and in the distinguishability tests in step 3, the sub-generators of two conditions, (G_N and G_F with $F \in \mathcal{F}$, or G_F and $G_{F'}$ with $F, F' \in \mathcal{F}, F \neq F'$) are used at a time. This is because of the tree structure of G (Figure 3.1) and Theorem 5.3.3. If G has up to m simultaneous failures ($m << p$), then G will have $\mathcal{O}(p^m M)$ states, while each single condition sub-generator (G_N and $G_F, F \in \mathcal{F}$) has $\mathcal{O}(M)$ states. Thus Procedure 5.4 shows that taking advantage of the tree structure of G, we can reduce the random-access memory requirement (i.e., space complexity) by a factor of $\mathcal{O}(p^{m-1})$.

Now we find the computational complexity of Procedure 5.4. Let us assume up to m simultaneous failures are possible (with $m << p$). Suppose the minimal distinguishers to start Procedure 5.4 are calculated using Procedure 5.2. Since there are $\mathcal{O}(p^m)$ conditions, there will be $\mathcal{O}(p^{2m})$ minimal distinguishers to compute and therefore, the computational complexity of finding the required minimal distinguishers is $\mathcal{O}(n_sp^{2m}M^4 + n_sp^{2m}n_eM^3)$. Step 2 in Procedure 5.4 can be performed in $\mathcal{O}(p^{2m}n_s)$. Step 3 executes $\mathcal{O}(n_s)$ times. Each time, it computes the RTSs $\tilde{G}_N$ and $\tilde{G}_F$ ($F \in \mathcal{F}$) which takes $\mathcal{O}(p^m n_e M^3)$, and performs $\mathcal{O}(p^{2m})$ distinguishability tests which can be performed in $\mathcal{O}(p^{2m}(M^4 + n_eM^2))$ time. Thus Step 3 is done in $\mathcal{O}(n_sp^{2m}M^4 + n_en_s(p^{2m}M^2 + p^mM^3))$ time. Therefore, the computational complexity of Procedure 5.4 is $\mathcal{O}(n_sp^{2m}M^4 + n_en_sp^{2m}M^3)$. Assuming n_e and M increase linearly and exponentially respectively with the number of system components, the complexity of Procedure 5.4 becomes $\mathcal{O}(n_sp^{2m}M^4)$.

Remark 5.4.1 *The procedures proposed in this chapter that are based on minimal distinguishers, take the tree structure of the plant G into account. Suppose we do not take advantage of the tree structure and use Theorem 5.1.1 (Diagnosability test) and Procedure 5.1 (with $\mathcal{D}$ in step 2a replaced by $\mathcal{DI}$) to find a minimal sensor set for failure detection and isolation. Assume simultaneous occurrence of up to m failure modes. Let us find the computational complexity of Theorem 5.1.1. The deadlock states in condition (1) can be examined and compared with states in $X - X_{\mathcal{F}_i}$ in $\mathcal{O}(|X_{\mathcal{F}_i}| + |X - X_{\mathcal{F}_i}|) = \mathcal{O}(p^mM)$. The cycles with constant outputs in $X_{\mathcal{F}_i}$ (condition(2)) can be computed and their outputs can be compared with the outputs of states in $X - X_{\mathcal{F}_i}$ in $\mathcal{O}(|X_{\mathcal{F}_i}| + |\theta_{\mathcal{F}_i}| + |X - X_{\mathcal{F}_i}|) = \mathcal{O}(p^mM + n_ep^{2m-2}M^2)$, where $\theta_{\mathcal{F}_i}$ is the set of transitions of $G_{\mathcal{F}_i}$ (Note that $G_{\mathcal{F}_i}$ has $\mathcal{O}(p^{m-1}M)$ states and $\mathcal{O}(n_ep^{2m-2}M^2)$ transitions.) $\tilde{G}_{N,\bar{\mathcal{F}}_i}$ and $\tilde{G}_{\mathcal{F}_i}$ have $\mathcal{O}(p^mM)$ and $\mathcal{O}(p^{m-1}M)$ states, and $\mathcal{O}(p^{2m}M^2)$ and $\mathcal{O}(p^{2m-2}M^2)$ transitions. Thus, condition (3) can be verified in $\mathcal{O}(p^{4m-2}M^4)$ time. This gives a complexity of $\mathcal{O}(p^{4m-2}M^4 + n_ep^{2m-2}M^2)$ for testing diagnosability of F_i using Theorem 5.1.1.*

In Procedure 5.1, the main loop is executed $\mathcal{O}(n_s)$ *times. In each loop* $\tilde{G}$ *is computed in* $\mathcal{O}(p^m M(p^m M + n_e p^{2m} M^2)) = \mathcal{O}(n_e p^{3m} M^3)$*. Also diagnosability of failure modes* $F_1, \ldots, F_n$ *is tested (in the worst case). Thus, the computational complexity of Procedure 5.1 is* $\mathcal{O}(n_s n_e p^{3m} M^3 + n_s p(p^{4m-2} M^4 + n_e p^{2m-2} M^2)) = \mathcal{O}(n_s p^{4m-1} M^4 + n_e n_s p^{3m} M^3)$*. Assuming* n_e *and* M *increase linearly and exponentially (respectively) with the number of system components, the complexity of using Procedure 5.1 for finding minimal sensor set for failure detection and isolation will be* $\mathcal{O}(n_s p^{4m-1} M^4)$*. Comparing this figure with the complexity of Procedure 5.4, shows that taking the tree structure of the plant into account reduces the computational complexity by a factor of* $\mathcal{O}(p^{2m-1})$*.*

Another reason for this reduction in computations is that the proposed method based on minimal distinguishers avoids some repetitive operations in diagnosability tests following Theorem 5.1.1. For instance, suppose $\mathcal{K} = \{N, F_1, F_2, F_{1,2}\}$*. To examine diagnosability of failure mode* F_1 *(following Theorem 5.1.1), we have to compose the output cycles in* F_1 *and* $F_{1,2}$ *with those in* N *and* F_2*. Then for verifying diagnosability of failure mode* F_2*, we compare the output cycles of* F_2 *and* $F_{1,2}$ *with those in* N *and* F_1*. Note here for example, the cycles of* N *and* $F_{1,2}$ *are compared twice. In verifying diagnosability using distinguishers, the output cycle of* $F_{1,2}$ *and* N *are compared once in examining the distinguishabilit of* $F_{1,2}$ *from* N*.* ■

Remark 5.4.2 *The computational complexity of verifying diagnosability using the method in [31] is* $\mathcal{O}(|X|^4 |\Sigma_0| \cdot p)$ *where* Σ_0 *is the set of observable events. Thus* $|\Sigma_0| \leq n_e$*. This gives a complexity of* $\mathcal{O}(n_e |\Sigma_0| \cdot p \cdot |X|^4) = \mathcal{O}(n_e |\Sigma_0| p^{4m+1} M^4)$ *for minimal sensor selection following [53]. In this case, we see that using minimal distin-*

guishers has reduced the computations by $\mathcal{O}(p^{2m+1})$ (n_s and $n_e|\Sigma_0|$ are not considered in comparison). ■

Remark 5.4.3 *Unlike the existing methods for sensor selection for fault detection and isolation (e.g., Procedure 5.1) where the entire sensor set is computed together by applying diagnosability tests on the whole plant, the algorithms proposed in this chapter break down the problem into smaller problems of finding minimal sets for distinguishing one condition from another. Next, the solution for sensor selection problem is obtained by combining these smaller solutions. Our studies show that the minimal distinguishers ($F|N$-distinguishers and $F|F'$-distinguishers) typically have only one or two sensors. The computation of these small sets can be speeded up using heuristics and expert knowledge. For example, a minimal set for distinguishing a "stuck-closed" failure of a valve from normal operation will likely include a flow-meter or pressure sensor near the valve. This shows how heuristics and expert knowledge may be incorporated into the algorithm for sensor selection.* ■

Example 5.4.3 *Consider the ozone generation plant described in Example 3.0.1 in Chapter 3. For brevity, we consider only the following three failure modes: V_1 stuck-closed (F_1), power supply unit (PSU) failed (F_2) and V_3 stuck-closed (F_3). Furthermore, we will assume single-failure scenario (no simultaneous failures).*

We would like to obtain a minimal set of sensors for failure detection and isolation (a minimal element of $\mathcal{DI}$). To use Procedure 5.4, we need a set of minimal distinguishers. To distinguish F_1 (valve V_1 stuck-closed) from N, we pick the flow meter C_{f1} and verify that $\{C_{f1}\}$ is a minimal $F_1|N$-distinguisher (Note that here we have used intuition along with a bottom-up approach to find a minimal distinguisher.) To *distinguish the failure of PSU (F_2) from normal behavior, we choose concentration analyzer and see that $\{C_c\}$ is a minimal $F_2|N$-distinguisher. $\{C_c\}$ is also a minimal $F_3|N$-distinguisher since if the flow of cooling water stops, the temperature in ozone generator rises, resulting in the decomposition of ozone. Furthermore, we can verify that the flow meter C_{f1} can be used to distinguish F_1 from F_2 or F_3 (and vise versa) and thus $\{C_{f1}\}$ is a minimal $F_1|F_2$, $F_1|F_3$, $F_2|F_1$ and $F_3|F_1$ -distinguisher. Now C_c can not be used to distinguish F_2 from F_3, or vice versa, since F_2 and F_3 both result in drop in ozone concentration. To distinguish the failure V_3 stuck-closed (F_3) from F_2 (PSU failure) we consider flow meter C_{f2} and verify that $\{C_{f2}\}$ is a minimal $F_3|F_2$ and $F_2|F_3$ -distinguisher. Now following Procedure 5.4, we obtain $\{C_{f1}, C_{f2}, C_c\}$ as a minimal set of sensors for failure detection and isolation.* ■

5.5 Summary

In this chapter, we show that testing failure diagnosability and sensor selection for diagnosis of permanent failures in discrete-event systems can be broken down into a set of smaller problems involving testing diagnosability of one system condition from another and computing minimal sensor sets that guarantee distinguishability (minimal distinguishers). This approach takes the structure of the system into account and reduces the complexity of testing diagnosability and sensor selection.

In the following chapter, we will see how the proposed algorithms based on minimal distinguishers can be used in sensor selection in a multi-resolution fault diagnosis system. Also, we will discuss how the use of minimal distinguishers can help with reducing the online computations required for reconfiguration of the diagnosis system.

Chapter 6

MULTI-RESOLUTION FAULT DIAGNOSIS

In this chapter, we present our framework for multi-resolution diagnosis and then discuss various design issues, in particular, sensor selection. For sensor selection, we adopt the procedures developed in Chapter 5 based on minimal distinguishers. An illustrative example is also provided. Finally, we discuss the reconfiguration of the diagnosis system online (when the plant is operational).

6.1 Structure of Multi-Resolution Fault Diagnosis

As discussed in Chapter 1, hierarchical approaches are among the methods to reduce the computational complexity in failure detection and isolation of discrete-event systems. In this thesis, we introduce an algorithm for failure diagnosis which uses a sequence of models of the plant, with increasing resolutions, to narrow down the range of possible diagnosis step by step and to finally isolate the failure. In this way, the original problem of failure diagnosis is broken down into a sequence of simple problems. This approach is similar to Branch and Bound techniques used in

Operations Research. In this section, we describe the proposed **Multi-Resolution Diagnosis** techniques. Some design issues (such as failure grouping, sensor selection) and performance issues (such as diagnosability and diagnosis delay) will be discussed in future sections.

