Location of a Faulty Module in a Computing System

TEIN-HSIANG LIN, MEMBER, IEEE, AND KANG G. SHIN, SENIOR MEMBER, IEEE

Abstract—**Considering the interplay between different phases of fault tolerance, a new problem of locating a faulty module in a computing system is formulated and solved in this paper. First, the probability of each module being faulty or *faulty probability* is calculated using the likelihood principle from the model parameters for fault detection, diagnostics, error propagation, and error detection. Then, based on the faulty probabilities and a given required diagnostic coverage, the order in which modules are to be diagnosed and the maximum time allotted to diagnose each module are determined by minimizing the average total diagnostic time. An example is presented and analyzed to answer the question of whether or not a system should delay the diagnosis upon detection of an error until more errors are detected.**

Index Terms—**Bayesian decision theory, detection latency, error propagation, fault (error) detection, fault (error) latency, fault location, periodic diagnostic, system level fault diagnosis.**

I. Introduction

The rapid progress in VLSI technology has made it possible to build systems with a massive number of components for high performance and high reliability. In such complex systems, it is extremely difficult to locate the faulty component(s) upon occurrence of a failure. A new probabilistic approach to the problem of locating a fault in a multicomponent system is thus proposed in this paper by taking into account the effects of such mechanisms as error detection, error propagation, and periodic diagnostics.

Due to the packaging and scope of each reconfigurable unit, it is sufficient in most systems to locate a ''macro component,'' called a *module,* which contains one or more faulty ''micro'' components. A module represents any well-defined subunit of the system; it could be a hardware unit, a software subrountine, or a combination of both. Each module is assumed to have its own independent detection mechanism(s).

Our approach is concerned with system level fault diagnosis for which various models have been proposed in the literature. The most notable are the model proposed in [22] and extensions thereof. Among them, only a couple of models are directly related to our work and thus will be reviewed below. (See Section VI for reviews on others.)

Bossen and Hsiao [4] proposed a probabilistic model for the detection of intermittent and transient faults. Since intermittent and transient faults that had occurred during normal operation may disappear during off-line testing, they used the information gathered from detection mechanisms to isolate faults. Their model was implemented as an automated diagnostic methodology for the IBM 3081 Processor Unit [30]. New hardware features were added in the IBM 3081 Processor Unit to record the system's status at the time of occurrence of errors. Thus, in most cases the faulty module can be directly isolated from this information without additional diagnosis. If direct isolation of faults is not possible, then *probable* faulty modules are tested for stuck-at faults in combinational networks of each of the modules. If the fault is still not found, modules are sequentially replaced until the system is repaired. The sequence of replacement is determined based on the numbers of circuits in each module.

The above approach is deterministic in nature since abundant information on the fault is available, thus almost always reducing the number of probable faulty modules down to one. By contrast, our model will require only the knowledge of the times and places (module identity) of error detections and a probabilistic model of fault behavior. Based on this minimal information, our objective is to locate the faulty module as quickly as possible. For this, a human or another computer system is brought in upon detection of one or more errors to diagnose every module in the system *sequentially* until the faulty module is identified under the single faulty module assumption[1] or all the modules are exhausted. A Bayesian decision approach will be used to determine the order of modules to be diagnosed and the maximum diagnostic times for individual modules.

The paper is organized as follows. Section II defines the parameters associated with fault detection, periodic diagnostics, error propagation, and error detection. In Section III, the faulty probability of each module is derived, and the optimal order in which modules are to be diagnosed is determined by using a Bayes decision criterion. In Section IV, the optimal diagnostic times for individual modules are determined by minimizing the average total diagnostic time. An example is then presented in Section V. In Section VI, we compare our model to other system level diagnosis models and remark on

Manuscript received June 15, 1987; revised December 6, 1987. This work was supported in part by the NASA under Grant NAG-1-296 and the ONR under Contracts N00014-85-K-0531 and N00014-85-K-0122. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

T.-H. Lin was with Real-Time Computing Laboratory, Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, MI 48109. He is now with the Department of Electrical and Computer Engineering, The State University of New York at Buffalo, Buffalo, NY 14260.

K. G. Shin is with Real-Time Computing Laboratory, Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, MI 48109.

IEEE Log Number 8931919.

[1] Relaxation of this assumption will be addressed in Section VI.

0018-9340/90/0200-0182$01.00 © 1990 IEEE

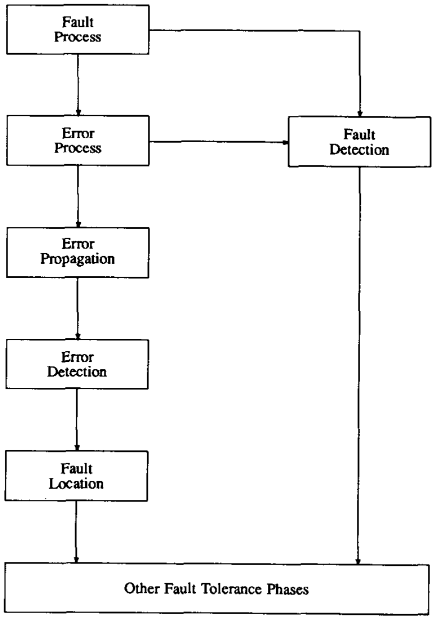

Fig. 1. Fault tolerance phases in a computing system.

how to relax the single faulty module assumption. The paper concludes with Section VII.

II. System Model

An N-module computing system is represented by a digraph $D = (V, E)$, where $V = \{v_1, \cdots, v_N\}$ denotes the set of nodes, $E = \{e_{ij};\ 1 \leq i, j \leq N\}$ denotes the set of directed edges. Each node in V represents a module in the system, and a directed edge e_{ij} represents a communication link[2] from v_i to v_j. Typical methods of communication between software modules are message passing or shared memory. Hardware modules can communicate via control and data signals. If there is no communication link from v_i *to* v_j, E will not contain e_{ij}, or e_{ij} is a null edge.

The fault tolerance in a multimodule system is achieved by several sequential phases as shown in Fig. 1. The *fault process* determines the time, the location, and the type of a fault that occurs in the system. A fault may induce error(s), which is described by the *error process.* To distinguish an error from a fault, both are defined as follows.

- A *fault* is a damage, defect, or deviation from the normal state of the computing system on which tasks execute.
- An *error* is a deviation from the specification of a task.

For example, a broken wire, an electromagnetic interference, or a bug in a program is a fault, whereas an incorrect result or an untimely control signal is an error. A thorough discussion on the terminology about faults and errors can be found in [13].

Based on the definitions of fault and error, the *detection phase* can be divided into *fault detection* and *error detection.* The former is concerned with detecting the manifestation of a fault by some means other than program execution (e.g., a built-in tester), while the latter deals with detecting those errors in program execution induced by fault(s). Hence, the fault location phase will be called for only after an error detection. The problem of locating the faulty module becomes complicated when error detection mechanisms are not perfect (i.e., have less than 100 percent coverage), since an error may propagate from one module to others via intermodule communication before it is detected in some module(s).

In the remainder of this section, the above fault-tolerant phases will be characterized with probabilistic models. The density and cumulative distribution functions of a random variable V_i (V_{ij}) will be denoted as f_i^V (f_{ij}^V) and F_i^V (F_{ij}^V), respectively.

A. Faults and Errors

A module is said to be *faulty* if it contains one or more faults and is said to become *contaminated* if it contains an error. Let T_i^F, the v_i's *faulty time,* denote the time instant a fault occurs in v_i. Let T_i^C, the v_i's *contaminating time,* denote the time instant the first error occurs in v_i as a result of either the manifestation of a fault within v_i or the propagation of error(s) from other module(s). We assume that the system contains at most one faulty module at a time, although the faulty module could contain multiple faults. An extension of our results to the case of multiple faulty modules will be addressed in Section VI.

If v_i is the faulty module, the *fault latency* of v_i, denoted by L_i, is defined as the time interval between v_i's faulty time and contaminating time. A methodology for determining the distribution of the faulty latency on real systems was developed in [27].

The rate of fault occurrence in v_i is characterized by the *fault cycle* in v_i, denoted by Y_i, which is the time interval between two consecutive faulty times of v_i. In Fig. 2, a time chart is used to illustrate the parameters defined in this subsection.

B. Error Propagation

If a module's detection mechanism is not perfect, faults in that module induce errors which may then propagate to other modules before some module detects one or more of these errors. To describe error propagation in the system, a (propagation) *path* from v_i to v_k, written as $(v_i, \cdots, v_k)$ is defined as a directed path in D where all nodes in the path are *distinct.* Only distinct nodes in an error propagation path are considered, because error propagation into an already contaminated module has negligible effects on the propagation behavior of the module. Error will propagate from v_i to v_k if there exists at least one propagation path from v_i to v_k. Define the *error propagation time* from v_i to v_j, denoted by X_{ij}, as the time interval between the contaminating time of v_i and that of v_j. Clearly, X_{ij} holds a physical meaning only when $X_{ij} \geq 0$. Thus, the definition of X_{ij} should be made under the assumption that v_i is the only faulty module or the first module to be contaminated in the system. Under the single faulty module assumption, the faulty module will be the first to be contaminated.

