A BAYESIAN APPROACH TO FAULT CLASSIFICATION[1]

Tein-Hsiang Lin† and Kang G. Shin‡

†Department of Electrical and Computer Engineering
State University of New York at Buffalo, Buffalo, New York

‡Real-Time Computing Laboratory
Department of Electrical Engineering and Computer Science
The University of Michigan, Ann Arbor, Michigan

ABSTRACT

According to their temporal behavior, faults in computer systems are classified into permanent, intermittent, and transient faults. Since it is impossible to identify the type of a fault upon its first detection, the common practice is to retry the failed instruction one or more times and then use other fault recovery methods, such as rollback or restart, if the retry is not successful. To determine an "optimal" (in some sense) number of retries, we need to know several fault parameters, which can be estimated only after classifying all the faults detected in the past.

In this paper we propose a new fault classification scheme which assigns a fault type to each detected fault based on its detection time, the outcome of retry, and its detection symptom. This classification procedure utilizes the Bayesian decision theory to sequentially update the estimation of fault parameters whenever a detected fault is classified. An important advantage of this classification is the early identification of presence of an intermittent fault so that appropriate measures can be taken before it causes a serious damage to the system. To assess the goodness of the proposed scheme, the probability of incorrect classification is also analyzed and compared with simulation results.

Index Terms — Fault classification, prior and posterior distributions, parameter estimation, retry, Bayesian decision theory, probability of incorrect classification.

1 INTRODUCTION

As defined in [1] and [2], a fault in a computer system is the source of an error while an error is defined as a logically incorrect state of the system. Based on its temporal effect, a fault can be classified to be *permanent* or *intermittent* or *transient* [3, 4]. Let the *active period* of a fault represent the time period during which the fault has adverse effects on the system (i.e., the fault if actived may induce errors), and let the *benign period* of a fault be the time period when the fault is not active. Then, a permanent fault has an infinite active period, a transient fault has a finite active period followed by an infinite benign period, and an intermittent fault cycles between a finite active period and a finite benign period.

Upon detection of a permanent fault, the system must be reconfigured to remove the faulty module as soon as possible. For a non-permanent fault, the system can retry or re-execute the latest instruction which could not be completed because of the fault. During the active period of a fault, retry will be unsuccessful, and thus, may need another retry. Retry for non-permanent faults will eventually be successful and the system will then continue its normal operation as if nothing had happened. Retry works very well for transient faults, but is sometimes not desirable at all for intermittent faults since they may recur an infinite number of times in future. On the other hand, retry may sometimes be useful for an intermittent fault when a task can be completed before the recurrence of the intermittent fault. The best strategy for intermittent faults depends upon the system's goal and is usually difficult to determine. The choice between reconfiguration and retry upon a fault detection is not easy to make either, because it is impossible to know *a priori* the type of the detected fault. A compromised solution is to use retry one or more times and then use reconfiguration only after retry became unsuccessful. This strategy works well because the majority of faults are known to be non-permanent [5, 6]. Thus, how to determine an "optimal" (in some sense) number of retries becomes an important research problem [7, 8, 9]. Since a finite number of retries can be easily converted to a retry period and vice versa [7], "retry period" (in place of "number of retries") will be used throughout the paper.

The most influential variables in determining an optimal retry period are the parameters related to the characteristics of faults, such as the fault occurrence rate, the mean active period of a transient or intermittent fault, and the probability of a detected fault being permanent, intermittent, or transient. These parameters are commonly assumed to be known *a priori*. Hence, the results obtained under such an assumption are useful only if fault parameters are known somehow or can be estimated accurately. No previous work except for [7] has addressed the problem of deriving retry policies in the absence of *a priori* knowledge about the fault parameters. Lee and Shin [7] proposed a Bayesian decision approach where the active and benign periods of an intermittent fault are estimated on-line and the estimated parameters are then used to derive the optimal retry periods

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.
© 1990 ACM 089791-359-0/90/0005/0058 $1.50

for future recurrences of the intermittent fault. However, they did not address how to estimate other fault parameters, and the criterion in recognizing an intermittent fault was *ad hoc*. That is, if two consecutive fault detections are made within a *short* period and their symptoms are similar, they are assumed to be two manifestations of the same intermittent fault.