Suppose the plant can be modelled a nondeterministic finite-state Moore automaton $G = (X, \Sigma, \delta, x_0, Y, \lambda)$ as described in Chapter 3. For now we assume all failure modes are permanent and we consider single-failure scenario (i.e., no simultaneous occurrence of two or more failures). As mentioned in Remark 3.0.1 in the case of nonpermanent failures, we can convert them into equivalent problem associated with all permanent failures. Simultaneous occurrence of failures will be discussed in Section 6.4. Let us assume that the failure conditions are grouped into l failure groups $\mathcal{F}^{(1)}, \mathcal{F}^{(2)}, \ldots, \mathcal{F}^{(l)}$. Failure conditions can be grouped, for instance, based on the subsystems that the failures occur. In other words, $\mathcal{F}^{(1)}$ contains the faulty conditions that may develop in subsystem 1 (and so on). Failure grouping will be discussed further in Section 6.6. Let $F_1^{(i)}, \ldots F_{p_i}^{(i)}$ denote the faulty conditions in group $\mathcal{F}^{(i)}$: $\mathcal{F}^{(i)} = \{F_1^{(i)}, \ldots, F_{p_i}^{(i)}\}$ $(1 \leq i \leq l)$. Therefore, the set of failure conditions and groups can be represented in the form of a hierarchy as shown in Fig. 3.2. Here $\mathcal{F} = \cup_{i=1}^{l} \mathcal{F}^{(i)} = \{F_1^{(1)}, \ldots, F_{p_1}^{(1)}, F_1^{(2)}, \ldots, F_{p_l}^{(l)}\}$ is the set of faulty conditions. In Fig. 3.2, $\mathcal{N} = \{N\}$, where N is the normal condition. The multi-resolution diagnosis proposed in this thesis is designed based on grouping of conditions in Fig. 3.2. Note that the three-level failure hierarchy in Fig. 3.2 can be replaced with hierarchies having more than three levels. In this dissertation, for simplicity, we present our results based on the three-level hierarchy in Fig. 3.2.

In the multi-resolution diagnosis system proposed here, a diagnoser is designed to detect faulty operation. In other words, the diagnoser determines whether the condition of plant is $\mathcal{N}$ or $\mathcal{F}$. Let $\mathbb{K}_1 = \{\mathcal{N}, \mathcal{F}\}$ denote the **level-one condition set**. Once a faulty behavior is detected, another diagnoser is used to identify the faulty group the plant's condition belongs to. We shall refer to this diagnoser and the set $\mathbb{K}_2 = \{\mathcal{F}^{(1)}, \ldots, \mathcal{F}^{(l)}\}$ as the second-level diagnoser and condition set. Once, the faulty group is identified as, say $\mathcal{F}^{(i)}$, then a third-level diagnoser is invoked to isolate the faulty condition of the plant. Thus, $\mathbb{K}_3 = \mathcal{F}^{(i)}$ $(1 \leq i \leq l)$ will be the third-level condition set. In the following, we will examine the design and operation of the above-mentioned diagnosers in more detail. For simplicity, we will assume $l = 3$ faulty groups, $p_1 = 3$, $p_2 = 2$, and $p_3 = 1$. Thus the plant condition set will be $\mathcal{K} = \{N, F_1^{(1)}, F_2^{(1)}, F_3^{(1)}, F_1^{(2)}, F_2^{(2)}, F_1^{(3)}\}$.

In the first step of multi-resolution diagnosis, we detect faulty behavior. The level-one diagnoser has a structure similar to that described in Section 2.2 of Chapter 2. In other words, the diagnoser is an observer which finds an estimate for system state and the corresponding condition estimate (which will be a subset of level-one condition set $\mathbb{K}_1 = \{\mathcal{N}, \mathcal{F}\}$). To reduce the size of diagnoser, we design it based on reduce RTS (Section 2.1 of Chapter 2). Since at this level, the objective is only to detect faulty behavior (and not to isolate the fault), we can use a minimal subset of the sensor set C_{tot}, say C_1. Efficient algorithms for sensor selection for multi-resolution fault diagnosis will be discussed in Section 6.2. Let λ_1 denote the output map, assuming sensors in C_1 are used. To construct the reduced RTS, we find the coarsest partition of X compatible with transition in the RTS that is finer than $\ker \lambda_1 \wedge \ker \kappa_1$ where

$\kappa_1 : X \to \mathbb{K}_1$ is the **level-one condition map**, "ker" refers to the equivalence kernel of the corresponding map [23] and $\wedge$ denotes the meet operation in the lattice of equivalence relations [64]. Let us call the corresponding reduced RTS $\bar{G}_1$ and denote the state set of $\bar{G}_1$ as $\bar{X}_1$. Note that since $\ker \lambda \leq \ker \lambda_1$ (where λ is the output map assuming all sensors in C_{tot} are used) and $\ker \kappa \leq \ker \kappa_1$ (where $\kappa : X \to \mathcal{K}$ is the condition map used in a diagnoser that solves failure detection and isolation in single step as in [23]), the reduced RTS $\bar{G}_1$ in first level diagnosis is likely to have fewer states than the reduced RTS $\bar{G}$ (in [23]) based on $\ker \lambda \wedge \ker \kappa$ (In the worst case, $\bar{G}_1$ and $\bar{G}$ have the same number of states.). Intuitively, detecting faulty behavior is easier than detecting *and isolating* failures. Therefore, for the detection problem a more simplified model of the plant (with less solution) and fewer sensors may be enough. Let $z_{1,k} \subseteq \bar{X}_1$ and $\kappa_1(z_{1,k}) \subseteq \mathbb{K}_1 = \{\mathcal{N}, \mathcal{F}\}$ be the state and condition estimates provided by the first level diagnoser after kth output readings. The initial state of level-one diagnoser $z_{1,o} \subseteq \bar{X}_1$ depends on the information about the plant state at the time the diagnosis system is started. If a failure occurs and as soon as $\kappa_1(z_{1,k}) = \{\mathcal{F}\}$ (for some k), the faulty behavior is detected, the second-level diagnoser is used to find the group the faulty condition belongs to. Let us denote the last state estimate provided by level-one diagnoser as $z_{1,f}$.

Once the faulty behavior is detected, the second level diagnoser is started to isolate the fault group. The second-level diagnoser is an observer which provides an estimate for the system state and thus condition. The condition estimate will be a subset of the second-level condition set $\mathbb{K}_2 = \{\mathcal{F}^{(1)}, \mathcal{F}^{(2)}, \mathcal{F}^{(3)}\}$. Similar to the first-level failure diagnosis, instead of C_{tot}, we only use a minimal subset of sensors, denoted here by C_2, that provides sufficient information for isolating fault group. To reduce the size of the diagnosis system, instead of RTS of $G_{\mathcal{F}}$ (the faulty subgenerator of G), the reduced RTS of $\bar{G}_{\mathcal{F},2}$ is used. Let $\bar{X}_{\mathcal{F},2}$ denote the state set of $\bar{G}_{\mathcal{F},2}$. The second-level diagnoser is initialized with state estimate $z_{2,0} = P_2 P_1^{-1} z_{1,f}$, where $z_{1,f}$ is the final state estimate provided by the first-level diagnoser; $P_1 : X \longrightarrow \bar{X}_1$ and $P_2 : X_{\mathcal{F}} \longrightarrow \bar{X}_{\mathcal{F},2}$ are the natural projections used in reducing RTSs in levels one and two, respectively. In this way, the information obtained by the first-level diagnoser is passed to the second-level diagnoser. Let λ_2 and $\kappa_2 : X_{\mathcal{F}} \longrightarrow \mathbb{K}_2$ denote the output and condition maps of the second-level. Thus $\ker \lambda_{|X_{\mathcal{F}}} \leq \ker \lambda_2$ (where $\lambda_{|X_{\mathcal{F}}}$ is the restriction of λ to $X_{\mathcal{F}}$) since at the second level diagnosis, a subset of sensors is used for isolating fault group. Furthermore, $\ker \kappa_{|X_{\mathcal{F}}} \leq \ker \kappa_2$ (where $\kappa_{|X_{\mathcal{F}}}$ is the restriction of κ to $X_{\mathcal{F}}$). Therefore, the reduced RTS $\bar{G}_{\mathcal{F},2}$ used in the second-level diagnosis will have fewer states than the reduced RTS $\bar{G}$ (or its subgenerator $\bar{G}_{\mathcal{F}}$) used in standard solution to failure detection and isolation. Intuitively, isolating a fault group is easier than isolating the specific fault condition and therefore a simplified model with less resolution should be enough.

Finally, at the third level, $l = 3$ diagnosers are needed, one for isolating the fault condition in each fault group. Let us suppose the second-level diagnoser isolates the fault group $\mathcal{F}^{(1)}$ and $z_{2,f}$ is the last state estimate provided by the second-level diagnoser. Now, the third-level diagnoser for isolating faults in $\mathcal{F}^{(1)}$ is started. For the purpose of fault isolation in $\mathcal{F}^{(1)}$, a subset of sensors C_3^1 is used along with the corresponding reduced RTS $\bar{G}_{\mathcal{F}^{(1)},3}$. The diagnoser is initialized with $z_{3,0}^{(1)} = P_3^{(1)} P_2^{-1} z_{2,f}$ where $P_3 : X_{\mathcal{F}_1} \longrightarrow \bar{X}_3^{(1)}$ is the natural projection used in obtaining $\bar{G}_{\mathcal{F}^{(1)},3}$

and $X_{\mathcal{F}_1}$ denotes state set of $G_{\mathcal{F}^{(1)}}$. Similar to the case of first and second level diagnosers, this diagnoser will have fewer states than a standard diagnoser for fault detection and isolation. The model reduction algorithm can particularly work very efficiently if fault partitioning is done properly. For instance, suppose G has three subsystems $G = G^{(1)}||G^{(2)}||G^{(3)}$. If each fault group corresponds to faults in each subsystem (i.e., $\mathcal{F}^{(i)}$ are faults in $G^{(i)}$), then for instance, for isolating faults in $\mathcal{F}^{(1)}$ (subsystem $G^{(1)}$), very simplified models of $G^{(2)}$ and $G^{(3)}$ may be enough. In fact, if $G^{(1)}$ and $G^{(2)}||G^{(3)}$ do not have any common unobservable events [1], state estimate in $G^{(1)}$ does not require the models of $G^{(2)}$ and $G^{(3)}$ at all (Proposition 3 in [50]). In this case, the multi-resolution approach works extremely efficiently.

In the following sections, we will discuss design issues such as sensor selection, and fault grouping. Extension of the proposed multi-resolution diagnosis scheme to the case of simultaneous faults will also be discussed.

6.2 Sensor Selection for Multi-Resolution Diagnosis

In this section, we discuss the issue of sensor selection for Multi-Resolution failure Diagnosis (**MRD**). The objective of sensor selection, for each level in multi-resolution approach, is to select a minimal set of sensors from the given sensor set which ensures that the desired goal (fault detection or diagnosis) can be met. To realize this, we present two sets of algorithms based on minimal distinguishers. The first set will be presented in this section. The second set of algorithms (which will be discussed

[1] Here it is assumed that the initial state estimate z_0 is in the form of $z_0 = z_0^{(1)} \times z_0^{(2)} \times z_0^{(3)}$ with $z_0^{(i)}$ is the initial state of subsystem $G^{(i)}$.

at the end of this chapter) is particularly suitable when we would like to be able to reconfigure the diagnosis system.

We take the same hierarchy as in Fig. 3.2 and assume n_s sensors in the system: $C_{tot} = \{c_1, \dots, c_{n_s}\}$. Suppose $\mathcal{F}^{(1)}, \dots, \mathcal{F}^{(l)}$ are the groups of failure conditions (i.e., blocks or failure partition) in the system. We say the **failure group isolation** problem is solvable if for failure group $\mathcal{F}^{(i)}$, assuming the diagnoser (observer of the system) initialized with $z_0 = X_{\mathcal{F}}$, there exists an integer $N_i \geq 0$ such that after the occurrence of a failure leading to a condition in $\mathcal{F}^{(i)}$ and initialization of the diagnoser, the diagnoser reaches a $z_k \in X_{\mathcal{F}^{(i)}}$ after the occurrence of at most N_i events in the system. Similarly, we say **failure isolation** for the group $\mathcal{F}^{(i)}$ is solvable if for any failure $F_j^{(i)}$ if the diagnoser of the system is started with $z_0 = X_{\mathcal{F}^{(i)}}$, there exists an integer $N_i \geq 0$ such that after initialization of the diagnoser, the diagnoser reaches $z_k \in X_{F_j^{(i)}}$ after the occurrence of at most N_i events in the system.