The error propagation times contain complete information

[2] Not to confuse this with a connection link.

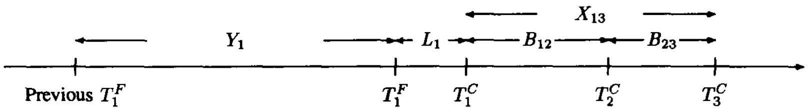

Fig. 2. Timing chart for an error propagating via (v_1, v_2, v_3).

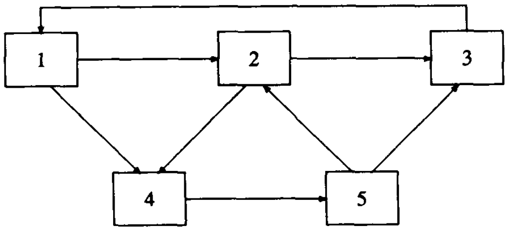

Fig. 3. System graph $D1$.

on the behavior of error propagation and can be measured experimentally. However, due to the difficulty and cost of measuring the error propagation times (see [28] for a detailed account of this), we define the *direct propagation time* of every nonnull edge e_{ij}, denoted by B_{ij}, as the time for an error to propagate from v_i to v_j via e_{ij}. The distribution function of B_{ij} is called the *direct propagation function* of e_{ij}.

The differences between an error propagation time and a direct propagation time are that 1) the latter is associated with a directed edge while the former is defined for every ordered pair of modules, and 2) the latter accounts for error propagation through a particular edge while the former is the minimum propagation time over all propagation paths between the given pair of modules. Since direct propagation times are defined for communication links in the system, without loss of generality, one can assume that they are independent of one another and their distributions do not change even if the underlying fault model is changed.

The relation between X_{ij}'s and B_{ij}'s can be obtained easily. First, identify all error propagation paths from v_i to v_j and calculate the error propagation time along each path by summing up the direct propagation times of all edges in the path. Then, X_{ij} is the minimum among all these path propagation times. For example, in the system graph $D1$ shown in Fig. 3,

$$X_{13} = \min\,(B_{12}+B_{23},\ B_{14}+B_{45}+B_{53},\ B_{14}+B_{45}+B_{52}+B_{23},\ B_{12}+B_{24}+B_{45}+B_{53}). \quad (2.1)$$

Assuming that an error propagates via the push (v_1, v_2, v_3) of graph $D1$, the relation between B_{ij} and X_{ij} is shown in Fig. 2.

Deriving the distributions of X_{ij}'s from that of B_{ij}'s is not trivial. To perform this systematically and efficiently, two algorithms have been proposed in [28] where experimental measurements of the direct propagation times in the fault-tolerant multiprocessor (FTMP) were also reported.

C. Fault and Error Detection

All detection mechanisms detect only errors, while faults have to be diagnosed by *coercing* them into detectable errors. Based on the detection methodology employed, the scheme in [26] classifies detection mechanisms into three types: 1) fault detection, 2) error detection, and 3) periodic diagnostics. Examples of fault detection (or *signal level detection* in [26]) mechanisms are built-in self-checking circuits, error detection codes, and duplicate complementary circuits. Error detection (or *function level detection* in [26]) can be implemented in various ways, such as capability checking, acceptance test, invalid op-code checking, and time-out. Periodic diagnostics are off-line testing programs which are run periodically and can exercise the system's components with imitated inputs to activate and then detect faults.

The distinct feature of fault detection mechanisms is the immediate detection when a fault induces an error so that an error cannot propagate without being detected. Consequently, the module where a detection is made by fault detection mechanisms is the faulty module; the fault location phase is not needed after a fault detection. The faults that can be detected by fault detection mechanisms in a module are defined as *FD-detectable* faults. The fault detection coverage of v_i, denoted by C_i^F, is the probability of a fault in v_i being FD-detectable. Ideally, every fault should be FD-detectable (i.e., $C_i^F = 1$ for all i), which is practically impossible. Thus, to complement (imperfect) fault detection mechanisms, the other two types of detection mechanisms must also be used.

A periodic diagnostic can usually detect more faults with less cost than a fault detection mechanism. The disadvantage is that errors may have been induced and propagated to other modules before the fault is diagnosed, since periodic diagnostics cannot detect faults between diagnostics. The fault location phase is not neccessary following a detection with periodic diagnostics. The faults which are not FD-detectable but are detectable during periodic diagnostics are defined as *PD-detectable* faults. The coverage of periodic diagnostics in module v_i, denoted by C_i^P, is then the probability of a fault in v_i being PD-detectable. C_i^P is a monotonically increasing function of the diagnostic time. Because the system will not be available for useful work during periodic diagnostics, the time for a complete diagnostic (i.e., $C_i^P = 1 - C_i^F$) is usually too long to apply frequently. It is a common practice to perform a short periodic diagnostic during normal operation and perform a complete diagnostic as such a need arises or when the system is idle. Periodic diagnostics do not have to be scheduled at uniform intervals, but the time to perform each diagnostic is usually fixed. The scheduling of incomplete periodic diagnostics has been studied extensively by many researchers [12], [16], [20], [21], [32], [31]. Based on these results, we will assume that the scheduling of periodic diagnostics on each module has been determined *a priori* and each module records the time when the last periodic diagnostic was performed.

The faults that are neither FD-detectable nor PD-detectable are *undetectable* faults. Undetectable faults can be captured only during the fault location phase after the errors induced by these faults are detected by error detection mechanisms.

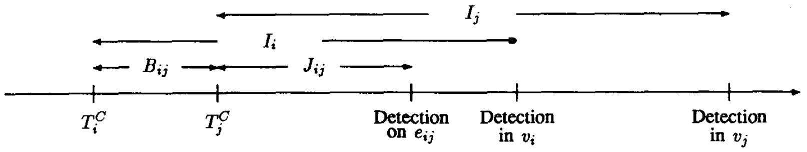

Fig. 4. A timing chart of error detections.

As discussed above, faults are classified according to the type of detection mechanisms that can detect them. This classification implies that there is no randomness in fault detection and periodic diagnostics. That is, if a fault is undetectable, it will never be detected by fault detection mechanisms or periodic diagnostics.

A module often employs several error detection mechanisms simultaneously to accelerate the error detection process. Errors induced by a fault proliferate until one of them is trapped by an error detection mechanism. The pattern of error proliferation depends heavily upon the system workload at the contaminating time, and thus, is random. Therefore, unlike the treatment of fault detection and periodic diagnostics, error detection will be modeled probabilistically.

Based on where errors are detected in a module, error detection mechanisms are divided into two groups: *internal* and *boundary* detection schemes. For a module, internal detection schemes refer to all the error detection mechanisms which check the module's internal structure, while the boundary detection schemes detect errors by checking the module's outputs. For example, the capability checking and acceptance test are internal detection schemes, whereas the NMR voting on the outputs is a boundary detection scheme.

Define the *internal detection latency* of v_i, denoted by I_i, as the time interval from the v_i's contaminating time until an error is detected by an internal detection scheme in v_i. If a module v_i does not have any internal error detection mechanism, then $I_i = \infty$.

Define the *boundary detection latency* on e_{ij}, denoted by J_{ij}, as the time interval from the v_j's contaminating time until an error is detected at the v_i's side of e_{ij} by a boundary detection scheme. If there is no boundary error detection mechanism on e_{ij}, then $J_{ij} = \infty$. Note that the starting point of J_{ij} is the time when an error propagates from v_i to v_j. Thus, Prob[$J_{ij} = 0$] can be intepreted as the probability of detecting an error just before it propagates to v_j via e_{ij}.

For any module v_i, we define the *detection time* as the time instant the first error detection is made in v_i by either an internal or a boundary detection scheme. The *detection latency of* v_i, denoted by K_i, is then defined as the time interval from v_i's contaminating time to its detection time. A time chart is shown in Fig. 4 to illustrate all the latencies defined in this subsection.

Denote the density and cumulative distribution functions of K_i by $c_i(\cdot)$ and $C_i(\cdot)$, respectively. Traditionally, the error detection coverage, defined as the probability of detecting errors existing in a module or system, is used as a measure to evaluate the effectiveness of error detection mechanisms. However, this definition does not specify the detection latency which plays a key role in determining the actual coverage. It also implies that if an error is not detected within a certain time limit, the error is considered to be undetectable forever. Hence, it would be more appropriate to define the error detection coverage as a function of time, $C_i(t)$. An error is undetectable only if its associated error latency approaches infinity.