In this paper, we shall develop a method to estimate *all* fault parameters on-line. A Bayesian approach [10, 11] is employed which has the advantage of estimating parameters progressively from sample data. The sample data for the fault parameters are obtained by monitoring the fault behavior through retry and detection mechanisms. Since faults are classified into three different types which are indistinguishable by fault detection mechanisms, a critical issue in parameter estimation is how to recognize the types of detected faults so that the related sample data can be interpreted correctly. As a resolution to this issue, a new fault classification scheme is proposed herein which estimates and assigns a fault type for each detected fault with a very high accuracy.

Throughout this paper, the term "fault detection" will be used to mean the detection of manifestation of a fault with signal-level detection mechanisms [1], such as built-in testers and error detecting codes. This term is adopted because in such a case a fault can be detected and located immediately upon its manifestation. Further, faults are also assumed to be detected immediately upon their occurrence. If faults cannot be detected upon their occurrence, instruction retry is no longer effective, since it is impossible to determine which instruction to retry.

The subject of parameter estimation is treated in Section 2. The proposed fault classification scheme is described in Section 3. In Section 4, the accuracy of the proposed fault classification scheme (i.e., the probability of correct classification) is analyzed and compared with simulation results. The paper concludes with Section 5.

2 PARAMETER ESTIMATION

Let $\mathcal{P}$ be the set of all fault parameters which may affect the retry mechanism. Our principal concern is whether a detected fault is permanent, intermittent, or transient, and hence, the fault parameters are related to the generation of different types of faults. Arrivals of permanent, intermittent, and transient faults are assumed to be independent Poisson processes with rate λ_p, λ_i, and λ_t, respectively. Thus, the intervals between two consecutive permanent, intermittent, and transient faults, denoted by Y_p, Y_i, and Y_t, respectively, are independent random variables with negative exponential distributions. The independence assumption is justified by the fact that causes for different types of faults are different and usually uncorrelated. Permanent faults result mainly from components aging, transient faults are induced mainly by temporary changes in environmental, electrical, or mechanical conditions, and intermittent faults are caused mainly by manufacturing defects such as loose connections or bonds. It is also assumed that the active period of a transient fault, denoted by A_t, and the active and benign periods of an intermittent fault, denoted by A_i and B_i, respectively, are distributed exponentially with rates λ_{ta}, λ_{ia}, and λ_{ib}, respectively. Hence, we have $\mathcal{P} \equiv \{\lambda_p, \lambda_t, \lambda_i, \lambda_{ta}, \lambda_{ia}, \lambda_{ib}\}$.

Samples of the fault parameters can be obtained from fault detection and retry mechanisms. The detection mechanism reveals the existence of a fault, whereas the retry mechanism indicates the active period of a fault. In addition to the knowledge of the existence and the active period of a fault, the type (permanent or intermittent or transient) of the fault must be known. An on-line method of parameter estimation is proposed below using a Bayesian approach. The issue of fault classification will be dealt with in the next section.

In the Bayesian analysis, the uncertainty about a parameter is quantified by assigning a distribution to the parameter. Let $\mathbf{p} = (\lambda_p, \lambda_t, \lambda_i, \lambda_{ta}, \lambda_{ia}, \lambda_{ib})$ and let $\pi(\mathbf{p})$ represent the joint density function of $\mathbf{p}$, then

$$\pi(\mathbf{p}) = \pi_p(\lambda_p)\pi_t(\lambda_t)\pi_i(\lambda_i)\pi_{ta}(\lambda_{ta})\pi_{ia}(\lambda_{ia})\pi_{ib}(\lambda_{ib}),$$

where $\pi_\ell(\lambda_\ell)$, $\ell = p, t, i, ta, ia, ib$, represents the density function of each corresponding parameter. Each fault parameter can be estimated separately since it is independent of others. The estimation procedures for all fault parameters are identical since each of them represents the rate of an exponential distribution. In the following, the notation λ is used to represent any of the six parameters in $\mathcal{P}$.