Let $\mathcal{D}$, $\mathcal{GI}$ and $\mathcal{FI}(\mathcal{F}^{(i)})$ denote the set of sensor sets for which the problems of fault detection, failure group isolation and failure isolation for group $\mathcal{F}^{(i)}$ are solvable, respectively. Therefore, $\mathcal{D} = \{C | C \subseteq C_{tot}$ and fault detection using sensors C is solvable$\}$, $\mathcal{GI} = \{C | C \subseteq C_{tot}$ and fault group isolation using sensors C is solvable$\}$ and $\mathcal{FI}(\mathcal{F}^{(i)}) = \{C | C \subseteq C_{tot}$ and fault isolation for group $\mathcal{F}^{(i)}$ using sensors C is solvable$\}$. Thus, $\mathcal{D}$, $\mathcal{GI}$ and $\mathcal{FI}(\mathcal{F}^{(i)}) (1 \leq i \leq l)$ are the set of sensor selections for first-level, second-level and third-level (lowest level) in multi-resolution diagnosis.

According to Theorem 5.3.1, $\mathcal{D} = \cap_{F \in \mathcal{F}} \mathbf{SD}(F|N)$. Similarly we have the following results.

Theorem 6.2.1 $\mathcal{GI} = \cap\{\boldsymbol{SD}(F|F')|\exists i, j, i \neq j, 1 \leq i \leq l, 1 \leq j \leq l; F \in \mathcal{F}^{(i)}, F' \in \mathcal{F}^{(j)}\}$

Proof The proof follows from Theorem 5.3.2. We omit the details for brevity. ■

Theorem 6.2.2 $\mathcal{FI}(\mathcal{F}^{(i)}) = \cap\{\boldsymbol{SD}(F|F')\ |F, F' \in \mathcal{F}^{(i)}$ *and* $F \neq F'\}$

Proof Similarly, the proof follows from Theorem 5.3.2 and it's straightforward. We omit the proof for brevity. ■

Our objective in this section is to find minimal elements of $\mathcal{D}$, $\mathcal{GI}$ and $\mathcal{FI}(\mathcal{F}^{(i)})$. The minimal elements of these sets are in general not unique. We shall propose procedures for finding one element for each set. In the rest of the chapter, $C_{\mathcal{F}|\mathcal{N}}, C_{giso}$ and $C_{iso}^{(i)}$ will denote the minimal elements obtained using the proposed procedures.

6.2.1 First Algorithm for Sensor Selection

Given the set of sensors C_{tot}, for each pair of system conditions A and A', a minimal $A|A'$-distinguisher can be obtained according to Procedure 5.2.

For computing a minimal sensor set for the fault detection problem, we can use Procedure 5.3 in Chapter 5. This gives $C_{\mathcal{F}|\mathcal{N}}$ (first-level sensor selection).

Next in order to perform sensor selection for second-level fault diagnosis (finding a minimal element for $\mathcal{GI}$), in the procedure for failure detection and isolation (Procedure 5.4), we substitute failure conditions $F, F' \in \mathcal{F}$ with failure groups $\mathcal{F}^{(i)}, \mathcal{F}^{(j)} \in \mathcal{F}$ and remove $\cup_{F \in \mathcal{F}} C_{\mathcal{F}|\mathcal{N}}$ from Step 2. Then the result will be a procedure for sensor selection for failure group isolation (Procedure 6.1). Briefly, in Procedure 6.1, Step 2 calculates an element of $\mathcal{GI}$ (using Theorem 6.2.1), and Step 3 reduces this element to a minimal element of $\mathcal{GI}$.

Procedure 6.1: Given a sensor set $C_{tot} = \{c_1, \ldots, c_{n_s}\}$ and minimal $F|F'$-distinguishers $C_{F|F'}$ for every $F \in \mathcal{F}^{(i)}, F' \in \mathcal{F}^{(j)}$ and $\mathcal{F}^{(i)} \neq \mathcal{F}^{(j)}$.

```
1. Initialization: $C_{giso} := \varnothing$
2. $C_{giso} := \cup \{ C_{F|F'} \mid \exists\, i, j, i \neq j, 1 \leq i \leq l, 1 \leq j \leq l;\ F \in \mathcal{F}^{(i)}$ and $F' \in \mathcal{F}^{(j)} \}$
3. For all $c \in C_{giso}$
        Compute all $\widetilde{G}_{\mathcal{F}^{(i)}}$ $(1 \leq i \leq l)$
            For all $F \in \mathcal{F}^{(i)}, F' \in \mathcal{F}^{(j)}$ with $i \neq j$
              If $C_{giso} - \{c\} \notin$ SD$(F|F')$
                  go to 3a
              End (If)
        End (For)
        $C_{giso} := C_{giso} - \{c\}$
3a      Continue
   End (For)
```

Procedure 6.1: C_{giso} is a minimal sensor set for second-level fault diagnosis.

Similarly, in order to obtain the sensor selection for fault isolation within a group $\mathcal{F}^{(i)}$, we substitute $\mathcal{F}$ in Procedure 5.4 with $\mathcal{F}^{(i)}$, and remove $\cup_{F \in \mathcal{F}} C_{F|N}$ in Step 2. Then the result will be a procedure for sensor selection for isolation within fault group $\mathcal{F}^{(i)}$ (third-level fault diagnosis).

Note that the minimal $F|F'$ distinguishers that are calculated for sensor selection at the third-level are also used for sensor selection at the second-level. In other words, minimal distinguishers for fault groups need not be calculated. This is one of the advantages of our sensor selection method.

6.3 Example: Multi-Resolution Failure Diagnosis in Ozone Generator System

In Section 6.1, we have presented our procedure for multi-resolution fault diagnosis (MRD). In this section, we study the design of a multi-resolution fault diagnosis system for the ozone generation plant introduced in Example 3.0.1 (Chapter 3). Here is the outline of the design procedure.

Multi-Resolution Diagnosis Procedure:

1. Group failure modes, say based on the subsystem each occurs in (Also, see Remark 6.6.1.
2. Compute minimal sensor sets $C_{\mathcal{F}|\mathcal{N}}$, C_{giso} and $C_{iso}^{(i)}$ $(1 \leq i \leq l)$ for the three levels of diagnosis (using procedures in Section 6.2.).
3. Compute reduced RTS $\bar{G}_1$ assuming only sensors in $C_{\mathcal{F}|\mathcal{N}}$ are used.
4. Compute reduced RTS $\bar{G}_{\mathcal{F},2}$ (assuming only sensors in C_{giso} are used for group isolation).
5. Compute reduced RTSs $\bar{G}_{\mathcal{F}^{(i)},3}$ $(1 \leq i \leq l)$ (based on sensors $C_{iso}^{(i)}$).

Now we consider the ozone generation plant introduced in Example 3.0.1 (Chapter 3) as an illustrative example. For brevity, we only consider failures in the components of the oxygen/ozone pipe and the cooling water pipe, in other words: V_1 stuck-closed (F_1), V_2 stuck-closed (F_2), V_2 stuck-open (F_3), V_3 stuck-closed (F_5) and V_3 stuck-open(F_6). We assume that the information about the state of controller (Fig. 3.6) is available for diagnosis. It can be shown that the flow meters c_{f1}, c_{f2} and concentration sensor c_c form a minimal set of sensors to ensure the diagnosability of all failure modes (i.e., $\{c_{f1}, c_{f2}, c_c\}$ is a minimal element of $\mathcal{DI}$). Therefore the output map can be considered to be of the form $\lambda = \lambda_c \times \lambda_s$ where $\lambda_c : X_c \rightarrow X_c$ provides the state of controller. $X_c = \{C_0, \ldots, C_7\}$ is the state set of controller and $\lambda_c(C_i) = C_i$ $(0 \leq i \leq 7)$. Furthermore, $\lambda_s : X \rightarrow \{ngf, gf\} \times \{nf, f\} \times \{al, a\}$ provides the output of sensors in the order c_{f1}, c_{f2} and c_c, where "ngf"and "gf" correspond to "no gas flow" and "normal gas flow", "nf" and "f" mean "no cooling water flow" and 'normal cooling water flow" and "al" and"a" imply "low ozone concentration" and "normal ozone concentration", respectively.

The plant model [2] is shown in Fig. 6.1. In each state, the state name and output are shown. The events are not shown on the figure to avoid clutter.

Let us start with the design of the first-level diagnoser. It can be shown (for instance, following Procedure 5.3) that $C_{\mathcal{F}|\mathcal{N}} = \{c_{f1}, c_{f2}\}$ is a minimal $\mathcal{F}|\mathcal{N}$-distinguisher. Specifically, c_{f1} detects faulty behaviors due to F_1, F_2 and F_3 and c_{f2} detects faulty behaviors due to F_5 and F_6. Furthermore, to detect failure in valves using the flow meters, the knowledge of PSU running or not is not important. Therefore the output λ_c for controller states can be replaced by a coarser map $\lambda_{c_1} : X_c \rightarrow X_c$ such that

[2] This model is similar to the model in [62], except for two changes. In our system, we assume that the controller command which stops the power supply unit after the power supply unit has been run for a certain minimum time (allowing ozone to be generated). Also we assume that when the oxygen gas inlet valve is stuck closed, there is remaining oxygen gas in the gas pipe for a short period. Under these circumstances, in state 14, when V_1 is stuck-closed, V_2 and V_3 are open and the PSU is running, the concentration of ozone may rise (state transition 14→ 18) but due to reduction of pressure in ozone generator, the concentration of ozone falls rapidly (state transition 18→ 14). The shut down process starts with PSU turning off (state transition 14→ 15).