The detection latency can be viewed as a result of the competition among internal detection and boundary detection schemes, i.e.,

$$K_i = \min\,(I_i,\, B_{ik_1} + J_{ik_1},\, \cdots,\, B_{ik_n} + J_{ik_n})$$

with $\{e_{ik_1}, \cdots, e_{ik_n}\}$ being the set of all outgoing edges at v_i. Assuming that I_i, B_{ij}, and J_{ij} are independent, $C_i(t)$ can be obtained by

$$[1 - C_i(t)] = [1 - F_i^I(t)][1 - F_{ik_1}^{B*J}(t)] \cdots [1 - F_{ik_n}^{B*J}(t)]$$

where "$*$" denotes convolution and

$$f_{ik_j}^{B*J}(t) = f_{ik_j}^{B}(t) * f_{ik_j}^{J}(t).$$

III. The Problem of Fault Location

When an error is detected in a module, the following two decisions have to be made before the faulty module can be located:

1) the order of modules to be diagnosed (or the diagnostic order)

2) the maximum diagnostic time for each module, i.e., when to stop diagnosing a module if a fault is not yet found in the module.

The former is dealt with in this section and the latter will be treated in the next section.

Under the single faulty module assumption, the determination of the diagnostic order can be formulated as a Bayesian decision problem. The parameter space contains N hypotheses:

$$H_i : v_i \text{ is the faulty module for } i = 1, \cdots, N.$$

If the *decision time* of a system, denoted by T, is the time a system's normal operation is stopped due to the detection of error(s), the sample space is the set of fault syndromes defined below. The *fault syndrome* of a system, denoted by $\mathcal{S}$, is defined as a record of error detections made from 0 to T, i.e.,

$$\mathcal{S} = [v_{b_1}, t_1; \cdots; v_{b_\delta}, t_\delta]$$

where t_i is the time the first error detection in v_{b_i} is made and δ is the number of modules that have made error detection(s) up to T. The prior and posterior faulty probabilities, denoted respectively by π_i' and π_i for $i = 1, \cdots, N$, are then defined as

$\pi_i' = \text{Prob}[H_i \text{ is true}]$ before considering the fault syndrome $\mathcal{S}$,

$\pi_i = \text{Prob}[H_i \text{ is true}]$ after considering the fault syndrome $\mathcal{S}$.

Usually, the decision time is immediately after the system's first error detection to minimize error propagation. However, it is possible for $\mathcal{S}$ to have multiple error-detecting modules, due mainly to the following two reasons. First, the decision time may be predetermined or at fixed intervals. Second, the decision time may be deliberately postponed to gather more information about error propagation.

An action A is defined as the actual diagnostic order and denoted by

$$A = [a_1, a_2, \cdots, a_N]$$

where v_{a_1} is the first module to be diagnosed, v_{a_2} is the second, and so on. If the number of modules diagnosed before locating the faulty module is defined as the loss function

$$L(H_i, A) = j \qquad \text{if } i = a_j,$$

the posterior Bayesian expected loss of A will be

$$\beta(A) = \sum_{k=1}^{N} \pi_k L(H_k, A)$$

$$= \sum_{j=1}^{N} j\pi_{a_j}.$$

The Bayes decision is then the action which has the minimum $\beta(A)$.

Theorem 1: Given the faulty probabilities π_i, $i = 1, \cdots, N$, $\beta(A)$ is minimized if $\pi_{a_1} \geq \pi_{a_2} \geq \cdots \geq \pi_{a_N}$.

Proof: Suppose there exist i and j such that $i < j$ and $\pi_{a_i} < \pi_{a_j}$ for an action $A = [a_1, \cdots, a_N]$. Construct a new action A' by exchanging a_i and a_j, i.e., $A' = [a_1, \cdots, a_j, \cdots, a_i, \cdots, a_N]$. Then

$$\begin{aligned}\beta(A) - \beta(A') &= i(\pi_{a_i} - \pi_{a_j}) + j(\pi_{a_j} - \pi_{a_i})\\ &= (j-i)(\pi_{a_j} - \pi_{a_i})\\ &> 0.\end{aligned}$$

Therefore, the Bayesian expected loss can be reduced by the above exchanges until the minimum is reached when $\pi_{a_1} \geq \pi_{a_2} \geq \cdots \geq \pi_{a_N}$. ■

From Bayes' Theorem, the posterior faulty probability π_i can be calculated as

$$\pi_i = \frac{\pi_i' \mathcal{L}_i(\mathcal{S})}{\sum_{k=1}^{N} \pi_k' \mathcal{L}_k(\mathcal{S})}$$

where $\mathcal{L}_i(\mathcal{S})$ is the *likelihood of the fault syndrome* $\mathcal{S}$ under the hypothesis H_i.

A. Prior Distribution of H_i

The prior distribution of H_i is our subjective belief about the location of the faulty module before the results from error detection are considered. If we know very little *a priori* the whereabout of the fault module, a *noninformative* prior can be employed, which assigns an equal probability to all hypotheses. An informative prior can be determined if we know the previous faulty times of all modules, denoted by T_i^Y, and the distributions of fault cycles of all modules, denoted by F_i^Y. In case v_i has never been faulty, T_i^Y may represent the time when v_i was first put into operation. This informative prior will be calculated on the basis of our suspicion of an individual module being faulty, which is just the conditional probability that one module is faulty and others are not at time T. For example, our suspicion of v_i being faulty is

$$F_i^Y(T - T_i^Y) \prod_{j=1, j\neq i}^{N} (1 - F_j^Y(T - T_j^Y)).$$

The prior probability of H_i is then the relative suspicion of v_i being faulty, i.e.,

$$\pi_i' = \frac{F_i^Y(T - T_i^Y) \prod_{j=1, j\neq i}^{N} (1 - F_j^Y(T - T_j^Y))}{\sum_{j=1}^{N} \left(F_j^Y(T - T_j^Y) \prod_{j=1, j\neq i}^{N} (1 - F_j^Y(T - T_j^Y)) \right)}.$$

B. Likelihood of Fault Syndrome

The likelihood $\mathcal{L}_i(\mathcal{S})$ is the conditional probability of $\mathcal{S}$ given that v_i is the faulty module. To determine $\mathcal{L}_i(\mathcal{S})$, we first derive the density function of v_i's contaminating time given $\mathcal{S}$, denoted by $f_i^C(t)$.

Let T_i^D denote the time when the previous complete diagnosis on v_i is done and let T_i^P denote the time when the previous periodic diagnostic on v_i is done. A complete diagnosis is assumed to have 100 percent coverage; so we are certain that v_i is fault-free immediately after T_i^D if it passed the complete diagnosis. T_i^D is assumed to equal T_i^Y if no complete diagnosis has been applied since T_i^Y. T_i^P is assumed to equal T_i^D if no periodic diagnostic has been applied since T_i^D. In v_i, only undetectable faults could have occurred during the interval between T_i^D and T_i^P, and only PD-detectable and undetectable faults could have occurred during the interval between T_i^P and t_1, the time of the first error detection. The likelihood of $T_i^F = t$, denoted by $\ell_i^F(t)$, is then

$$\ell_i^F(t) = \begin{cases} (1 - C_i^F - C_i^P) f_i^Y(t - T_i^Y)(1 - F_i^Y(t_1 - t) \\ \qquad \text{if } T_i^D \leq t < T_i^P \\ (1 - C_i^P) f_i^Y(t - T_i^Y)(1 - F_i^Y(t_1 - t)) \\ \qquad \text{if } T_i^P \leq t < t_1 \end{cases}.$$

Thus, the density function of T_i^F is expressed as

$$f_i^{T^F}(t) = \frac{\ell_i^F(t)}{\int_{T_i^D}^{t_1} \ell_i^F(\tau)\, d\tau}.$$

By definition, $T_i^C = T_i^F + L_i$ so that the density function of T_i^C can be calculated as

$$f_i^{T^C}(t) = f_i^{T^F}(t) * f_i^L(t)$$

where "*" denotes the convolution. However, if $\mathcal{S}$ is given, T_i^C can be no later than t_1 and no earlier than T_i^D. Hence, $f_i^C(t)$ should be expressed as

$$f_i^C(t)=\frac{f_i^{T^C}(t)}{\int_{T_i^D}^{t_1} f_i^{T^C}(\tau)\,d\tau}.$$

Now, the likelihood of $\mathcal{S}$ can be calculated as

$$\mathcal{L}_i(\mathcal{S})=\int_{T_i^C}^{t_1} \mathcal{L}_i(\mathcal{S},\ \tau)f_i^C(\tau)\,d\tau$$

where $\mathcal{L}_i(\mathcal{S},\ \tau)$ is the likelihood of $\mathcal{S}$ given that v_i is the faulty module and $T_i^C = \tau$.