The distribution function $\pi(\lambda)$ is updated whenever a new sample of λ is obtained. A better nomenclature would be $\pi^0(\lambda)$, $\pi^1(\lambda)$, $\pi^2(\lambda)$, ..., representing the density function before using any sample, the function after using the first sample, the function after using the second sample, and so on, since samples are obtained sequentially. Let x_i be the i-th sample. According to the Bayesian theorem,

$$\pi^i(\lambda) = \frac{\pi^{i-1}(\lambda) f(x_i|\lambda)}{\int \pi^{i-1}(\lambda) f(x_i|\lambda)\, d\lambda} \tag{2.1}$$

where $f(x_i|\lambda)$ is the sample probability given the parameter λ. In Eq. (2.1), $\pi^{i-1}(\lambda)$ ($\pi^i(\lambda)$) will be called the *prior* (*posterior*) distribution of λ with respect to x_i. The posterior distribution for the current sample is the prior distribution for the next sample. The first prior distribution (i.e., $\pi^0(\lambda)$) is determined from experiences[2], but all the other distributions will be calculated from sample information. To simplify the calculation from sample to sample, it is desirable that both the prior and the posterior distributions belong to the same distribution family. In such a case, Eq. (2.1) can be reduced to a transformation on the key parameters of the distribution. For a class of sample density functions, any class of prior density functions which has the above desirable property is called a *conjugate family* [10]. The con-

[2] If no information is available, the non-informative prior distribution will be used for $\pi^0(\lambda)$.

jugate prior distribution family for the class of exponential sample distributions is derived below.

Let $\pi(\lambda)$ and $\pi^*(\lambda)$ denote the prior and posterior density functions of λ with respect to a sample $X = x$ where X is exponentially distributed with rate λ. The sample $X = x$ is called a *direct sample.* Another type of samples to be considered is the *censored samples* which are in the form of $X > y$.

For a direct sample $X = x$, its sample probability is

$$f(x|\lambda) = \lambda e^{-\lambda x}.$$

By Eq. (2.1), the relation between $\pi(\lambda)$ and $\pi^*(\lambda|X = x)$ can be expressed as

$$\pi^*(\lambda|X = x) = M \cdot \lambda e^{-\lambda x} \cdot \pi(\lambda), \qquad (2.2)$$

where M is a term independent of λ. If $\pi^*(\lambda|X = x)$ and $\pi(\lambda)$ belong to the same distribution family, then $\pi(\lambda) \propto \lambda^p e^{-q\lambda}$, where p and q are positive real numbers yet to be determined. The only distribution which has the above property is the Gamma distribution. Therefore, the conjugate prior distribution for the exponential sample distribution is the Gamma distribution.

Suppose $\pi(\lambda)$ is the Gamma distribution $\mathcal{G}(a, b)$, where a and b are its two parameters. The density function of $\mathcal{G}(a, b)$ is

$$f_{\mathcal{G}}(z) = \frac{1}{\Gamma(a)} b^a z^{a-1} e^{-bz}, \qquad z \geq 0,$$

where $\Gamma(a)$ is the Gamma function defined as

$$\Gamma(a) = \int_0^\infty e^{-u} u^{a-1}\, du.$$

It is easy to see from Eq. (2.2) that

$$\pi^*(\lambda|X = x) \quad \propto \quad \lambda^a e^{-(x+b)\lambda},$$

meaning that the posterior distribution for a direct sample $X = x$ is $\mathcal{G}(a + 1, b + x)$.

For a censored sample $X > y$, the sample probability is

$$\int_y^\infty f(x|\lambda)\, dx = \int_y^\infty \lambda e^{-\lambda x}\, dx = e^{-\lambda y},$$

and hence,

$$\pi^*(\lambda|X > y) \quad \propto \quad \lambda^{a-1} e^{-(y+b)\lambda},$$

which is the density function of $\mathcal{G}(a, b + y)$.