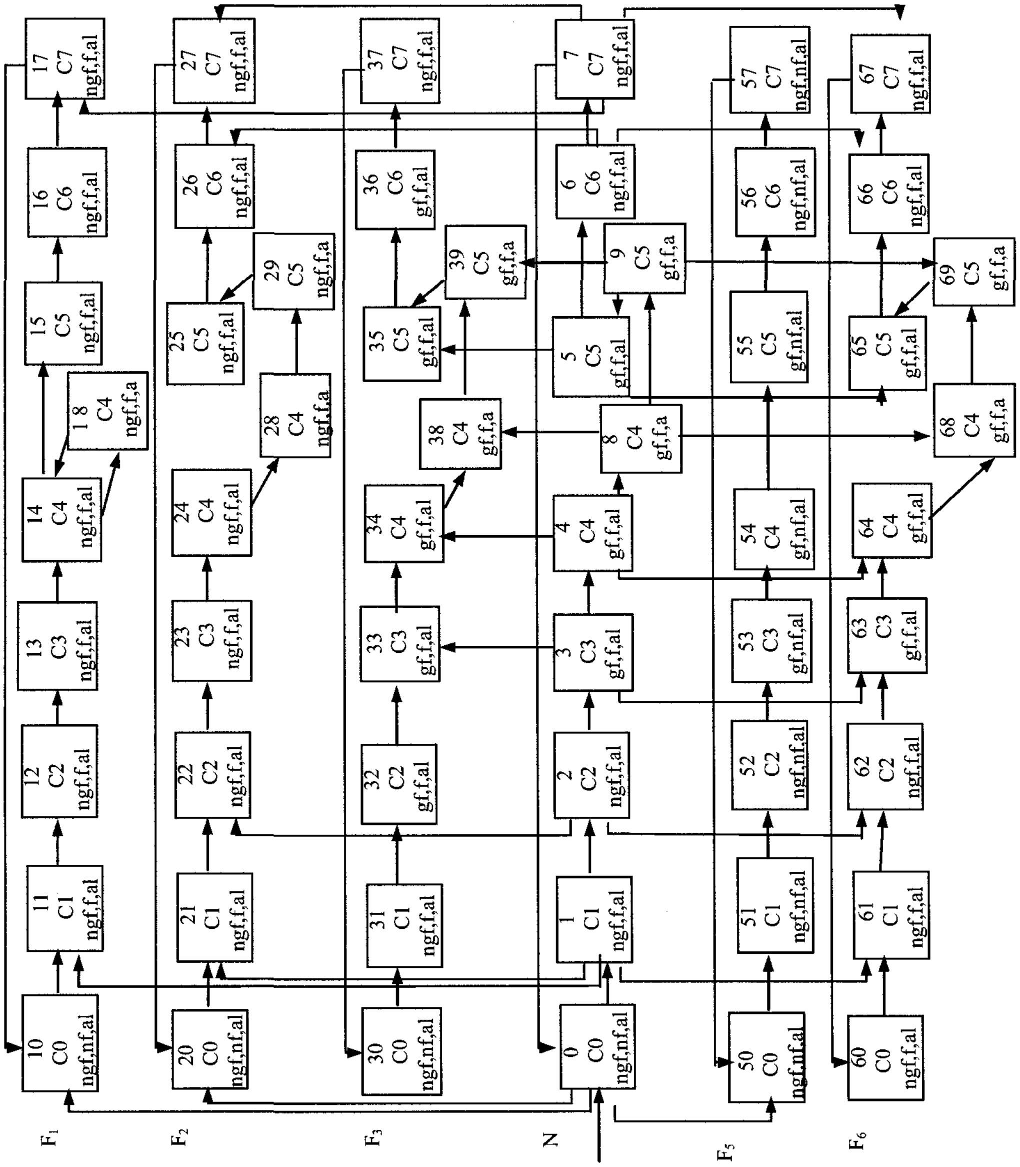


Fig. 6.1. The ozone plant model with five failures.

$\lambda_{c_1}(C_3) = \lambda_{c_1}(C_4) = \lambda_{c_1}(C_5) = C_3$, and for other controller states $\lambda_{c_1}(C_i) = C_i$ [3]. Hence the overall output map will be $\lambda_1 : \lambda_{c_1} \times \lambda_{s_1}$ with $\lambda_{s_1} : X \rightarrow \{ngf, gf\} \times \{nf, f\}$. Of course, at the first-level, the condition map is $\mathbb{K}_1 = \{\mathcal{N}, \mathcal{F}\}$. Based on the aforementioned output and condition sets, the reduced RTS $\bar{G}^1$ for the first-level diagnosis is obtained and shown in Fig. 6.2. In the plant model, states 10 and 20 are equivalent and grouped in a block, designated in $\bar{G}^1$ as state '1-10'. The other equivalent states are grouped similarly. Overall the original number of 57 states is now reduced to 30.

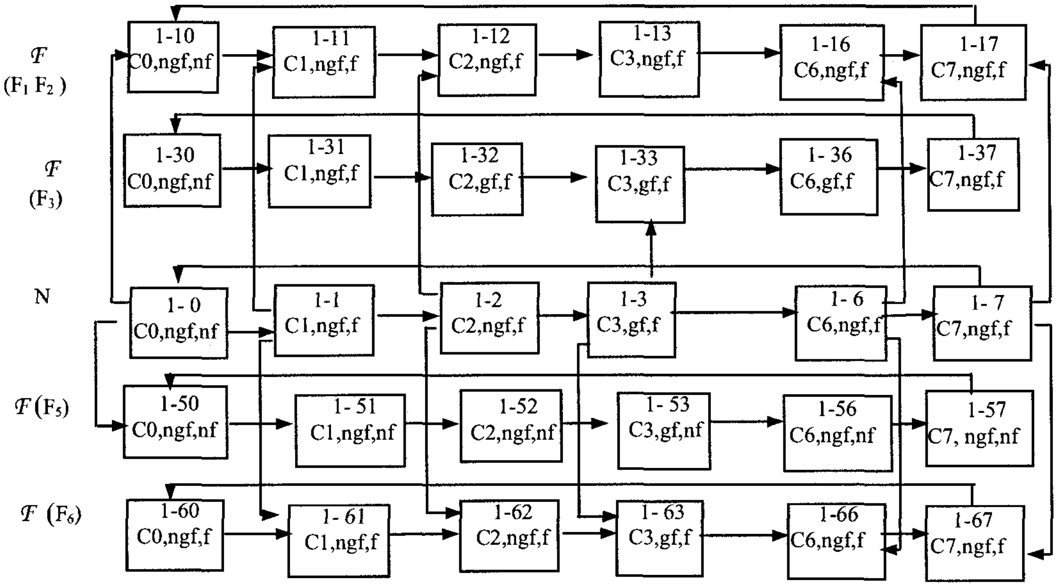


Fig. 6.2. The reduced RTS for first-level diagnosis.

In order to design the second-level diagnoser, we need to group the failure modes. Let us group the failure according to the subsystem they occur in: $\mathcal{F}^{(1)} = \{F_1, F_2, F_3\}$,

[3] The controller output λ_c can be considered as the resulting output of eight "virtual sensors": $\lambda_c = \lambda_c^0 \times \lambda_c^1 \times \ldots \lambda_c^7$ where $\lambda_c^i(C_i) = 1$ and $\lambda_c^i(C_j) = 0$ $(j \neq i)$ for $0 \leq i \leq 7$. "Virtual sensor" λ_c^i generates output 1 only when controller is in state C_i and outputs 0 otherwise. These new sensors can be included in the Procedures for minimal sensor selection, if desired.

$\mathcal{F}^{(2)} = \{F_5, F_6\}$. $\mathcal{F}^{(1)}$ includes the failure modes in the oxygen/ozone pipes and $\mathcal{F}^{(2)}$ includes the failure modes in the cooling water subsystem.

We can use Procedure 6.1 t find a minimal sensor set for group isolation problem. We can verify that $C_{giso} = \{c_{f1}\}$ is such a minimal sensor set. Intuitively, c_{f1} can be used to monitor flow in the oxygen and ozone pipes and detect faults in V_1 and V_2. Absence of fault (in the oxygen and ozone pipe system) would indicate that the fault is in the cooling water system. In this case, sensors c_{f2} and c_c are not used and the information about the commands *Run PSU*, *Stop PSU*, *Open* V_3 and *Close* V_3 are not required for diagnosis. Thus the information about controller states is reported by the map $\lambda_{2,c} : X_c \to X_c$ with $\lambda_{2,c}(C_0) = \lambda_{2,c}(C_1) = \lambda_{2,c}(C_7) = C_0$, $\lambda_{2,c}(C_3) = \lambda_{2,c}(C_4) = \lambda_{2,c}(C_5) = C_3$, $\lambda_{2,c}(C_2) = C_2$ and $\lambda_{2,c}(C_6) = C_6$. Using the above sensor and controller state information, the reduced RTS $\bar{G}_{\mathcal{F},2}$ for the second-level diagnosis is obtained and shown in Fig. 6.3. $\bar{G}_{\mathcal{F},2}$ has only 12 states. $\bar{G}_{\mathcal{F},2}$ of course does not contain any normal states.

Note that the dynamics of the cooling water system (i.e., the operation of V_3) has been essentially removed in $\bar{G}_{\mathcal{F},2}$ by the model reduction algorithm. If the cooling water has more components (say, another valve and a pump), the information about the corresponding opening and closing sequences would have been removed by the model reduction algorithm and $\bar{G}_{\mathcal{F},2}$ would have the same number of states.

For the third-level diagnosis, we have to consider two cases:(i) the failure is in the oxygen /ozone pipe ($\mathcal{F}^{(1)}$), and (ii) the failure is in the cooling water system ($\mathcal{F}^{(2)}$). For case (i), a minimal sensor set is $\{c_{f1}, c_c\}$. Furthermore, information about control

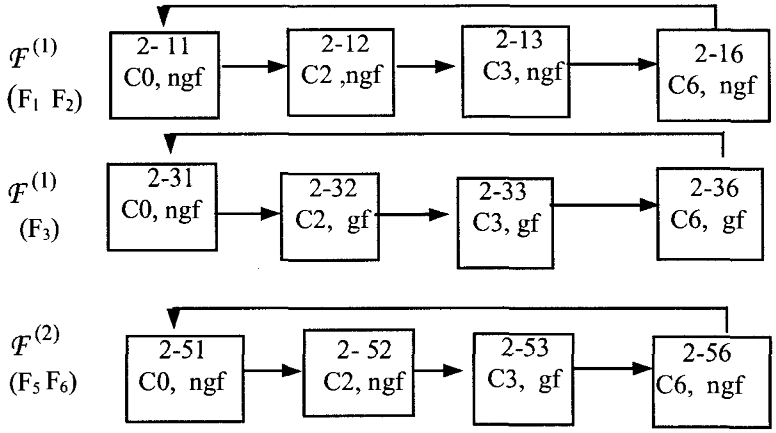


Fig. 6.3. The reduced RTS for second-level diagnosis.

commands *Open* V_3, *Close* V_3, *Open* V_2 and *Close* V_2 is not necessary. The resulting RTS which contains 23 states is shown in Fig. 6.4.

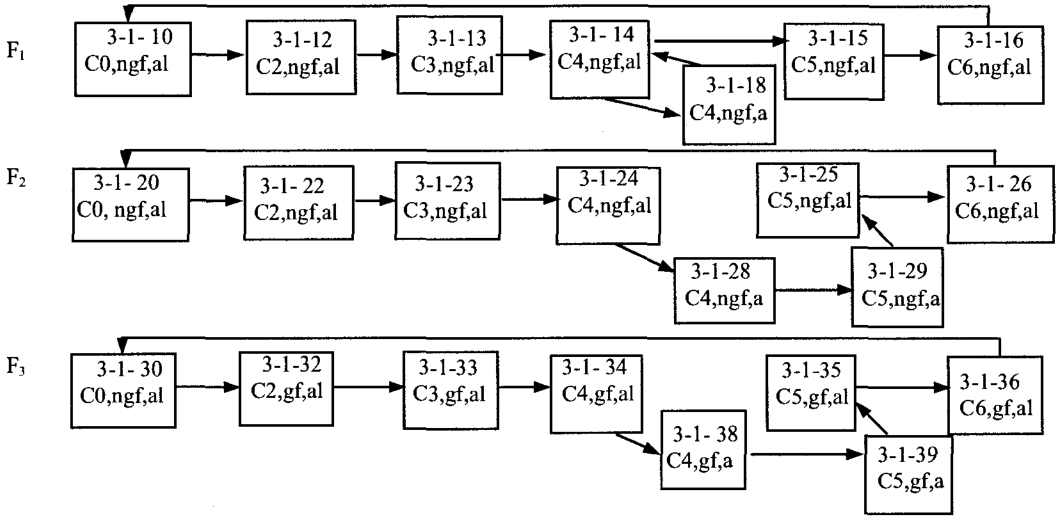


Fig. 6.4. The reduced RTS for third-level diagnosis.

In case (ii), $\{c_{f2}\}$ would be a minimal sensor set for failure isolation within group $\mathcal{F}^{(2)}$, and the information about control commands *Open* V_3 and *Close* V_3 would be necessary. The reduced RTS containing 4 states is shown in Fig. 6.5. In this case, the dynamics of the oxygen/ozone pipe has been removed by the model reduction algorithm.

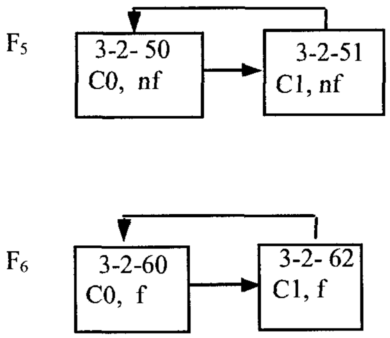


Fig. 6.5. The reduced RTS for third-level diagnosis if $\mathcal{F}^{(2)}$ is isolated.