Let the error latency from v_i to v_j, denoted by E_{ij}, be the time interval from the v_i's contaminating time to the v_j's detection time. Then, E_{ij} is the sum of the error propagation time X_{ij} and the detection latency in v_j, i.e.,

$$E_{ij}=X_{ij}+K_j.$$

Note that if $i = j$, $E_{ij} = K_j$. The spread of errors from a faulty v_i can be characterized by the joint distribution of E_{i1}, E_{i2}, $\cdots$, and E_{iN}, called the v_i's *joint distribution of error latencies*. Having the joint distribution of error latencies,

$$\mathcal{L}_i(\mathcal{S},\ \tau)=\text{Prob}[E_{ib_1}=t_1-\tau,\ \cdots,\ E_{ib_\delta}=t_\delta-\tau,$$
$$E_{iw_1}>T-\tau,\ \cdots,\ E_{iw_{N-\delta}}>T-\tau]$$

where $v_{\omega_1}, \cdots, v_{\omega_{N-\delta}}$ are the modules which had not detected any error. It is very complex and time consuming to analytically derive the joint distribution of error latencies, because E_{ij}'s are not independent. A more practical approach to get $\mathcal{L}_i(\mathcal{S},\ \tau)$ will be numerical simulation using random number generators.

IV. Optimal Diagnosis Procedure

Given a system diagnostic coverage and the optimal diagnostic order determined earlier, we now derive the optimal diagnostic times for individual modules by minimizing the average total diagnostic time.

Periodic diagnostics are usually self-diagnostic routines, but diagnostics at this stage have to be run by an independent system to ensure consistency and high coverage. Assuming that all modules are diagnosed sequentially by an independent fault-free system, the average total diagnostic time, denoted by TD, is defined as the average time from the start of the first diagnostic until the faulty module is identified or all the modules are exhausted. Given that there is a faulty module in the system, the system diagnostic coverage, denoted by M, is the conditional probability of locating the faulty module. Let $\Phi_i(t)$ be the percentage of faults that the diagnostic can discover in v_i for the diagnostic time t, and let $\phi_i(t) = d\Phi_i(t)/dt$. All $\phi_i(t)$'s are assumed to be continuous functions on $[0, \infty)$. Let D_i, $1 \le i \le N$, denote the optimal diagnostic time for v_i. Our problem can be formally stated as follows.

The Optimal Diagnostic Problem: Given M and π_i, $1 \le i \le N$, determine D_i, $1 \le i \le N$, which minimize TD.

Without loss of generality, we can assume that $\pi_1 \ge \pi_2 \ge \cdots \ge \pi_N$ and the diagnostic order is $A = [1, 2, \cdots, N]$. The following theorem provides a necessary condition for D_i.

Theorem 2: Given M and $\pi_1 \ge \pi_2 \ge \cdots \ge \pi_N$, D_i's must satisfy the following equations:

$$\begin{aligned}
\phi_{N-1}(D_{N-1})&=\frac{\pi_N\phi_N(D_N)+\pi_N\Phi_N(D_N)}{\pi_{N-1}[1+D_N]}\\
\phi_{N-2}(D_{N-2})&=\frac{\pi_N\phi_N(D_N)+\pi_N\Phi_N(D_N)+\pi_{N-1}\Phi_{N-1}(D_{N-1})}{\pi_{N-2}[1+D_N+D_{N-1}]}\\
&\ \ \vdots \qquad\qquad \vdots\\
\phi_1(D_1)&=\frac{\pi_N\phi_N(D_N)+\pi_N\Phi_N(D_N)+\pi_{N-1}\Phi_{N-1}(D_{N-1})+\cdots+\pi_2\Phi_2(D_2)}{\pi_1[1+D_N+D_{N-1}+\cdots+D_2]}
\end{aligned}\tag{4.1}$$

Proof: The diagnostic coverage of the system M is a weighted sum of all modules' individual diagnostic coverages, i.e.,

$$M=\sum_{i=1}^{N} \pi_i\Phi_i(D_i).$$

According to the given diagnostic order, TD is computed as

$$\begin{aligned}
\text{TD}=\pi_1 &\left[\int_0^{D_1} x_1\,d\Phi_1(x_1)+(1-\Phi_1(D_1))\sum_{i=1}^{N} D_i\right]\\
&\ \vdots \qquad\qquad \vdots\\
&+\pi_k\left[\sum_{i=1}^{k-1} D_i+\int_0^{D_k} x_k\,d\Phi_k(x_k)+(1-\Phi_k(D_k))\sum_{i=k}^{N} D_i\right]\\
&\ \vdots \qquad\qquad \vdots\\
&+\pi_N\left[\sum_{i=1}^{N-1} D_i+\int_0^{D_N} x_N\,d\Phi_N(x_N)+(1-\Phi_N(D_N))D_N\right]\\
&=\sum_{i=1}^{N} \pi_i\int_0^{D_i} x_i\,d\Phi_i(x_i)+\sum_{i=1}^{N} D_i-\sum_{i=1}^{N} \pi_i\Phi_i(D_i)\sum_{j=i}^{N} D_j.
\end{aligned}\tag{4.2}$$

Let θ be a Lagrangian multiplier. The solution for D_i, $1 \le i \le N$, must satisfy the differential equation

$$\nabla\left[\text{TD}+\theta\left(M-\sum_{i=1}^{N}\pi_i\Phi_i(D_i)\right)\right]=0,$$

which is equivalent to

$$\frac{\partial \mathrm{TD}}{\partial D_1} - \pi_1\phi_1(D_1) = \frac{\partial \mathrm{TD}}{\partial D_2} - \pi_2\phi_2(D_2)$$
$$= \cdots = \frac{\partial \mathrm{TD}}{\partial D_N} - \pi_N\phi_N(D_N) = \theta. \qquad (4.3)$$

For $1 \le k \le N$,

$$\frac{\partial \mathrm{TD}}{\partial D_k} = \pi_k D_k \phi_k(D_k) + 1 - \sum_{i=1}^{k} \pi_i \Phi_i(D_i) - \pi_k\phi_k(D_k) \sum_{i=k}^{N} D_i$$
$$= 1 - \sum_{i=1}^{k} \pi_i\Phi_i(D_i) - \pi_k\phi_k(D_k) \sum_{i=k+1}^{N} D_i. \qquad (4.4)$$

Equation (4.4) is derived by the fundamental theorem of calculus that gives

$$\frac{\partial}{\partial D_k}\left[\pi_k \int_0^{D_k} x_k \, d\Phi_k(x_k)\right] = \pi_k \frac{\partial}{\partial D_k}\left[\int_0^{D_k} x_k\phi_k(x_k)\, dx_k\right]$$
$$= \pi_k D_k \phi_k(D_k),$$

and the fact

$$\sum_{i=1}^{N} \pi_i\Phi_i(D_i) \sum_{j=i}^{N} D_j = \sum_{j=1}^{N} D_j \sum_{i=1}^{j} \pi_i\Phi_i(D_i)$$

since π_i, $\Phi_i(D_i)$, and D_i are positive. From (4.3) and (4.4), we can derive for $1 \le k \le N - 1$,

$$0 = \left(\frac{\partial \mathrm{TD}}{\partial D_k} - \pi_k\phi_k(D_k)\right) - \left(\frac{\partial \mathrm{TD}}{\partial D_N} - \pi_N\phi_N(D_N)\right)$$
$$= \sum_{i=k+1}^{N} \pi_i\Phi_i(D_i) - \pi_k\phi_k(D_k)$$
$$\cdot \sum_{i=k+1}^{N} D_i - \pi_k\phi_k(D_k) + \pi_N\phi_N(D_N)$$
$$= \pi_N\phi_N(D_N) + \sum_{i=k+1}^{N} \pi_i\Phi_i(D_i)$$
$$- \pi_k\phi_k(D_k)\left(1 + \sum_{i=k+1}^{N} D_i\right)$$

which is another form of (4.1) ■

Using the following numerical procedure, the exact value of D_i's can be determined by (4.1) and the constraint $M = \Sigma_{i=1}^{N} \pi_i\Phi_i(D_i)$.

```
Step 1: Assign an initial value to $D_N$.
Step 2: Let $n := N$.
Step 3: Let $\Delta t := D_n$.
Step 4: Calculate $D_i$, $1 \le i \le n - 1$, according to (4.1).
Step 5: Calculate $m = \pi_1\Phi_1(D1) + \cdots + \pi_N\Phi_N(D_N)$.
Step 6: If $|m - M| < \epsilon$ where $\epsilon$ is a prespecified accuracy,
        then stop. Otherwise, do the following in sequence:
    1. If $(m - M)\cdot\Delta t < 0$ then let $\Delta t := -\Delta t/4$.
    2. If $D_n + \Delta t \le 0$, let $D_n := 0$, decrement $n$ by 1,
       and then go to Step 3.
    3. Let $D_n := D_n + \Delta t$ and then go to Step 4.
```

TABLE I
THREE CASES OF FAULTY PROBABILITIES

%	π_1	π_2	π_3	π_4	π_5
Case 1	20.0	20.0	20.0	20.0	20.0
Case 2	51.6	25.8	12.9	6.5	3.2
Case 3	67.0	22.3	7.4	2.5	0.8

The unidirectional search employed in the above procedure is feasible because m is a monotonically increasing function of D_i for $1 \le i \le N$. If $D_i = 0$ in the final result, it indicates that π_i is so small that the required diagnostic coverage can be achieved without even diagnosing v_i.