Let $x_1, \cdots, x_n$ denote n direct samples and $y_1, \ldots, y_n$ denote m censored samples. Then, by induction, $\pi^*(\lambda|x_1, \cdots, x_n, y_1, \cdots, y_m)$ is the density function of

$$\mathcal{G}\left(a + n\ ,\ b + \sum_{i=1}^n x_i + \sum_{i=1}^m y_i\right)$$

where $\mathcal{G}(a, b)$ is the initial prior distribution of λ. Let $\hat{\lambda}^*$ and $\bar{\lambda}^*$ denote the mode and the mean of the posterior distribution, then

$$\hat{\lambda}^* = \frac{a + n - 1}{b + \sum_{i=1}^n x_i + \sum_{i=1}^m y_i},$$

$$\bar{\lambda}^* = \frac{a + n}{b + \sum_{i=1}^n x_i + \sum_{i=1}^m y_i},$$

and the variance of the posterior distribution is

$$\frac{a + n}{(b + \sum_{i=1}^n x_i + \sum_{i=1}^m y_i)^2}.$$

As the number of samples increases, $\hat{\lambda}^*$ and $\bar{\lambda}^*$ becomes less dependent on the initial prior distribution, and the variance gets smaller.

3 FAULT CLASSIFICATION

The true type of a fault is unrecognizable upon its first detection, but can be surmised fairly accurately after retry and/or diagnosis. In some cases, however, it could be very difficult or impossible to identify the true fault type even after retry and/or diagnosis. For example, a transient fault with a long active period may be indistinguishable from a permanent fault, and two consecutive transient faults with the same symptom are indistinguishable from two occurrences of an intermittent fault. To estimate parameters from fault samples, we propose a fault classification scheme which unambiguously assigns a temporal type to every detected fault. The assigned fault type may not always be correct, but the probability of incorrect classification or *classification error* will be minimized using the maximum likelihood principle.

Let ω_i denote the i-th detected fault whose sample information is represented by a four-tuple (Td_i, S_i, Tr_i, I_i), where Td_i denotes the detection time of ω_i, S_i the detection symptom of ω_i, Tr_i the retry period for ω_i, and I_i a flag indicating whether retry for ω_i is successful ($I_i = 0$) or not ($I_i = 1$). The detection symptom of ω_i refers to the observable, incorrect system state, which has led to the detection of ω_i. Different faults may produce the same detection symptom but for the sake of classification, different detection symptoms are assumed to be produced by different faults. Let $h(S_i)$ represent the likelihood that any two different faults will result in the same detection symptom S_i. The fault type assigned to ω_i is denoted by Type(ω_i) whose possible values are **pf**, **tf**, **if1**, and **if2**, representing the permanent fault, the transient fault, the first occurrence of an intermittent fault, and the recurrence of an intermittent fault, respectively.

A complete retry procedure is shown in Fig. 1. I_{flag} in Fig. 1 is used to indicate the existence of an intermittent fault. If $I_{\text{flag}} = 0$ at the time of detection of ω_i, the module will start the before-retry classification procedure (Fig. 2) prior to the retry for ω_i. One important function of the

$$\mathrm{IT}_{\mathrm{ratio}} = \frac{P[\mathtt{if1},\mathtt{if2}]}{h(S_i)\cdot(P[\mathtt{tf},\mathtt{tf}]+P[\mathtt{tf},\mathtt{pf}]+P[\mathtt{tf},\mathtt{if1}])} = \frac{\lambda_{ib}\lambda_i e^{-\lambda_{ib}(Td_i-Td_{i-1}-Tr_{i-1})}}{h(S_i)\lambda_t(\lambda_p+\lambda_t+\lambda_i)e^{-\lambda_i(Td_i-Td_{i-1})}}, \tag{3.1}$$

$$P[\mathtt{if1},\mathtt{if2}] = \lambda_i e^{-\lambda_i(Td_{i-1}-T_I)}\lambda_{ib}e^{-\lambda_{ib}(Td_i-Td_{i-1}-Tr_{i-1})}e^{-\lambda_t(Td_i-T_T)}e^{-\lambda_p(Td_i-T_P)} \tag{3.2}$$