This example shows how a multi-resolution approach breaks down the diagnosis problem into a sequence of smaller problems and how in the solution of the smaller problems, coarser (simpler) models of the plant can be used. The reduction in the size of models in a given step of diagnosis becomes efficient when the dynamics of a subsystem is either removed or significantly simplified by the model reduction algorithm. We saw instances of this in the second-level and third-level diagnoses in our example.

Now let us examine how the proposed multi-resolution diagnosis system detects and isolates a failure, say V_1 *stuck* $-$ *closed*. Let us assume that V_1 becomes stuck-closed at the beginning of the start-up sequence. Therefore, the plant (Fig. 6.1) follows the sequence $0 \longrightarrow 10 \longrightarrow 11 \longrightarrow \ldots \longrightarrow 17 \longrightarrow 10 \longrightarrow \ldots$. We assume the diagnosis has no initial knowledge of the plant state, and the initial state estimate of the first-level diagnoser is the entire state set in Fig. 6.2 ($z_{1,0} = \bar{X}_1$) and $\kappa_1(z_{1,0}) = \{\mathcal{N}, \mathcal{F}\}$. As the plant goes from 0 to 13 (valves V_1, V_2 and V_3 are opened; V_1 is of course stuck-closed) (Fig. 6.1), the output monitored by the first-level diagnoser will be $(C0, ngf, nf)$, $(C1, ngf, f)$, $(C2, ngf, f)$, $(C3, ngf, f)$. The corresponding state and condition estimates will be

$$z_{1,0} = \bar{X}_1 \qquad \kappa_1(z_{1,0}) = \{\mathcal{N}, \mathcal{F}\}$$

$$z_{1,1} = \{1-1, 1-11, 1-31, 1-61\} \qquad \kappa_1(z_{1,1}) = \{\mathcal{N}, \mathcal{F}\}$$

$$z_{1,2} = \{1-2, 1-12, 1-32, 1-62\} \qquad \kappa_1(z_{1,2}) = \{\mathcal{N}, \mathcal{F}\}$$

$$z_{1,3} = \{1-13\} \qquad \kappa_1(z_{1,3}) = \{\mathcal{F}\}$$

At this point, the faulty behavior is detected and the second-level diagnoser is initialized with $z_{2,0} = P_2 P_1^{-1} z_{1,3} = \{2-13\}$. Since $\kappa_2(z_{2,0}) = \{\mathcal{F}^{(1)}\}$, the failure group is immediately found and the third-level diagnosis is initialized with $z_{3,0} = P_3 P_2^{-1} z_{2,0} = \{3-13, 3-23\}$, and $\kappa_3(z_{3,0}) = \{F_1, F_2\}$. Now in the plant the PSU is started (13 → 14 in Fig. 6.1). After a while the PSU is stopped (14 → 15 in Fig. 6.1). This results in state estimates $z_{3,1} = \{3-1-14, 3-1-24\}$ and $z_{3,2} = \{3-1-15\}$. $\kappa_3(z_{3,2}) = \{F_1\}$ and therefore, F_1 is isolated. In this case, the fact that by the time the PSU is turned off, the ozone concentration has not become high, has led the diagnoser to identify F_1 as the source of malfunction.

6.4 Simultaneous Failures

In the previous discussion, single-failure scenario was assumed. Now we examine the case of simultaneous failures. First we assume a maximum of two failure modes can occur simultaneously. The design of multi-resolution diagnosis system follows the same steps described in Section 6.1, except that this time $\mathcal{F}$ includes simultaneous failure conditions as well. Let us study how the multi-resolution diagnosis system reacts to simultaneous failures.

Let F_1 and F_2 be two failure modes and $F_{1,2}$ the condition when both F_1 and F_2 have occurred. Suppose failure event f_1 occurs and then f_2 occurs and therefore the system condition changes from N to F_1 and finally to $F_{1,2}$. There are two cases that need to be discussed.

First suppose F_1 and $F_{1,2}$ are in the same failure group, say $\mathcal{F}^{(1)}$. In this case, the third-level diagnoser will ultimately (within a bounded number of events) isolate condition $F_{1,2}$.

Now let us suppose F_1 and $F_{1,2}$ belong to two different failure groups. In this case, if f_2 occurs before the second-level diagnoser isolates the failure group, then the second-level diagnoser will correctly identify the group $F_{1,2}$ belongs to. If, on the other hand, f_2 occurs after the second-level diagnoser has identified the group F_1 (say, $\mathcal{F}^{(i)}$) belongs to, since $F_{1,2}$ does not belong to this group ($\mathcal{F}^{(i)}$), either a misdiagnosis occurs (the third-level diagnoser announces a failure condition different from $F_{1,2}$) or an output sequence will be observed that is not consistent with any of the output sequences possible in the failure conditions of the isolated group ($\mathcal{F}^{(i)}$). In this case, if the second-level diagnosis system is reset (with $z_{2,0} = \bar{X}_{\mathcal{F},2}$) and restarted, then the correct fault group and fault condition ($F_{1,2}$) will be identified.

In summary, whenever in multi-resolution diagnosis, the third-level diagnoser isolates a failure condition, the diagnosis should be reset from the second-level in case other failure events have occurred during the diagnosis process. The above discussion also applies to cases when simultaneous occurrence of more than two failure modes are possible. In practice, however, simultaneous occurrence of more than three independent failure modes is very unlikely.

6.5 Properties of Multi-Resolution Diagnosis

In this section, some of the properties of the proposed multi-resolution diagnosis are examined.

6.5.1 Failure Diagnosability

In this chapter, our assumption is that using sensors in C_{tot}, all failure modes are diagnosable and in other words, $C_{tot} \in \mathcal{DI}$. This, in turn, guarantees that all sensors selection problems for the three levels of multi-resolution diagnosis system have solutions (In fact, C_{tot} is a solution for failure detection $\mathcal{D}$, failure group isolation $\mathcal{GI}$, and failure isolation $\mathcal{FI}^{(i)}$). This thus guarantees that all failure modes remain diagnosable using the multi-resolution diagnosis system.

Note that the diagnosis delay (the time from occurrence of failure event to the time the failure is diagnosed) is the sum of diagnosis delays of the first-level, second-level and third-level diagnosers.

6.5.2 Computational Complexity of Design

The design of multi-resolution diagnosis systems consists of two problems: sensor selection and model reduction. The computational (time) complexity of sensor selection problem was discussed in Chapter 5, where we showed that the complexity is of $\mathcal{O}(|X|^4)$ (X is the state set of G). The computational complexity of model reduction using Relational Coarsest Partition (RCP) problem is $\mathcal{O}(|X|^2 \log |X|)$ [5]. Therefore, sensor selection is the most computationally intensive part of design (at least based on worst-case computational scenarios).

It should be noted that the implementation of the multi-resolution diagnosis is in the form of online implementation: the reduced RTS of the three levels are stored in computer memory and used in obtaining state and condition estimates for the plant. The worst-case size of the state space of reduced RTSs is of the same order as the original plant. However, in practice, for instance as shown in the example in Section 6.3, the model reduction techniques based on Relational Coarsest Partition (RCP) problem (bisimulation) can be effective in reducing model size.

6.6 Online Reconfiguration of a Multi-Resolution Diagnosis System

The design of a diagnosis system is based on a given set of sensors. In practice, information from a sensor (or a set of sensors) may become unavailable permanently (due to failures) or temporarily (due to failures or perhaps loss of communication). In such cases, we may wish to reconfigure the diagnosis system so that it relies on the available sensors.

In the case of the multi-resolution diagnosis system proposed in this thesis, the most computationally intensive part of the design is the sensor selection for various levels which involves distinguishability (or diagnosability) tests. In this section, we propose algorithms for sensor selection that do not involve distinguishability tests.

The drawback of these algorithms is that they require the entire sets of minimal distinguishers **SMD**$(F|N)$ (for all $F \in \mathcal{F}$) and **SMD**$(F|F')$ (for all $F, F' \in \mathcal{F}$ with $F \neq F'$).

With n_s sensors, the total number of sensor combinations are $2^{n_s} - 1$. Therefore, there are two issues in using **SMD**s. First, the computation of each **SMD** is exponential in the number of sensors. However, it should be noted that **SMD**s are calculated offline (at the design stage) and not online (when the plant becomes operational). In Section 6.6.1, we will present results that can reduce the computation of **SMD**s (The computational complexity will still remain exponential.). The second issue with the use of **SMD**s is that their cardinality (size) in the worst case is exponential. From a practical point of view, this may not cause serious problem since our observations have shown that minimal distinguishers for failure and normal conditions typically contain one or two sensors. (This will be explained in Section 6.6.1). Therefore, the cardinality of **SMD**$(F|N)$ and **SMD**$(F|F')$ are of order n_s^2.

6.6.1 Calculation of SMDs

In this subsection, we discuss the computation of **SMD**$(F|N)$ $(F \in \mathcal{F})$ and **SMD**$(F|F')$ $(F \neq F', F, F' \in \mathcal{F})$. As mentioned previously, with n_s sensors, there are $2^{n_s} - 1$ sensor combinations and as a result, the cardinalities of the above **SMD**s are exponential in n_s. However, in practice, we notice that in order to distinguish normal behavior from faulty behavior in a faulty condition, one or two sensors are enough. For example, in the ozone generator plant in Example 3.0.1, **SMD**$(F_1|N)$=$\{\{c_{f1}\}, \{c_c\}, \{c_{p1}\},$

$\{c_{p2}\}\}$, and therefore all minimal $F_1|N$-distinguishers contain a single sensor. Hence in this section, we will assume the following.

Assumption 6.6.1 *(i). Each minimal $F|N$-distinguisher ($F \in \mathcal{F}$) consists of one or two sensors; (ii). Each minimal $F|F'$-distinguisher ($F \neq F'$, F and $F' \in \mathcal{F}$) consists of one or two sensors.* ■

In light of the above assumption, in order to find the set of **SMD**$(F|N)$'s and **SMD**$(F|F')$'s, we have to examine single-sensor and double-sensor combinations. The total number of such combinations is $n_s + \frac{n_s(n_s-1)}{2}$, and thus of order n_s^2, which is polynomial. However, in a given problem, we have to verify Assumption 6.6.1 as well. And for this, we need to examine sensor combinations of size 3 to n_s. The number of such sensor combinations is exponential in n_s. In the following, we will present two results that while they do not change the (worst-case) exponential complexity of the verification of Assumption 6.6.1, they reduce the required computations.

Consider two distinct conditions $A, A' \in \mathcal{K}$, $A \neq A'$.

Definition 6.6.1 *A set of minimal $A|A'$-distinguishers $C_1, C_2, \ldots C_k$ are called* ***disjoint*** *if for any C_i and C_j with $i \neq j$ and $1 \leq i, j \leq k$, we have $C_i \cap C_j = \emptyset$.* ■

Proposition 6.6.2 *Suppose a set of minimal $A|A'$-distinguishers has 'a' disjoint minimal $A|A'$-distinguishers. If there exists a minimal $A|A'$-distinguisher C not belonging to the above set, then C cannot contain more than '$n_s - a$' sensors.*

Proof Let $C_1, C_2, \ldots C_a$ be the disjoint minimal distinguishers. If there exists a new minimal distinguisher C, then $C_i - C \neq \emptyset$ $(1 \leq i \leq a)$ (otherwise $C_i \subseteq C$ which violates the assumption). Since C_i's are disjoint, $(C_i - C) \cap (C_j - C) = \emptyset$ $(1 \leq i, j \leq a$ and $i \neq j)$. Therefore, $\mid \cup_{i=1}^{a}(C_i - C) \mid \geq a$, and as a result,

$$\mid C \mid \leq n_s - \mid \cup_{i=1}^{a}(C_i - C) \mid \leq n_s - a.$$

■

Proposition 6.6.2 can be used to verify Assumption 6.6.1 in the following way. After we find all single and double sensor minimal $A|A'$-distinguishers, we find the number of disjoint distinguishers, 'a'. Next we only need to check sensor combinations of size '3' to '$n_s - a$' for existence of any other minimal distinguishers (i.e., sensor sets of size '$n_s - a + 1$' to 'n_s' can be ignored).