The value of D_N should satisfy

$$1 - \frac{1 - M}{\pi_N} < \Phi_N(D_N) < 1$$

since

$$M = \sum_{i=1}^{N-1} \pi_i\Phi_i(D_i) + \pi_N\Phi_N(D_N)$$
$$< \sum_{i=1}^{N-1} \pi_i + \pi_N\Phi_N(D_N)$$
$$= (1 - \pi_N) + \pi_N\Phi_N(D_N).$$

The above range is useful in selecting an initial value of D_N.

If M is so close to 1 that for $1 \le i \le N$, $\Phi_i(D_i) \approx 1$, $\phi_i(D_i) \ll 1$, and the tail part of $\phi_i(t)$ is approximated by the exponential function $(1/\lambda_i)e^{-t/\lambda_i}$, then the optimal diagnostic time can be approximated by

$$\frac{D_{N-1}}{\lambda_{N-1}} \approx \ln\left[\left(\frac{\pi_{N-1}}{\pi_N}\right)\left(\frac{1 + D_N}{\lambda_{N-1}}\right)\right]$$
$$\frac{D_{N-2}}{\lambda_{N-2}} \approx \ln\left[\left(\frac{\pi_{N-2}}{\pi_N + \pi_{N-1}}\right)\left(\frac{1 + D_N + D_{N-1}}{\lambda_{N-2}}\right)\right]$$
$$\vdots \qquad\qquad \vdots$$
$$\frac{D_1}{\lambda_1} \approx \ln\left[\left(\frac{\pi_1}{\pi_N + \pi_{N-1} + \cdots + \pi_2}\right)\right.$$
$$\left.\cdot\left(\frac{1 + D_N + D_{N-1} + \cdots + D_2}{\lambda_1}\right)\right].$$

From this approximation, it is easy to see that when all modules are equally likely to be faulty (i.e., $\pi_1 = \cdots = \pi_N = 1/N$), the optimal diagnostic time of each module increases exponentially with its position in the diagnostic order.

To see the effect of the faulty probabilities on the optimal diagnostic times, we consider three cases of the faulty probabilities for a five-module system which are tabulated in Table I. Diagnostic time for individual modules are assumed to be exponentially distributed with $\lambda_1 = \cdots = \lambda_5 = 100$. The choice for the units of all time variables and λ_i's are arbitrary but identical for the purpose of comparison. In Case 1, all five modules have the same faulty probability. In Case 2 (Case 3),

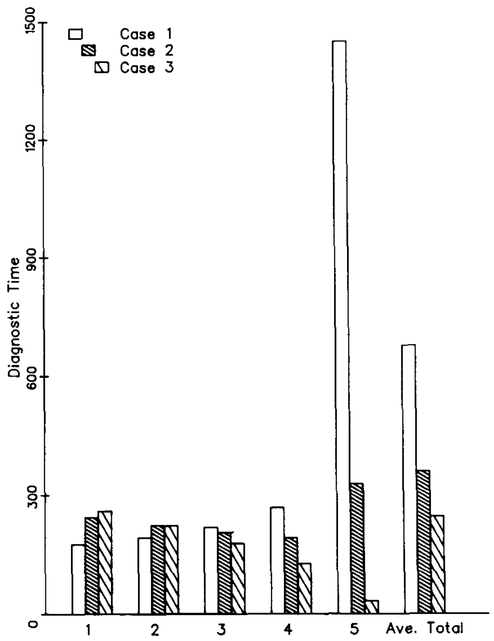

Fig. 5. The optimal diagnostic times in three cases.

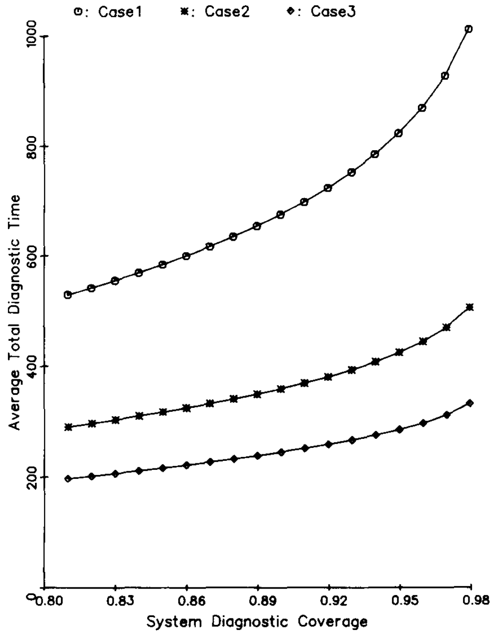

Fig. 6. Average total diagnostic time versus system diagnostic coverage in three cases.

the faulty probability of a module is twice (three times) that of the next module in the diagnostic order. Using the numerical procedure described earlier, the optimal diagnostic times for all three cases under 90 percent system diagnostic coverage are computed and plotted in Fig. 5. The last category in Fig. 5 is the average diagnostic time derived from (4.2) and the exponential distribution of diagnostic times as follows:

$$\mathrm{TD}=\sum_{i=1}^{N}\pi_i(\lambda_i-(D_i+\lambda_i)e^{-D_i/\lambda_i})$$
$$+\sum_{i=1}^{N}D_i-\sum_{i=1}^{N}\pi_i\Phi_i(D_i)\sum_{j=i}^{N}D_j.$$

Several conclusions can be drawn from Fig. 5 as follows.

- If two modules have the same faulty probability and the same distribution of diagnostic times, the one diagnosed earlier requires less diagnostic time, as evidenced in Case 1.
- Generally, the diagnostic time for any module increases (and decreases) with its faulty probability, but the amount of increase (decrease) is larger for those modules diagnosed later.
- The average total diagnostic time is shorter when there are larger disparities in the faulty probabilities. For example, Case 1 (Case 3) has the least (most) disparity in the faulty probabilities and, thus, the longest (shortest) average total diagnostic time.

In Fig. 6, the average total diagnostic times are computed for all three cases with the system diagnostic coverage ranging from 81 to 98 percent. The average total diagnostic time is observed to increase substantially when the system diagnostic coverage approaches 1.

V. Numerical Example

An example is presented here to demonstrate the use of our model for locating a faulty module. For a given distribution of each basic random parameter introduced thus far, the following will be computed for the example system:

1) the prior faulty probabilities,
2) the likelihoods of two types of fault syndromes,
3) the posterior faulty probabilities,
4) the optimal diagnostic times for individual modules.

All random parameters, except for J_{ij}'s and B_{ij}'s, are assumed to be exponentially distributed. J_{ij} is assumed to have the following density function:

$$f_{ij}^{J}(t)=\begin{cases}\zeta_{ij} & \text{if } t=0\\ (1-\zeta_{ij})(1/\kappa_{ij})e^{-t/\kappa_{ij}} & \text{if } t>0.\end{cases}$$

The discrete probability at $J_{ij} = 0$ is to describe the instantaneous detection coverage of the boundary detection mechanisms.

From our experiments on FTMP [28], B_{ij}'s are found to have multimodal distributions. Hence, the B_{ij}'s will be assumed to have a bimodal distribution, i.e.,

$$B_{ij}=\begin{cases}\mathfrak{N}(\mu_{ij}^{1},\sigma_{ij}^{1}) & \text{with probability } \eta_{ij}\\ \mathfrak{N}(\mu_{ij}^{2},\sigma_{ij}^{2}) & \text{with probability } 1-\eta_{ij}\end{cases}$$

where $\mathfrak{N}(\mu, \sigma)$ denotes a normally distributed random variable with mean μ and standard deviation σ.

The likelihoods of fault syndromes are obtained via simulation because of the analytic difficulty discussed in Section II. To increase accuracy, every independent random parameter in our model is generated by an independent random generator in the simulation. One million sets of data are generated for the example.