$$P[\mathtt{tf},\mathtt{tf}] = \lambda_t e^{-\lambda_t(Td_{i-1}-T_T)}\lambda_t e^{-\lambda_t(Td_i-Td_{i-1})}e^{-\lambda_i(Td_i-T_I)}e^{-\lambda_p(Td_i-T_P)} \tag{3.3}$$

$$P[\mathtt{tf},\mathtt{pf}] = \lambda_t e^{-\lambda_t(Td_{i-1}-T_T)}\lambda_p e^{-\lambda_p(Td_i-T_P)}e^{-\lambda_i(Td_i-T_I)}e^{-\lambda_t(Td_i-Td_{i-1})} \tag{3.4}$$

$$P[\mathtt{tf},\mathtt{if1}] = \lambda_t e^{-\lambda_t(Td_{i-1}-T_T)}\lambda_i e^{-\lambda_i(Td_i-T_I)}e^{-\lambda_t(Td_i-Td_{i-1})}e^{-\lambda_p(Td_i-T_P)} \tag{3.5}$$

$$\mathrm{ITT}_{\mathrm{ratio}} = \frac{P[\mathtt{if1},\mathtt{tf},\mathtt{if2}]}{h(S_i)\cdot(P[\mathtt{tf},\mathtt{if1},\mathtt{tf}]+P[\mathtt{tf},\mathtt{if1},\mathtt{pf}]+P[\mathtt{tf},\mathtt{tf},\mathtt{tf}]+P[\mathtt{tf},\mathtt{tf},\mathtt{pf}]+P[\mathtt{tf},\mathtt{tf},\mathtt{if1}])} = \frac{\lambda_{ib}\lambda_i e^{-\lambda_{ib}(Td_i-Td_{i-2}-Tr_{i-2})}/h(S_i)}{\left[\lambda_i(\lambda_t+\lambda_p)e^{-\lambda_{ib}(Td_i-Td_{i-1}-Tr_{i-1})-\lambda_i(Td_{i-1}-Td_{i-2})}+\lambda_t(\lambda_p+\lambda_t+\lambda_i)e^{-\lambda_i(Td_i-Td_{i-2})}\right]} \tag{3.6}$$

before-retry classification is to decide on whether or not ω_i is the recurrence of an intermittent fault. I_{flag} will be set to 1 if ω_i is concluded to be the recurrence of a previous intermittent fault.

Further classification depends on the outcome of the current retry. If the current retry is successful, the after-retry classification procedure (Fig. 3) is used, and otherwise, the unsuccessful-retry classification (Fig. 4) is applied following an off-line system reconfiguration which removes or replaces the faulty module. The unsuccessful-retry classification assigns a fault type to every unclassified fault. The procedure of no retry (i.e., $r = 0$) is the same as the procedure after an unsuccessful retry. System reconfiguration is assumed to reset the system to a fault-free state (and thus, I_{flag} is set to 0).

To describe the three classification procedures shown in Figs. 2, 3, and 4, the following variables need to be defined. Let I_{sym} denote the detection symptom just before I_{flag} was last set to 1. Thus, I_{sym} represents the detection symptom of an existing intermittent fault. Let N_{uf} denote the number of unclassified faults in the system. An unclassified fault can be either transient or intermittent, since a permanent fault always results in an unsuccessful retry and every unclassified fault will be assigned a fault type after an unsuccessful retry. Initially, N_{uf} is set to 0.

3.1 Before-Retry Classification

Before-retry classification begins with checking the value of N_{uf}. The value of N_{uf} is made to range from 0 to 2, and hence, a fault will be classified no later than the detection of a second fault after its detection. (Extending N_{uf} to 3 slightly improves the accuracy of fault classification but greatly increases the number of different scenarios to consider, thus greatly increasing the computation overhead. More on this will be discussed in Section 4.) In Fig. 2, ω_{i-1} and ω_{i-2} represent the two faults detected before detecting ω_i. The main goal in the before-retry classification is to determine if ω_i is the recurrence of an earlier unclassified fault. If $N_{\mathrm{uf}} = 0$, nothing needs to be done since all previously detected faults have already been classified. If $N_{\mathrm{uf}} = 1$ and $S_i \neq S_{i-1}$, nothing needs to be done either, since ω_i cannot be the recurrence of ω_{i-1}. If $N_{\mathrm{uf}} = 2$, $S_i \neq S_{i-1}$, and $S_i \neq S_{i-2}$, then $N_{\mathrm{uf}} := N_{\mathrm{uf}} - 1$ and classify ω_{i-2} to be a transient fault based on the assumption that λ_{ib} is much larger than λ_t and λ_i, since ω_i is definitely not a recurrence of ω_{i-1} or ω_{i-2}, and ω_{i-2} must be classified at this time[3].