Example 6.6.1 *Suppose $C_{tot} = \{c_1, c_2, \ldots, c_{10}\}$ $(n_s = 10)$, and the single and double $A|A'$-distinguishers are $\{c_1\}$, $\{c_2, c_3\}$ $\{c_3, c_4\}$. Thus there are $a = 2$ disjoint minimal distinguishers. It follows from Proposition 6.6.2, that there are no minimal distinguishers of size '9' and '10'.* ■

Proposition 6.6.3 *Suppose in k minimal $A|A'$-distinguishers $C_1, C_2, \ldots C_k$, there are l distinct sensors: $\mid \cup_{i=1}^{k} C_i \mid = l$. If there exists a minimal distinguisher C not in the above set and consisting of at most $n_s - l$ sensors $(|C| \leq n_s - l)$, then there exists an $A|A'$-distinguisher C' consisting of $n_s - l$ sensors $(|C'| = n_s - l)$ with $C \subseteq C'$ such that C' is not a superset of any C_i $(i = 1, \ldots, k)$.* ■

Before presenting the proof let us examine the proposition. The proposition implies that if each of $A|A'$-distinguishers of size '$n_s - l$' is a superset of one of C_i's, then there are no minimal distinguishers of size '$n_s - l$' or less (other than $C_1, \ldots, C_k$).

Proof of Proposition 6.6.3:

Let $C_l = \cup_{i=1}^{k} C_i$. Then $C_{tot} - C_l$ contains '$n_s - l$' sensors. Suppose $|C| = m \leq n_s - l$. Then let C' consist of the sensors in C and $n_s - l - m$ ($\leq n_s - l$) sensors from $C_{tot} - C_l$. Obviously C' is a superset of C and thus is an $A|A'$-distinguisher. C is a minimal distinguisher and does not belong to the set $C_1, \ldots, C_k$. Thus C cannot be a superset of any C_i, and neither can C' (since the sensors in $C' - C$ come from $C_{tot} - C_l$). ■

Proposition 6.6.3 can be used to verify Assumption 6.6.1 in the following way. After we find all single-sensor and double-sensor minimal distinguishers, we find the number of sensors in them and call it 'l'. If Assumption 6.6.1 is true, then every distinguisher of size '$n_s - l$' must be a superset of at least one of single-sensor or double-sensor minimal distinguishers. Once this is verified, we can ignore sensor combinations of size '3' to '$n_s - l - 1$' in the process of the verification of Assumption 6.6.1.

Example 6.6.2 *(Continued from Example 6.6.1)*

In the single-sensor or double-sensor minimal distinguishers $\{c_1\}$, $\{c_2, c_3\}$ $\{c_3, c_4\}$, *there are* $l = 4$ *sensors. Therefore we examine all sensors combinations of size* $n_s - l = 6$ *and make sure that they are supersets of single-sensor and double-sensor sets. Once this is verified, then sensor combinations of size* $3, 4$ *and* 5 *can be ignored. As a result, based on Proposition 6.6.2 and Proposition 6.6.3, to verify Assumption 6.6.1, we only need to examine sensor combinations of size* $6, 7$ *and* 8. ■

In the following subsection, we will discuss algorithms for sensor selection for a multi-resolution diagnosis that use **SMD**$(F|N)$'s and **SMD**$(F|F')$'s and do not require the computationally expensive distinguishability tests. **SMD**$(F|N)$'s and

SMD($F|F'$)'s have potential application in the computation of minimal sensor sets with *minimal cardinality* for detection and diagnosis problems. This problem is known to have worst-case exponential complexity [52] and its solution involves diagnosability tests. Once the **SMD**($F|N$)'s and **SMD**($F|F'$)'s are found, then the tests for verifying whether a given sensor set is suitable for the detection or diagnosis problem can be performed *without a diagnosability or distinguishability* tests.

Remark 6.6.1 *Another use of **SMD's** is in choosing the fault groups in multi-resolution diagnosis. A rule of thumb is to group the faults according to the subsystem they occur. It is desirable to group the faults so that the required sensor sets for group isolation (C_{giso}) and fault isolation ($C_{iso}^{(i)}$) have the smallest number of sensors. To achieve this, we may look for failure modes that similar behaviors or affect the system similarly. For such failures we can expect the **SMD**($F|N$) to be similar; in other words, the required sensors to distinguish them from normal mode would be the same. If placed in a group, these similar failures can be distinguished from other failures using a small set of sensors. To find similar **SMD**($F|N$), first we can look for identical **SMDs** and next, for **SMD**($F|N$)'s that have elements (minimal distinguishers) in common that are not presented in other **SMD**($F|N$)'s.* ■

6.6.2 Alternative Algorithm for Sensor Selection Using SMD's

The algorithms proposed in this subsection, use SMD($F|N$)'s and SMD($F|F'$)'s to find minimal sensor sets for multi-resolution diagnosis. As such they do not require distinguishability tests. As a result, the computational complexity of these algorithms are polynomial in the number of sensors and fault conditions but do not depend on the size of the system states. This makes them suitable for online implementation to be used, say, in the reconfiguration of the diagnosis system (Note that the algorithms for sensor selection given in Section 6.2 have a complexity of $\mathcal{O}(|X|^4)$). Of course, the price to pay is the computation of $\mathbf{SMD}(F|N)$'s and $\mathbf{SMD}(F|F')$'s which is done offline, at the design stage.

We begin the discussion with sensor selection for the first level (failure detection problem). Procedure 5.3 discussed for finding a minimal element of $\mathcal{D}$, first constructs an element of $\mathcal{D}$ and then removes sensors from that element until it becomes minimal. The algorithm proposed in this section follows the same procedure, except that in order to verify whether a sensor set belongs to $\mathcal{D}$ (i.e., is suitable for failure detection), instead of performing distinguishability tests, it relies on the $\mathbf{SMD}(F|N)$'s to find the answer. The following theorem explains the detail.

Theorem 6.6.4 *Let $C \subseteq C_{tot}$. Then $C \in \mathcal{D}$ if and only if for every $F \in \mathcal{F}$, there exists $C'_F \in$ **SMD**$(F|N)$ such that $C'_F \subseteq C$.*

Proof $C \in \mathcal{D} = \cap_{F\in\mathcal{F}}\mathbf{SD}(F|N)$ implies that for every $F \in \mathcal{F}$, $C \in\mathbf{SD}(F|N)$, which means there exists $C' \in\mathbf{SMD}(F|N)$ such that $C' \subseteq C$. Conversely, $\cup_{F\in\mathcal{F}}C'_F \in\mathbf{SD}(F|N)$ for all $F \in \mathcal{F}$ and therefore $\cup_{F\in\mathcal{F}}C'_F \in \cap_{F\in\mathcal{F}}\mathbf{SD}(F|N) = \mathcal{D}$. Hence $C \in \mathcal{D}$. ■

Procedure 6.2 finds a minimal element of $\mathcal{D}$ using $\mathbf{SMD}(F|N)$'s. After initialization of $C_{\mathcal{F}|\mathcal{N}}$, Step 2 and 3 calculate an element of $\mathcal{D}$. Specifically in Step 2, the fault conditions that have single minimal $F|N$-distinguishers are identified and the sensors in those minimal $F|N$-distinguishers are added to $C_{\mathcal{F}|\mathcal{N}}$. These sensors of course are present in any element of $\mathcal{D}$. Next in Step 3, sensors are added to $C_{\mathcal{F}|\mathcal{N}}$ so that it

Procedure 6.2: Let $\mathcal{F} = \{F_1, .., F_p\}$. Assume the set of minimal distinguishers **SMD**($F|N$) are given for every $F \in \mathcal{F}$.

```
1. Initialization: $C_{\mathcal{F}N} = \varnothing$, $I = \{1, 2, ...p\}$.
2. For $i=1, ..., p$
        If | SMD($F_i|N$)|=1 with SMD($F_i|N$)=C
                Then
                $C_{\mathcal{F}|N} = C_{\mathcal{F}N} \cup C$
            $I=I-\{i\}$
        End(If)
   End{For}
3. For all $i \in I$
         If $\{C \in$ SMD $(F_i| N)| C \subseteq C_{\mathcal{F}|N}\} = \varnothing$ then
            Choose $C \in$ SMD $(F_i| N)$ and $C_{\mathcal{F}|N} = C_{\mathcal{F}|N} \cup C$
        End (If)
    End (For)
4. For all $c \in C_{\mathcal{F}N}$
      For all $F \in \mathcal{F}$
4a.         If $\{C \in$ SMD $(F|N)$ | $C \subseteq C_{\mathcal{F}|N}$ - $\{c\}\} = \varnothing$ then
              Go to 4b
           End{If}
      End{For}
      $C_{\mathcal{F}|N} = C_{\mathcal{F}|N} - \{c\}$
4b.   Continue
   End (For)
```

Procedure 6.2: $C_{\mathcal{F}|N}$ is a minimal sensor set for fault detection.

becomes a superset of a minimal $F|N$-distinguisher (for every F). Thus by Theorem 6.6.4, after Step 3, C will be an element (not necessarily minimal) of $\mathcal{D}$. In Step 4, sensors are removed from $C_{\mathcal{F}|\mathcal{N}}$ one by one until $C_{\mathcal{F}|\mathcal{N}}$ becomes minimal. At step 4a, when a sensor is removed, in order to verify whether the resulting sensor set is an element of $\mathcal{D}$, Theorem 6.6.4 is used.

Now we find the complexity of Procedure 6.2. As in Section 6.5.1, we assume the elements of $\mathbf{SMD}(F|N)$'s are single-sensor and double-sensor sets and therefore each $\mathbf{SMD}(F|N)$ has $\mathcal{O}(n_s^2)$ elements. Step 2 of Procedure 6.2 can be performed in $\mathcal{O}(p)$ (where p is the number of failure modes). Step 3 can be performed in $\mathcal{O}(p \times n_s^2)$. Step 4 is executed $\mathcal{O}(n_s \times p)$ times and each time, it takes $\mathcal{O}(n_s^2)$ time. Therefore Step 4 is done in $\mathcal{O}(n_s^3 p)$. Thus assuming $\mathbf{SMD}(F|N)$'s are available, the time complexity of Procedure 6.2 becomes $\mathcal{O}(n_s^3 p)$.

Next we discuss an algorithm for finding a minimal sensor set for the fault group isolation problem. The proposed algorithm relies on the following theorem.

Theorem 6.6.5 *Let $C \subseteq C_{tot}$. Then $C \in \mathcal{GI}$ if and only if for every $F \in \mathcal{F}^{(i)}$ and $F' \in \mathcal{F}^{(j)}$ $(1 \leq i, j \leq l, i \neq j)$, there exists $C' \in \boldsymbol{SMD}(F|F')$ such that $C' \subseteq C$.*

Proof $C \in \mathcal{GI}$ implies (by Theorem 6.2.1) that $C \in \mathbf{SD}(F|F')$ for every $F \in \mathcal{F}^{(i)}$, $F' \in \mathcal{F}^{(j)}$ with $1 \leq i, j \leq l$ and $i \neq j$. Therefore there exists $C' \in \mathbf{SMD}(F|F')$ such that $C' \subseteq C$. The reverse can be shown similarly. ■

Procedure 6.3 finds a minimal element of $\mathcal{GI}$ using $\mathbf{SMD}(F|F')$'s. Similar to Procedure 6.2, Step 2 and 3 find an element of $\mathcal{GI}$ and then Step 4 trims the element to a minimal element of $\mathcal{GI}$. Procedure 6.3 does not involve distinguishability tests.