Two types of fault syndromes are considered: 1) the *single-detection* type where the decision time is set at the detection time of the first error-detecting module of the system, and 2)

TABLE II
DISTRIBUTION OF RANDOM PARAMETERS

Variable	Distribution	Parameters
Y_i	Exponential	Mean = 100000
L_i	Exponential	Mean = 40
I_i	Exponential	Mean = 140
J_{ij}	Hybrid	$\zeta_{ij} = 0.2$ $\kappa_{ij} = 100$
B_{ij}	Bimodal Normal	$\eta_{ij} = 0.6$ $\mu_{ij}^1 = 40,\ \sigma_{ij}^1 = 25$ $\mu_{ij}^2 = 80,\ \sigma_{ij}^2 = 60$

TABLE III
CONSTANT PARAMETERS

Module	C_i^F	C_i^P	T_i^Y	T_i^D	T_i^P
1	0.4	0.4	0	50000	99400
2	0.4	0.4	0	60000	99300
3	0.4	0.4	0	70000	99200
4	0.4	0.4	0	80000	99100
5	0.4	0.4	0	90000	99000

the *double-detection* type where the decision time is set at the detection time of the second error-detecting module of the system. The single-detection and double-detection types of fault syndromes will be written as $(\mathfrak{I}, T)$, $(\mathfrak{I}, T - dt;, \mathfrak{J}, T)$, respectively, where T is the decision time and $\mathfrak{I}$ and $\mathfrak{J}$ are the first and second error-detecting modules. To obtain the double-detection type of fault syndrome, the system has to delay diagnosis for time dt from the first error detection. Will the average total diagnostic time for the double-detection type of fault syndrome be shorter than that for the single-detection type? Our example will answer this question. Since there is a small probability that the delay for the second error-detecting module is unrealistically long, the system will stop waiting after some predetermined time. The delay is set to be no longer than 100 units of time because there are very few double-detection syndromes with a delay longer than 100 in our simulation.

The example system is represented by the graph $D1$ as shown in Fig. 3. For all nodes and edges of $D1$, the associated random parameters are assumed to have the distributions in Table II. The assumed values for the constant parameters of $D1$ are tabulated in Table III.

Among the simulation results for all single-detection and double-detection types of fault syndrome, we present only a subset of them that are deemed interesting. The prior faulty probabilities for individual modules are calculated to be $\pi_1' = 0.348$, $\pi_2' = 0.267$, $\pi_3' = 0.193$, $\pi_4' = 0.126$, and $\pi_5' = 0.066$. The posterior faulty probabilities for syndromes $(4, T - dt; 1, T)$, $(4, T - dt; 2, T)$, $(4, T - dt; 3, T)$, and $(4, T - dt; 5, T)$ are plotted in Figs. 7-10 where the point for $dt = 0$ is actually the result of syndrome $(4, T)$.

As shown in Fig. 7 where v_4 is the first error-detecting module of the system, its faulty probability (about 0.5) is the highest among all the modules in $D1$. However, if v_1 subsequently detects an error, v_1's faulty probability becomes the highest at about 0.9, because it is much easier for an error to propagate from v_1 to v_4 than from v_4 to v_1. In Fig. 8, where v_2 will detect an error after v_4, if the delay $dt < 60$, v_2 has the highest faulty probability; but, if the delay $dt \geq 60$, v_4 again

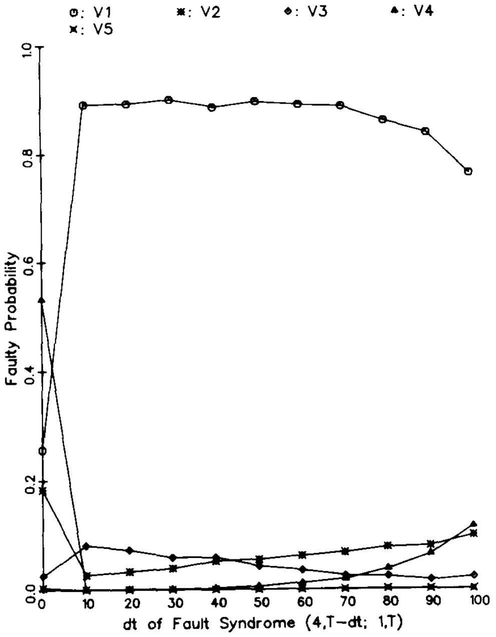

Fig. 7. The posterior faulty probabilities for fault syndrome (4, $T - dt$; 1, T) in $D1$.

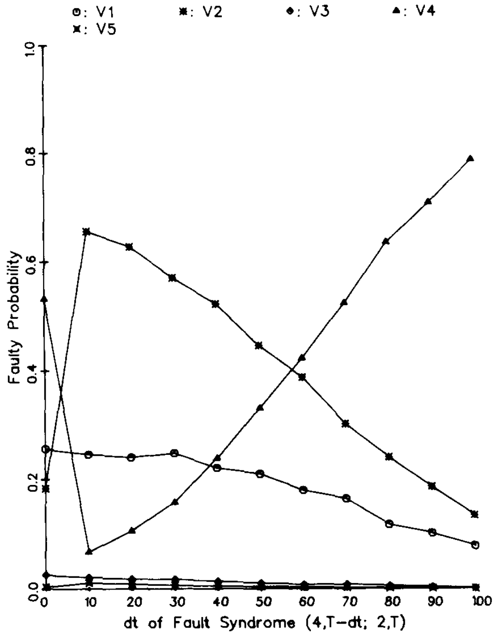

Fig. 8. The posterior faulty probabilities for fault syndrome (4, $T - dt$, 2, T) in $D1$.

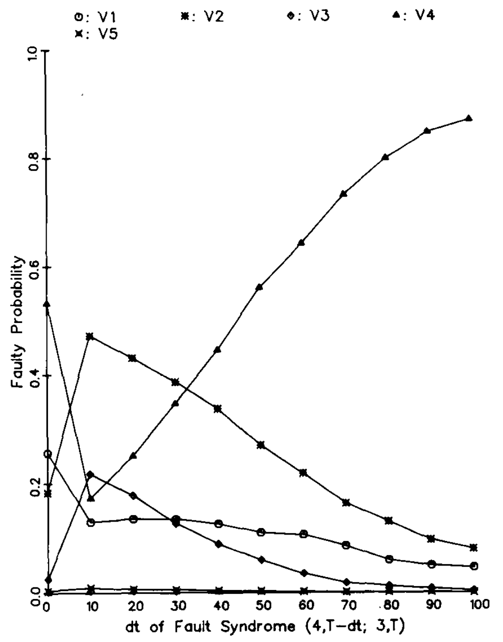

Fig. 9. The posterior faulty probabilities for fault syndrome (4, $T - dt$; 3, T) in $D1$.

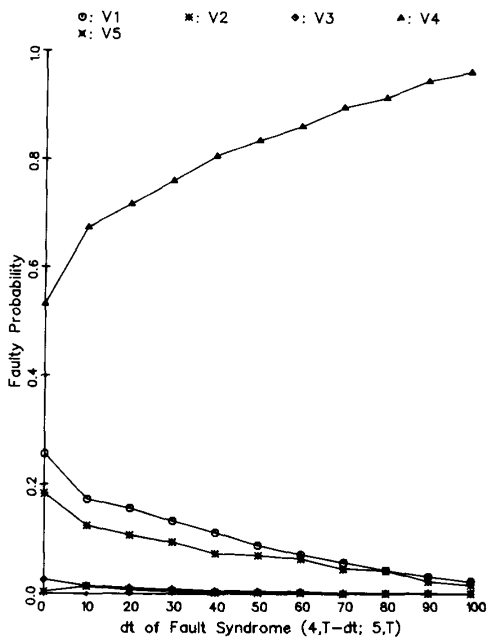

Fig. 10. The posterior faulty probabilities for fault syndrome (4, $T - dt$; 5, T) in $D1$.

has the highest faulty probability. This is because errors propagating from v_4 to v_2 via v_5 take about twice the time than propagating from v_2 to v_4 and the detection coverages of v_2 and v_4 are the same. Note that v_1 also has a moderate faulty probability since errors are also likely to elude the detection mechanisms in v_1 and propagate to v_2 and v_4. In Fig. 9, if dt is small, all modules except v_2 would have some nonzero faulty probabilities, but when dt becomes larger, v_4 will dominate others in the faulty probability. Fig. 10 is typical for those syndromes where the first error-detecting module has an output link to the second error-detecting module. The longer the delay, the higher the faulty probability of the first error-detecting module will result.