In other cases, fault classification will depend on the likelihoods of all possible scenarios. If $N_{\mathrm{uf}} = 1$ and $S_i = S_{i-1}$, there are two possible scenarios. The first scenario is that ω_i and ω_{i-1} are two occurrences of an intermittent fault, i.e., ω_{i-1} is of type `if1` and ω_i is of type `if2`. The second scenario is that ω_i and ω_{i-1} are two distinct faults with the same detection symptom, i.e., ω_{i-1} is a transient fault and ω_i could be transient, permanent, or intermittent. One of the two scenarios will be chosen based on the maximum likelihood principle. Let $\mathrm{IT}_{\mathrm{ratio}}$ denote the ratio of the first scenario's likelihood to the second scenario's likelihood. If $\mathrm{IT}_{\mathrm{ratio}} \geq 1$, the first scenario is chosen, thereby resulting in $\mathrm{Type}(\omega_{i-1}) \equiv \mathtt{if1}$, $I_{\mathrm{sym}} = S_i$, and $I_{\mathrm{flag}} = 1$. If $\mathrm{IT}_{\mathrm{ratio}} < 1$, then $\mathrm{Type}(\omega_{i-1}) \equiv \mathtt{tf}$ and $N_{\mathrm{uf}} = 0$. In both scenarios, ω_i will be classified after retry since it could be a permanent fault with the same symptom as w_{i-1}.

Let $P[\mathtt{tf},\mathtt{pf}]$ denote the likelihood that $\mathrm{Type}(\omega_{i-1}) = \mathtt{tf}$ and $\mathrm{Type}(\omega_i) = \mathtt{pf}$. The notations $P[\mathtt{if1},\mathtt{tf}]$, $P[\mathtt{if1},\mathtt{if2}]$, and so on, are defined similarly. Then, $\mathrm{IT}_{\mathrm{ratio}}$ is computed as in Eq. 3.1 where each λ parameter is the mode of its corresponding distribution. This formula is derived from Eqs. 3.2–3.5 where T_P, T_T, and T_I are the detection times of the last classified permanent, transient, and intermittent (type `if1`) faults, respectively. The evaluation of $P[\mathtt{if1},\mathtt{if2}]$ (and other likelihoods) is based on the assumption that occurrences of permanent, transient, and intermittent faults

[3] Otherwise, N_{uf} may become 3, which is not allowed.

are independent.

If $N_{\text{uf}} = 2$ and $S_i = S_{i-2} \neq S_{i-1}$, two possible scenarios are (1) that ω_{i-2} and ω_i are respectively the occurrence and the recurrence of an intermittent fault, and ω_{i-1} is a transient fault occurred during the benign period of the intermittent fault, and (2) that ω_{i-2} is a transient fault independent of ω_i even though they have the same symptom. Let $\text{ITT}_{\text{ratio}}$ denote the likelihood ratio between these two scenarios. Then $\text{ITT}_{\text{ratio}}$ is computed as in Eq. 3.6 where $P[\texttt{tf},\texttt{pf},\texttt{if1}]$ is the likelihood that $\text{Type}(\omega_{i-2}) = \texttt{tf}$, $\text{Type}(\omega_{i-1}) = \texttt{pf}$, and $\text{Type}(\omega_i) = \texttt{if1}$, and others are defined similarly. Correctness of this formula can be easily verified by evaluating all the likelihoods. If $\text{ITT}_{\text{ratio}} \geq 1$, the first scenario is chosen so that $\text{Type}(\omega_{i-2}) \equiv \texttt{if1}$, $\text{Type}(\omega_{i-1}) \equiv \texttt{tf}$, $\text{Type}(\omega_i) \equiv \texttt{if2}$, $I_{\text{sym}} = S_i$, and $I_{\text{flag}} = 1$. Otherwise, $\text{Type}(\omega_{i-2}) \equiv \texttt{tf}$ and ω_{i-1} is left unclassified. In both scenarios, ω_i will not be classified at this time.