Assuming the elements of $\mathbf{SMD}(F|F')$'s are single-sensor and double-sensor sets and therefore each $\mathbf{SMD}(F|F')$ has $\mathcal{O}(n_s^2)$ elements (sensor sets), the complexity of Step 2 and 3 of Procedure 6.3 are $\mathcal{O}(p^2)$ and $\mathcal{O}(p^2 \times n_s^2)$ respectively. Step 4 executes $\mathcal{O}(n_s \times p^2)$ times and each time, it takes $\mathcal{O}(n_s^2)$. Thus Step 4 is done in $\mathcal{O}(n_s^3 \times p^2)$. Therefore, the complexity of Procedure 6.3, assuming $\mathbf{SMD}(F|F')$'s are available, is $\mathcal{O}(n_s^3 \times p^2)$.

A procedure for fault isolation at the third-level of multi-resolution diagnosis can be similarly developed. Here we only provide the required test in the following theorem. The procedure itself will be similar to Procedure 6.2 and 6.3 and is omitted for brevity.

```
Procedure 6.3:
1. Initialization: $C_{giso} = \emptyset$.
2. For $i=1, \dots, l$
      For $j=i, \dots, l$
         For $k=1, \dots, p_i$
            For $r=1, \dots, p_j$
               If $|\mathbf{SMD}(\mathcal{F}_k^{(i)} | \mathcal{F}_r^{(j)})|=1$ with $\mathbf{SMD}(\mathcal{F}_k^{(i)} | \mathcal{F}_r^{(j)})=\{C\}$ then
                   $C_{giso} = C_{giso} \cup C$
               End(If)
            End(For)
         End(For)
      End(For)
   End(For)
3. For $i=1, \dots, l$
      For $j=i, \dots, l$
         For $k=1, \dots, p_i$
            For $r=1, \dots, p_j$ and $(k,r) \in I_{ij}$
               If $\{C \in \mathbf{SMD}\ (\mathcal{F}_k^{(i)} | \mathcal{F}_r^{(j)}) | C \subseteq C_{giso}\} = \emptyset$ then
                  Choose $C \in \mathbf{SMD}\ (\mathcal{F}_k^{(i)} | \mathcal{F}_r^{(j)})$ and $C_{giso} = C_{giso} \cup C$
               End (If)
            End(For)
         End(For)
      End(For)
    End (For)
4. For all $c \in C_{giso}$
      For $i=1, \dots, l$
        For $j=i, \dots, l$
          For $k=1, \dots, p_i$
             For $r=1, \dots, p_j$
                If $\{C \in \mathbf{SMD}\ (\mathcal{F}_k^{(i)} | \mathcal{F}_r^{(j)}) \mid C \subseteq C_{giso} - \{c\}\} = \emptyset$ then
                    Go to 4a
                End{If}
             End(For)
          End(For)
        End(For)
     End (For)
End{For}
      $C_{giso} = C_{giso} - \{c\}$
4a.   Continue
   End (For)
```

Procedure 6.3: C_{giso} is a minimal sensor set for second-level fault diagnosis.

Theorem 6.6.6 *Let $C \subseteq C_{tot}$. Then $C \in \mathcal{FI}(\mathcal{F}^{(i)})$ for some $1 \leq i \leq l$ if and only if for every $F \in \mathcal{F}^{(i)}$ and $F' \in \mathcal{F}^{(i)}$ with $F \neq F'$, there exists $C' \in \boldsymbol{SMD}(F|F')$ such that $C' \subseteq C$.*

Proof The proof is similar to the proof of Theorem 6.6.5 and is based on Theorem 6.2.2. ∎

Finally note that minimal $F|F'$ distinguishers computed for sensor selection at the third-level are reused for sensor selection at the second-level and there is no need to find minimal distinguishers for fault groups. This is an advantage of the proposed method based on minimal distinguishers that becomes particularly useful if the length of the fault hierarchy (number of steps in multi-resolution diagnosis) becomes greater than three discussed here.

Chapter 7

CONCLUSION

7.1 Summary

In this thesis, a framework for multi-resolution fault diagnosis in discrete-event systems (DES) is proposed. Here a sequence of plant models, with increasing resolution, are used in fault diagnosis and the range of possible diagnosis is narrowed down step by step, until finally the failure mode is isolated. In this way, the original problem of fault diagnosis is replaced by a sequence of smaller problems. The plant models used at each step of diagnosis are abstractions of the original plant model. In this thesis, we propose to use model reduction through the solutions of the Relational Coarsest Partition problem to obtain these abstractions. For each diagnosis step, a minimal sensor set is chosen to have coarser output maps and hence, to improve the efficiency of model reduction. We discuss how the state and condition estimates obtained in each stage of diagnosis can be passed to next level so that the information in the previously monitored output sequence is preserved in going from one step of diagnosis to the next step. As an illustrative example, the proposed method is used to design a multi-resolution diagnosis system for an ozone generator plant.

The discrete-event plant can be in normal condition or in one of several faulty conditions. In this thesis, a polynomial algorithm is proposed that verifies failure diagnosability by examining the distinguishability of two plant conditions at a time. A procedure is presented that finds minimal sensor sets for distinguishability of one condition from another. These minimal sets are referred to as minimal distinguishers. A polynomial procedure is introduced that combines minimal distinguishers to obtain a minimal sensor set for fault diagnosis. The proposed method based on minimal distinguishers reduces the computational complexity of sensor selection, especially in the cases involving simultaneous failures. Another benefit of using minimal distinguishers is that their computation maybe speeded up using heuristics and expert knowledge.

The proposed method for sensor selection is particularly suitable for multi-resolution diagnosis since it permits some of the results of computations performed for sensor selection at the lowest (finest) level of multi-resolution diagnosis (specifically, the minimal distinguishers), to be reduced at higher levels. This feature is particularly useful in reducing the computations necessary for online reconfiguration of the multi-resolution diagnosis system. Specifically, for sensor selection which is the most computationally intensive part of design, an algorithm is proposed that does not require any distinguishability tests online. This reduction in online computations comes at the extra cost of offline (design stage) computations.

An important procedure used in sensor selection is testing diagnosability. In this thesis, a new procedure for testing diagnosability in timed DES is introduced based on timed Reachability Transition System (timed RTS). It is shown, through example, that if timed RTS has been already computed, say for diagnoser design, then the proposed test for diagnosability maybe executed with significantly fewer computations compared to tests developed for untimed models and adapted for timed systems. Furthermore, two new sets of sufficient conditions are provided under which diagnoser design (discussed in the literature) and diagnosability tests (introduced in this thesis), based on timed RTS can be performed efficiently even in cases when the transition-time sets associated with output changes are unbounded sets.

7.2 Future Research

The research presented in this thesis can be extended in several directions. Over the past few years, there has been a great interest in the study of the topic of decentralized diagnosis (with and without communication). A Possible future work includes incorporating a multi-resolution approach in solutions to decentralized problems. In this regard, issues such as fault grouping and sensor selection and their effect on the required inter-subsystem communication would be challenging and interesting.

In this thesis, we used natural projections on state sets to perform model reduction and obtain coarser models. Other approaches to abstraction such as the more general causal maps can also be investigated. Causal maps in particular have been used in the study of hierarchical supervisory control systems.

One of the benefits of using minimal distinguishers is that their computations may be speeded up using heuristics and expert knowledge. Intuitively, in sensor selection problems, developing heuristics for finding sensor sets for distinguishing of one faulty condition from another is easier than obtaining heuristics for finding sensor set for a more complex fault diagnosis problem We have not explored this idea in this thesis and leave it for future research.

One of the methods to manage computational complexity is to take advantage of any algebraic regularity in the model of the system. Vector discrete-event systems or Petri nets are examples of such cases. The extension of our results to vector DES or Petri nets is another direction for future research.

An important design decision in multi-resolution diagnosis is fault grouping. As mentioned in Chapter 6, putting "similar" faults in one group may help to reduce the size of sensor set required for diagnosis. The set of minimal distinguishers may provide clues in finding similarity between faults. For instance, if for two faults F and F', $\mathbf{SMD}(F|N)$ and $\mathbf{SMD}(F'|N)$ are "similar", then it may be inferred that F and F' affect that system similarly. We leave this topic for future research.

Finally, in this dissertation, we applied our results to a simplified generator system with a size manageable by manual computations. In order to assess the capability of our methodology, it is important to apply our technique to other real-world problems.

LIST OF REFERENCES

LIST OF REFERENCES

[1] J.J. Gertler, *Fault detection and diagnosis in engineering systems,* CRC press, 1998.

[2] C.G. Cassandras, S. Lafortune, *Introduction to Discrete Event Systems,* Springer, 1999.

[3] F. Lin, *Diagnosability of Discrete event systems and its applications,* Discrete Event dynamic Systems, vol.4, no. 2, pp.197-212, May 1994.

[4] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen and D. Teneketzis, *Diagnosaility of discrete-event systems,* IEEE Transactions on Automatic Control, vol.40, no. 2, pp.1555-1575, September 1995.

[5] S. Hashtrudi-Zad, R.H. Kwong and W.M. Wonham, *Fault diagnosis in discrete-event systems: Framework and model reduction,* IEEE Transactions on Automatic Control, vol. 48, no. 7, pp. 1199-1212, July 2003.

[6] D. Lawesson, U. Nilsson and I. Klein, *Fault isolation in discrete-event systems by observational abstraction,* Proceeding of the 42nd IEEE Conference on Decision and Control, Maui, Hawaii, USA, vol. 5, pp. 5118- 5123, December 2003.

[7] P. Jalote, *Fault tolerance in distributed systems,* P T R Prentice Hall, Englewood Cliffs, New Jersey, 1994.

[8] R. Milne, *Strategies for Diagnosis,* IEEE Transactions on Systems, Man and Cybernetics, vol. 17, no. 3, pp. 333 - 339, May 1987.

[9] J. Jr. Sottile and L.E. Holloway, *An overview of fault monitoring and diagnosis in mining equipment,* IEEE Transactions on Industry Applications, vol. 30, no. 5, pp. 1326 - 1332, September 1994.

[10] A. S. Willsky, *A Survey of Design Methods for Failure Detection in Dynamic System,* Automatica, vol.12, pp. 29-32, May 1976.

[11] R. Isermann, *Process Fault Detection Bsed on Modeling and Estimation Methods–A Survey,* Automatica, vol.20, pp. 387-404, July 1984.

[12] R. Isermann, *Fault-Diagnosis of Machines Via Parameter Estimation and Knowledge Processing–Tutorial Paper,* Automatica, vol.29, pp. 815-835, July 1993.

[13] R. Patton, R. Clark and P. M. Frank, *Issues of Fault Diagnosis for Dynamic Systems,* Springer, 2000.

[14] C. Angeli and A. Chatzinikolaou, *Online fault detection techniques for technical systems: a survey,* International Journal of Computer Science and Applications, vol.1, no. 1, pp. 12-30, January 2004.

[15] R. Isermann, *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance,* Springer, 2006.

[16] J. J. Gertler, *Survey of Model-Based Failure Detection and Isolation in Complex Plant,* IEEE Control Systems Magazine, vol.8, pp. 3-11, December 1988.

[17] W. Li and S. Shah, *Data-driven Kalman filters for non-uniformly sampled multirate systems with application to fault diagnosis,* Proceedings of the 2005 American Control Conference, Oregon, Portland, vol.4, pp. 2768 - 2774, June 2005.

[18] S. Jiang and R. Kumar, *Failure diagnosis of discrete-event systems with linear-time temporal logic specifications,* IEEE Transactions on Automatic Control, vol.49, no.6, pp. 934 - 945, June 2004.