Fig. 11 compares the average total diagnostic times for all the syndromes shown in Figs. 7–10. It shows that not all double-detection fault syndromes lead to shorter average diagnostic times than single-detection fault syndromes do. There is the time delay (dt) penalty for the double-detection type. Then, how can we justify the use of the double-detection type? Given that v_4 has made the first error detection, the likelihoods of obtaining those different double-detection fault syndromes are derived and listed in Table IV. It is apparent that a double-detection fault syndrome may increase the average total diagnostic time over the single-detection type only with a probability of 0.091. To evaluate these two types of faulty syndromes further, we calculate and compare the time overheads each of which is defined as the mean time from the first error detection until the end of diagnosis. The results are listed in Table V. It can be concluded that if the first error detection is made in v_1, v_2, or v_3, it is better to diagnose the system immediately; otherwise, it would be better to wait for a second error detection.

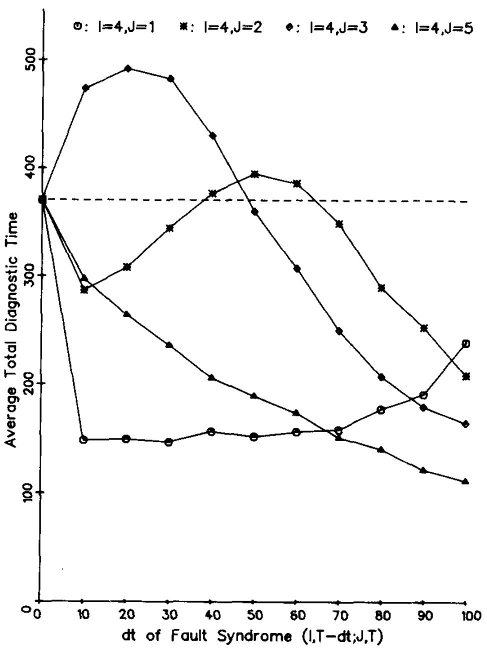

Fig. 11. Average total diagnostic times for fault syndromes with v_4 being the first error-detecting module in $D1$.

VI. Remarks

System level fault diagnosis has been dealt with in numerous papers [1], [5], [10], [11], [18], [19], [22], [29], [24], [23]. The models used/proposed in these papers will collectively be referred to as the PMC model. The basic framework of the PMC model is a system consisting of modules which are

TABLE IV
THE LIKELIHOODS OF GETTING DIFFERENT FAULT SYNDROMES

Fault Syndrome	Likelihood
$(4, T - dt; 1, T)$	19.8%
$(4, T - dt; 2, T)$	26.2%
$(4, T - dt; 3, T)$	9.1%
$(4, T - dt; 5, T)$	44.9.%

TABLE V
TIME OVERHEADS FOR SINGLE- AND DOUBLE-DETECTION FAULT SYNDROMES IN $D1$.

First error-detecting module	single-detection time overhead	double-detection time overhead
1	155.15	188.18
2	218.33	232.14
3	221.01	239.54
4	370.28	283.01
5	358.27	318.91

capable of diagnosing other modules via available test links between modules. The object of the PMC model is to identify the faulty module(s) using the test results from all modules. Interesting problems in the PMC model are 1) determination of the number of faulty modules which can be identified for a given set of test links, 2) assignment of test links to be able to diagnose t faulty modules, and 3) development of algorithms to identify faulty modules from test results.

There are three major differences between our model and the PMC model as discussed below. The first difference lies in our assumption that an independent fault-free system is available to diagnose all modules. In our model, a module can be defined at a lower level since it is not required to diagnose other modules and there must be test links from the independent system to all modules. In the PMC model, an independent fault-free system is also required to collect and analyze the test results. The second difference is that we consider fault diagnosis as the next step to error detection while in the PMC model fault diagnosis does not consider any interplay between different phases of fault tolerance. We can therefore use the fault syndrome to determine the posterior faulty probabilities and the diagnostic order. In an extension of the PMC model [15], the failure rates of individual modules are considered. The third difference is in the assumption on the diagnosis itself. The PMC model assumes that every diagnosis is complete, i.e., a nonfaulty module can always diagnose a faulty module. This unrealistic assumption is relaxed somewhat in [3] and [2] where incomplete diagnosis is considered. In our model, not only that all diagnoses are incomplete but the diagnostic coverage is assumed to be an increasing function of the diagnostic time.

Although our model and the PMC model have different assumptions and approaches, they could complement each other. One interesting problem is how to minimize the diagnosis time for a PMC model, using partial test results and sequential decision theory.

Another approach to system level fault diagnosis is based on comparison [6], [8], [14]. The idea behind this approach is to detect and identify the faulty units with user tasks. Assuming that a set of identical processing units are available, a task (or job) will be executed on a selected pair of processing units and the results from both units are compared. Disagreement between the results indicates that one of the processing units is faulty. The faulty processing unit may be identified when a sufficient number of comparisons are made. Although the comparison can be carried out by hardware matchers [17] or any processing unit other than the units that had executed the task [14], an independent system is required to collect and analyze the comparison results. The advantages of this approach are that 1) it covers transient and intermittent faults, 2) it does not have the hardware and software overheads of detection mechanisms and diagnostic programs. It can even utilize the idle time of the processing units to execute duplicated tasks [7]. However, usefulness of the comparison approach is limited because a) it only covers the faults in the processing units, b) it may take a very long time to diagnose the faulty units executing user tasks, and c) it does not consider intertask communications and hence error propagation among tasks.

A fault can be diagnosed only if it is active during diagnosis. When a fault syndrome is obtained, we do not know whether it is the result of a transient, intermittent, or permanent fault. To derive the optimal diagnostic time for each module, the system diagnostic coverage is determined first under the assumption that the fault is permanent. The determination of this diagnostic coverage is an interesting problem in its own right, which is related to system reconfiguration and error recovery. If no faults are found after diagnosing all the modules, the fault could be transient or intermittent or permanent but nondiagnosable. At that time, the posterior faulty probabilities can still be used, in deciding what to do next.

In this paper, we assumed that there is at most one faulty module in the system, i.e., the single faulty module assumption. (There could be multiple faults in the faulty module, though.) To extend our model to the case of multiple faulty modules, the number of hypotheses in the parameter space of the Bayesian decision problem has to be increased as follows:

$$H_{ij} : v_i \text{ and } v_j \text{ are the faulty modules for } 1 \leq i < j \leq N,$$

$$H_{ijk} : v_i,\ v_j, \text{ and } v_k \text{ are the faulty modules for } 1 \leq i < j \leq N,$$

$$\vdots \qquad \vdots$$

Then, the same approach for the case of single faulty module can be used to obtain the posterior likelihood of each hypothesis being true. However, the complexity in calculating these likelihoods will increase dramatically.

The main issue in using our model for real systems is how to collect the necessary statistics and estimate the distributions of the parameters introduced in our model. This issue is discussed below from two different viewpoints. First, the difficulty in obtaining the information on input parameters is inevitable and is the price to pay for any realistic model like ours. As mentioned in Section II, we have already developed methodologies to measure direct propagation times and the distribution of fault latencies on real systems [27], [28]. The error latency and coverage have also been measured by some other researchers [9], [25] to evaluate different detection mechanisms. Similar methodologies are expected to be developed in the future to monitor the behavior of faults and errors. It is also important to note that the parameters introduced in our model are also needed to model other phases of fault tolerance.

Second, distributions of the random parameters in our model do not have to be exact to be useful. The faulty module is actually diagnosed by the diagnostic programs applied to each module. All these distributions are used to decide the diagnostic order and the diagnostic time. It is meaningless to verify that a parameter follows a certain distribution, because the distribution is just a summary of our past information about the parameter. If we know little about a parameter, a uniform distribution will be used for the parameter. If a time parameter is more likely to be short than long, then an exponential distribution can be used for the parameter. This applies to the determination of the prior faulty probabilities as well as other parameters. The key idea is to make the best use of available information for fault diagnosis, instead of assuming the perfect (or very accurate) information about faults and errors.

VII. Conclusion

In this paper, we have formulated and solved a new problem of locating a faulty module. The distinguishing characteristics of our model as compared to other models on system level fault diagnosis are the consideration of 1) the information from other phases of fault tolerance, 2) the time overheads of fault diagnosis, and 3) the imperfectness of diagnostic coverage.

Based on the likelihood principle, the faulty probabilities for individual modules have been calculated from parameters of fault detection, periodic diagnostic, error propagation, and error detection. The optimal diagnostic order of modules is then found to be the decreasing order of the faulty probabilities. Given a system diagnostic coverage and the optimal diagnostic order, the optimal diagnostic time for each module is then determined to minimize the average total diagnostic time. A numerical example is presented and discussed, and the following conclusions have been drawn. First, the disparity in the modules' faulty probabilities has a great effect on shortening the average total diagnostic time. Second, in most cases the faulty probabilities show large disparities so that it is worthwhile to compute the faulty probabilities for individual modules before actual diagnosis. Third, delaying diagnosis until the second error detection can sometimes improve the total diagnostic time overhead. The determination of the optimal delay period for diagnosis purpose is an interesting future problem.