3.2 After-Retry Classification

After-retry classification procedure is called for if retry for the current fault ω_i is successful. In this procedure, N_{uf} is incremented by 1 if $I_{\text{flag}} = 0$, i.e., there is no active intermittent fault, since ω_i is either transient or intermittent (type if1). If $I_{\text{flag}} = 1$, the type of ω_i will depend on S_i. If S_i matches the intermittent fault's symptom, I_{sym}, then ω_i is classified as a recurrence of the intermittent fault (type if2); otherwise, it is classified as a transient fault. Upon completion of the current task, As mentioned in the introduction, intermittent faults may be treated differently according to the system's goal, but the unit containing an intermittent fault (indicated by $I_{\text{flag}} = 1$) would eventually have to be removed and/or repaired offline. When this happens, the system will be reconfigured, resetting N_{uf} and I_{flag} to zero.

3.3 Unsuccessful-Retry Classification

The unsuccessful-retry classification assigns a type to every unclassified fault. If $I_{\text{flag}} = 1$ and $S_i = I_{\text{sym}}$, ω_i is classified as the recurrence of an intermittent fault whose active period is longer than the retry period. If $I_{\text{flag}} = 1$ and S_i does not match I_{sym}, ω_i cannot be the recurrence of an intermittent fault and its type depends on the results of diagnosis. In such a case, the diagnosis and repair of a faulty module will be more difficult because the module also contains a latent intermittent fault. If $I_{\text{flag}} = 0$, all unclassified faults (at most one fault could be unclassified at this time) except ω_i are classified to be transient. In the last two cases, the classification of ω_i depends on the outcome of the diagnosis performed on the faulty module. A successful diagnosis (i.e., discovery of the faulty component) will enable us to classify ω_i as a permanent fault. An unsuccessful diagnosis will imply that ω_i be a transient fault, since the reason for not finding any fault is most likely that ω_i is no longer active during the diagnosis. It is possible to make incorrect decisions on whether ω_i is an undiagnosable permanent fault or a transient fault with a very long active period. However, the probability of making incorrect decisions is negligible if the diagnosis has high coverage and the diagnosis time is much longer than the transient fault's active period.

3.4 Sample Collection

The samples for all fault parameters are deduced from detection samples. Specifically, the samples for Y_p, Y_i, and Y_t are obtained from the difference between the detection times of two consecutive faults of type pf, if1, and tf, respectively. The samples for B_i are obtained from the difference between the detection times of consecutive faults of type if1 or if2. The samples for A_t are obtained from ω_i with $\text{Type}(\omega_i) \equiv \texttt{tf}$, whereas the samples for A_i are obtained from ω_i with $\text{Type}(\omega_i) \equiv \texttt{if1}$ or $\text{Type}(\omega_i) \equiv \texttt{if2}$. For example, let the first eight detected faults $\omega_1, \cdots, \omega_8$ be classified as

$$\begin{array}{lclcl}
\text{Type}(\omega_1) & = & \text{Type}(\omega_7) & = & \texttt{pf} \\
\text{Type}(\omega_2) & = & \text{Type}(\omega_6) & = & \texttt{tf} \\
\text{Type}(\omega_3) & = & \text{Type}(\omega_8) & = & \texttt{if1} \\
\text{Type}(\omega_4) & = & \text{Type}(\omega_5) & = & \texttt{if2}
\end{array}$$

and let $I_2 = I_3 = I_4 = 0$ and $I_5 = I_6 = I_8 = 1$. Then, the following samples can be collected:

$$\begin{array}{ll}
Y_p = Td_7 - Td_1; & Y_t = Td_6 - Td_2; \\
Y_i = Td_8 - Td_3; & \\
A_t = Tr_2; & A_i = Tr_3; \\
A_i = Tr_4; & A_i > Tr_5; \\
A_t > Tr_6; & A_i > Tr_8; \\
B_i = Td_4 - Td_3 - Tr_3; & B_i = Td_5 - Td_4 - Tr_4.
\end{array}$$

Unsuccessful retries for transient or intermittent faults only indicate that their active periods are longer than the corresponding retry periods.