[19] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. C. Teneketzis, *Failure diagnosis using discrete-event models,* IEEE Transactions on Control Systems Technology, vol.4, no.2, pp. 105 -124, March 1996.

[20] S. Bavishi and E. K. Chong, *Automated fault diagnosis using a discrete event systems framework,* IEEE Proceedings of the 9th International Symposium on Intelligent Control, Columbus, Ohio, USA, vol.16, no.18, pp. 213-218, August 1994.

[21] P. Baroni, G. Lamperti, P. Pogliano and M. Zanella, *Diagnosis of large active systems,* Artificial Intelligence, vol.110, no.1, pp. 135-183, May 1999.

[22] M. Sampath, R. Sengupta, S. Lafortune and D. Teneketzis, *Active Diagnosis of discrete-event systems,* IEEE Transactions on Automatic Control, vol.43, no.7, pp. 908-929, July 1998.

[23] S. Hashtrudi Zad, R.H. Kwong and W.M. Wonham, *Fault diagnosis in discrete-event systems: Incorporating timing information,* IEEE Transactions on Automatic Control, vol. 50, no. 7, pp. 1010-1015, July 2005.

[24] W. M. Wonham, *Supervisory on control of discrete -event systems,* University of Toronto, 2006.

[25] G. Lamperti, *A Bridged Diagnostic Method for the Monitoring of Polymorphic Discrete-Event Systems,* IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 34, no. 5, pp. 2222- 2244, October 2004.

[26] A. Darwiche, *Model-based diagnosis using causal networks,* Proceedings of Joint Conference on Artificial Intelligence (IJCA) on Artificial Intelligence, Montreal, Canada, pp. 211-217, August 1995.

[27] A. Misra, G. Provan, G. Karsai, G. Bloor and E. Scarl, *A generic and Symbolic Model-based Diagnostic Reasoner with Highly Scalable properties,* IEEE International Conference on Systems, Man, and Cybernetics, San Diego, CA, vol. 4, pp. 3154 - 3160, October 1998.

[28] Y. E. Fattah and G. Provan, *Modeling Temporal Behavior in the model-based Diagnosis of Discrete Event Systems,* Proceedings Of International Workshop in Principles of Diagnosis, Mont-Saint-Michel, France, pp. 43-50, September 1997.

[29] D. N. Pandalai and L. E. Holloway, *Template languages for fault monitoring of timed discrete event processes,* IEEE Transaction On Automatic Control, vol. 45, no. 5, pp. 868-882, May 2000.

[30] Y. Chen and G. Provan, *Modeling and Diagnosis of timed discrete event systems-A factory automation Example,* Proceedings Of American Automatic Control Conference, New Mexico, USA, vol. 1, pp. 31-36, June 1997.

[31] S. Jiang, Z. Huang and R. Kumar, *A polynormial Algorithm for Testing Diagnosability of Discrete event Systems,* IEEE Transactions On Automatic Control, vol. 46, no.8, pp. 1318-1321, August 2001.

[32] T. Yoo and S. Larfortune, *Polynomial time verification of diagnosability of partially observed discrete event systems,* IEEE Transactions On Automatic Control, vol. 47, no.9, pp. 1491-1495, September 2002.

[33] J. Pan and S. Hashtrudi-Zad, *Diagnosability Test for Timed Discrete-Event Systems,* Proceeding of 18th IEEE Conference on Tools with Artificial Intelligence, Washington, USA, pp. 63-70, November 2006.

[34] H. T. Simsek and R. Sengupta, *Fault Diagnosis for Intra-Platoon Communication,* Proceedings of the 38th IEEE Conference on Decision and Control, Phoenix, Arizona, USA, vol. 4, pp. 3520-3525, December 1999.

[35] D. N. Godbole, J. Lygeros, E. Singh, A. Deshpande and A. Lindsey, *Communication Protocols for a Fault Tolerant Automatic Highway Systems,* IEEE Transactions On Control Systems Technology, vol. 8, no.5, pp. 787 - 800, September 2000.

[36] L. Rozé and M. Cordier, *Diagnosing Discrete Event Systems: Extending the "Diagnoser Approach" to Deal With Telecommunoication Networks,* Discrete Event Dynamic Systems: Theory and Application, vol. 12, no.1, pp. 43-81, January 2002.

[37] A. Benveniste, S. Harr, E. Fabre and C. Jard, *Distributed and asynchronous discrete event systems diagnosis,* Proceedings of the 42nd IEEE Conference on Decision and Control, Maui, Hawaii USA, vol. 5, pp. 3742-3747, December 2003.

[38] P. Baroni, G. Lamperti, P. Pogliano and M. Zanella, *Diagnosis of a class of distributed discrete event systems,* IEEE Transactions on Systems, Man, and Cybernetics-PartA: Systems and Humans, vol. 30, no. 6, pp. 731-752, November 2000.

[39] A. Benveniste, E. Fabre, C. Jard, S. Haar, *Diagnosis of asynchronous discrete event systems, a net unfolding approach,* 15th IEEE Transactions on Automatic Control, vol. 48, no. 5, pp. 714-727, May 2003.

[40] S. Hashtrudi-Zad, R. Kwong and W. M. Wonham, *Fault Diagnosis and Consistency in Hybrid Systems,* 15th Proceeding of 38th Annual Allerton Conference on Communication, Control, and Computing, University of Illinois at Urbana-Champaign, USA, pp. 1135-1144, October 2000.

[41] F. Zhao, X. Koutsoukos, H. Haussecker, J. Reich, and P. Cheung, *Monitoring and Fault Diagnosis of Hybrid Systems,* IEEE Transactions on Systems, Man, and Cybernetics - Part B, vol. 35, no. 6, pp. 1225-1240, December 2005.

[42] P. J. Mosterman, *Diagnosis of Physical Systems With Hybrid Models Using Parametrized Causality,* The 4th International Conference on Hybrid Systems: Computation and Control, Rome, Italy, pp. 447-458, March, 2001.

[43] A. Ramírez-Treviñn, E. Ruiz-Beltrán, I. Rivera-Rangel and E. López-Mellado, *Online fault diagnosis of discrete event systems. A petri net-based approach,* IEEE Transactions on Automation Science and Engineering, vol. 4, no. 1, pp. 31-39, January, 2007.

[44] A. Paoli and S. Lafortune, *Safe diagnosability of discrete event systems,* Proceedings of the 42nd IEEE Conference on Decision and Control, Maui, Hawaii, USA, vol. 1, pp. 2658-2664, December, 2003.

[45] R. Debouk, S. Lafortune and D. Teneketzis, *Coordinated decentralized protocols for failure diagnosis of discrete event system,* Discrete Event Dynamical Systems: Theory and Applications, vol. 10, no. 1, pp. 33-86, January, 2000.

[46] G. Provan, *A model-based diagnosis framework for distributed systems,* Proceeding on the 13th International Workshop on Principles of Diagnosis (DX'02),Semmering, Austria, pp. 16-22, May, 2002.

[47] R. Debouk, R. Malik and B. Brandin, *A modular architecture for diagnosis of discrete-event systems,* Proceeding of the 41st IEEE Conference on Decision and Control, Las Vegas, NV, pp. 417-422, December 2002.

[48] Y.L. Chen and F. Lin, *Hierarchical Modeling and Abstraction of Discrete Event Systems using finite state machines with parameters,* Proceeding Of the 40th IEEE Conference on Decision and Control, Orlando, Florida, USA, vol. 5, pp. 4110-4115, December, 2001.

[49] O. Contant, S. Lafortune, and D. Teneketzis, *Diagnosis of Discrete Event Systems with Modular Structure,* Discrete Event Dynamic Systems: Theory and Applications, Vol. 16, No. 1, pp. 9-37, January 2006.

[50] A. M. Idghamishi and S. Hashtrudi-Zad, *Fault Diagnosis in Hierarchical Discrete- Event Systems,* Proceeding of 43rd IEEE Conference on Decision and Control, Atlantis, Bahamas, pp. 63-68, December, 2004.

[51] I. Mozetic, *Hierarchical model-based diagnosis,* International Journel of Man-Machine Studies, vol. 35, no. 3, pp. 329-352, 1991.

[52] A. Haji-Valizadeh and K. A. Loparo, *Minimizing the cardinality of an Events Set for Supervisors of Discrete Event Dynamical Systems,* IEEE Transactions On Automatic Control, vol. 41, no. 11, pp. 1579-1593, November, 1996.

[53] S. Jiang, and R. Kumar, *Optimal Sensor Selection for Discrete-Event Systems with Partial Obsrvation,* IEEE Transactions On Automatic Control, vol. 48, no. 3, pp. 369 - 381, March, 2003.

[54] R. Debouk, S. Lafortune and D. Teneketzis, *On an Optimization Problem in Sensor Selection,* Discrete event dynamic systems: Theory and Applications, vol. 12, no. 4, pp.417-445, October, 2002.

[55] T. Yoo and S. Lafortune, *NP-Completeness of Sensor Selection Problems Arising in Partially Observed Discrete Event Systems,* IEEE Transactions On Automatic Control, vol. 47, no. 9, pp. 1495-1499, September, 2002.

[56] J. Pan and S. Hashtrudi-Zad, *Diagnosability Analysis and Sensor Selection in Discrete-Event Systems with Permanent Failures,* Proceeding of 3rd IEEE Conference on Automation, Science and Engineering (CASE), Phoneix, USA, pp. 869-874, September, 2007.

[57] L. Aguirre-Salas, *Sensor selection for observability in interpreted petri nets: a genetic approach,* Proceedings of the 42nd IEEE Conference on Decision and Control, Maui, Hawaii USA, vol. 6, pp. 3760-3765, December, 2003.

[58] S. Khuller, G. Kortsarz, and K. Rohloff, *Approximating the minimal sensor selection for supervisory control,* The 7th IFAC Workshop on Discrete-Event Systems, Reims, France, pp. 85-90, September, 2004.

[59] Y. Ru and C. N. Hadjicostis, *Approximating optimal place sensor selection for structural observability in discrete event systems modeled by petri nets,* Proceedings of the 46th IEEE Conference on Decision and Control, New Oriean, LA, USA, vol. 17, pp. 1892-1897, December, 2007.

[60] R. Su and W. M. Wonham, *Hierarchical fault diagnosis for discrete-event systems under global consistency,* Discrete Event Dynamic Systems: Theory and Applications, vol. 16, no. 1, pp. 39-70, January, 2006.

[61] C. M. Özveren and A. S. Willsky, *Aggregation and Multi-level Control in Discrete Event Systems,* Automatica, vol. 28, no. 3, pp. 565-577, May, 1992.

[62] A. Mohammadi-Idghamishi and S. Hashtrudi-Zad, *Hierarchical fault diagnosis: Application to an ozone plant,* IEEE Transactions on Systems, Man, and Cybernetics: Part C, vol. 37, no. 5, pp. 1040-1047, September 2007.

[63] B. A. Brandin and W. M. Wonham, *Supervisory Control of Timed Discrete Event Systems,* IEEE Transactions On Automatic Control, vol. 39, no. 2, pp. 329-342, February 1994.

[64] S. Maclane and G. Birkhoff, *Algebra,* New York: Macmilla, 1967.

[65] M. Larsson, *On modeling and diagnosis of discrete event dynamic systems,* Thesis in Linköping Studies in Science and Technology, http://www.control.isy.liu.se, 1997.

[66] I. Roychoudhury, G. Biswas, X. Koutsoukos and S. Abdelwahed, *Designing distributed diagnosers for complex physical systems* 16th International Workshop on Principles of Diagnosis (DX 05), Monterey, California, USA, pp. 31-36, June 2005.

[67] R. Mohammadi and S. Hashtrudi-Zad, *A recursive algorithm for diagnosis in hierarchical finite-state machines,* Proceedings of 2007 IEEE International Conference on Systems, Man and Cybernetics, Montreal, QC, Canada, pp. 1345-1350, October 2007.