Appendix

List of Variables

F_i^V	Distribution function of the random variable V_i.
f_i^V	Density function of the random variable V_i
F_{ij}^V	Distribution function of the random variable V_{ij}.
f_{ij}^V	Density function of the random variable V_{ij}.
Y_i	Fault cycle of v_i.
L_i	Fault latency of v_i.
C_i^F	Coverage of fault detection in v_i.
C_i^P	Coverage of the periodic diagnostic in v_i.
B_{ij}	Direct propagation time from v_i to v_j.
X_{ij}	Error propagation time from v_i to v_j.
I_i	Internal detection latency in v_i
J_{ij}	Boundary detection latency on e_{ij}.
K_i	Detection latency in v_i.
H_i	Hypothesis of v_i being the faulty module.
π_i'	Prior Prob$[H_i]$.
π_i	Posterior Prob$[H_i]$.
$\mathcal{S}$	Fault syndrome.
$\mathcal{L}_i(\mathcal{S})$	Fault syndrome's likelihood function if H_i is true.
E_{ij}	Error latency for the error originated from v_i and detected in v_j.
T_i^Y	Last time a repair was done on v_i.
T_i^D	Last time a thorough diagnostic was done on v_i.
T_i^P	Last time a periodic diagnostic was done on v_i.
T_i^C	The contaminating time of v_i.
T_i^F	The faulty time of v_i.
T	The decision time of the system.
Φ_i	Percentage of faults that the diagnostic can uncover in v_i and $\phi_i = d\Phi_i/dt$.
D_i	Optimal diagnostic time for v_i.

References

[1] F. Barsi, F. Grandoni, and P. Maestrini, "A theory of diagnosability without repairs," *IEEE Trans. Comput.*, vol. C-25, no. 6, pp. 585-593, June 1976.

[2] M. L. Blount, "Modeling of diagnosis fail-softly computer systems," in *Dig. Papers FTCS-8*, June 1978, pp. 53-58.

[3] ——, "Probabilistic treatment of diagnosis in digital systems," in *Dig. Papers FTCS-7*, June 1977, pp. 72-77.

[4] D. C. Bossen and M. Y. Hsiao, "Model for transient and permanent error-detection and faulty isolation coverage," *IBM J. Res. Develop.*, vol. 26, no. 1, pp. 67-77, Jan. 1982.

[5] K. Y. Chwa and S. L. Hakimi, "On fault identification in diagnosable systems," *IEEE Trans. Comput.*, vol. C-30, no. 6, pp. 414-422, June 1981.

[6] A. T. Dahbura and G. M. Masson, "Greedy diagnosis as the basis of an intermittent-fault/transient-upset tolerant system design," *IEEE Trans. Comput.*, vol. C-32, no. 10, pp. 953-957, Oct. 1983.

[7] A.T. Dahbura and K. K. Sabnani, "Performance analysis of a fault detection scheme in multiprocessor systems," *Perform. Eval. Rev.*, pp. 143-154, May 1987.

[8] A. T. Dahbura, K. K. Sabnani, and L. L. King, "The comparison approach to multiprocessor fault diagnosis," in *Dig. Papers, FTCS-15*, June 1985, pp. 260-265.

[9] A. Damm, "The effectiveness of software error-detection mechanisms in real-time operating systems," in *Dig. Papers, FTCS-16*, June 1986, pp. 171-176.

[10] S. L. Hakimi and A. T. Amin, "Characterization of connection assignment of diagnosable systems," *IEEE Trans. Comput.*, vol. C-23, pp. 86-88, Jan. 1974.

[11] S. L. Hakimi and K. Nakajama, "On adaptive system diagnosis," *IEEE Trans. Comput.*, vol. C-33, no. 3, pp. 234-240, Mar. 1984.

[12] B. E. Helvik, "Periodic maintenance, on the effect of imperfectness," in *Dig. Papers FTCS-10*, June 1980, pp. 204-206.

[13] J. C. Laprie, "Dependable computing and fault tolerance: Concepts and terminology," in *Dig. Papers FTCS-15*, June 1985, pp. 2-11.

[14] J. Maeng and M. Malek, "A comparison connection assignment for self-diagnosis of multiprocessor systems." in *Dig. Papers FTCS-11*, June 1981, pp. 173-175.

[15] S. N. Maheshwari and S. L. Hakimi, "On models for diagnosable systems and probabilistic fault diagnosis," *IEEE Trans. Comput.*, vol. C-25, no. 3, pp. 228-236, Mar. 1976.

[16] S. V. Makam and A. Avizienis, "Modelling and analysis of periodically renewed closed fault-tolerant systems," in *Dig. Papers FTCS-11,* June 1981, pp. 134-141.

[17] M. Malek, "A comparison connection assignment for diagnosis of multiprocessor systems," in *Proc. 7th Symp. Comput. Architecture,* May 1980, pp. 31-35.

[18] S. Mallela and G. M. Masson, "Diagnosable systems for intermittent faults," *IEEE Trans. Comput.,* vol. C-27, no. 6, pp. 560-566, June 1978.

[19] ——, "Diagnosis without repairs for hybrid fault situations," *IEEE Trans. Comput.,* vol. C-29, no. 6, pp. 461-470, June 1980.

[20] T. Nakagawa, "Optimum policies when preventive maintenance is imperfect," *IEEE Trans. Reliability,* vol. R-28, no. 4, pp. 331-332, Oct. 1979.

[21] T. Nakagawa, K. Yasui, and S. Osaki, "Optimum maintenance policies for a computer system with restart," in *Dig. Papers FTCS-11,* June 1981, pp. 148-150.

[22] F. P. Preparata, G. Metze, and R. T. Chien, "On the connection assignment problem of diagnosable systems," *IEEE Trans. Electron. Comput.,* vol. EC-16, pp. 848, Dec. 1967.

[23] J. D. Russell and C. R. Kime, "System fault diagnosis: Closure and diagnosability without repairs," *IEEE Trans. Comput.,* vol. C-24, no. 11, pp. 1078-1089, Nov. 1975.

[24] ——, "System fault diagnosis: Masking, exposure and diagnosability without repairs," *IEEE Trans. Comput.,* vol. C-24, no. 12, pp. 1151-1161, Dec. 1975.

[25] M. A. Schuette, J. P. Shen, D. P. Siewiorek, and Y. X. Zhu, "Experimental evaluation of two concurrent error detection schemes," in *Dig. Papers FTCS-16,* June 1986, pp. 138-143.

[26] K. G. Shin and Y.-H. Lee, "Error detection process—Model, design, and its impact on computer performance," *IEEE Trans. Comput.,* vol. C-33, no. 6, pp. 529-540, June 1984.

[27] ——, "Measurement and application of fault latency," *IEEE Trans. Comput.,* vol. C-35, no. 4, pp. 370-375, Apr. 1986.

[28] K. G. Shin and T.-H. Lin, "Modeling error propagation in a multimodule computing system," *IEEE Trans. Comput.,* vol. C-37, no. 9, pp. 1053-1066, Sept. 1988.

[29] A. K. Somani, V. K. Agarwal, and D. Avis, "A generalized theory for system level diagnosis," *IEEE Trans. Comput.,* vol. C-36, no. 5, pp. 538-546, May 1987.

[30] N. N. Tendolkar and R. L. Swann, "Automated diagnostic methodology for the IBM 3081 processor complex," *IBM J. Res. Develop.,* vol. 26, no. 1, pp. 78-88, Jan. 1982.

[31] Y. W. Yak, T. S. Dillon, and K. E. Forward, "The effect of imperfect periodic maintenance on fault-tolerant computer systems," in *Dig. Papers FTCS-14,* June 1984, pp. 66-70

[32] Y. W. Yak, T. S. Dillon, and K. E. Forward, "Incorporation of recovery and repair time in the reliability modeling of fault-tolerance systems," in *Proc. IEE/IFAC SAFECOMP,* 1983, pp. 45-52.

Tein-Hsiang Lin (S'83-M'88) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Republic of China, in 1980, the M.S. degree in electrical engineering from Iowa State University, Ames, in 1984, and the Ph.D. degree in computer, information, and control engineering from the University of Michigan, Ann Arbor, in 1988.

He is currently an Assistant Professor in the Department of Electrical and Computer Engineering, The State University of New York at Buffalo. His research interests include fault-tolerant computing, real-time distributed systems, and performance evaluation.

Kang G. Shin (S'75-M'78-SM'83), for a photograph and biography, see p. 18 of the January 1990 issue of this TRANSACTIONS.