4 ANALYSIS OF INCORRECT CLASSIFICATION

As mentioned earlier, the proposed fault classification scheme does not always identify the true fault type. To assess the goodness of the proposed scheme, an estimate of the probability of incorrect classification, denoted by P_e, is derived and compared to simulation results.

Let $\text{Real}(\omega_i)$ denote the true fault type of ω_i. Since a permanent fault is always classified correctly, P_e can be approximated as in Eq. 4.1 where P_e^t is the conditional probability of $\text{Type}(\omega_i) \neq \texttt{tf}$ given that $\text{Real}(\omega_i) = \texttt{tf}$ and P_e^i is the conditional probability of $\text{Type}(\omega_i) \neq \texttt{if1}$ given that $\text{Real}(\omega_i) = \texttt{if1}$. In the expression for P_e, we ignored the classification errors resulting from the recurrence of an intermittent fault since if2-type faults which occur only after a if1-type fault are very rare as compared to transient and permanent faults. Moreover, since intermittent faults are

$$
\begin{aligned}
P_e &= \text{Prob}[\text{Type}(\omega_i) \neq \mathtt{tf} | \text{Real}(\omega_i) = \mathtt{tf}] \cdot \text{Prob}[\text{Real}(\omega_i) = \mathtt{tf}] \\
&\quad + \text{Prob}[\text{Type}(\omega_i) \neq \mathtt{if1} | \text{Real}(\omega_i) = \mathtt{if1}] \cdot \text{Prob}[\text{Real}(\omega_i) = \mathtt{if1}] \\
&= \frac{\lambda_t}{\lambda_t + \lambda_i + \lambda_p} P_e^t + \frac{\lambda_i}{\lambda_t + \lambda_i + \lambda_p} P_e^i, \qquad (4.1)
\end{aligned}
$$

$$
\begin{aligned}
P_e^t &= e^{-\lambda_{ta} T_D} + \left(1 - e^{-\lambda_{ta} T_R}\right) h(S_i) \left(1 - e^{-(\lambda_t + \lambda_p + \lambda_i) L_c}\right) \\
&= e^{-\lambda_{ta} T_D} + \left(1 - e^{-\lambda_{ta} T_R}\right) h(S_i) \left[1 - \left(\frac{h(S_i)\lambda_t(\lambda_t + \lambda_p + \lambda_i)}{\lambda_{ib}\lambda_i e^{\lambda_{ib}/\lambda_{ta}}}\right)^{\left(\frac{\lambda_t + \lambda_p + \lambda_i}{\lambda_{ib} - \lambda_i}\right)}\right] \qquad (4.2)
\end{aligned}
$$

$$
\begin{aligned}
P_e^i &= e^{-\lambda_{ia} T_R} + \left(1 - e^{-\lambda_{ia} T_R}\right) \frac{h(S_i)(\lambda_t + \lambda_p) + \lambda_{ib}}{\lambda_t + \lambda_p + \lambda_{ib}} e^{-(\lambda_t + \lambda_p + \lambda_{ib}) L_c} \\
&= e^{-\lambda_{ia} T_R} + \left(1 - e^{-\lambda_{ia} T_R}\right) \frac{h(S_i)(\lambda_t + \lambda_p) + \lambda_{ib}}{\lambda_t + \lambda_p + \lambda_{ib}} \left[\frac{h(S_i)\lambda_t(\lambda_t + \lambda_p + \lambda_i)}{\lambda_{ib}\lambda_i e^{\lambda_{ib}/\lambda_{ia}}}\right]^{\left(\frac{\lambda_t + \lambda_p + \lambda_{ib}}{\lambda_{ib} - \lambda_i}\right)} \qquad (4.3)
\end{aligned}
$$

identified promptly, most if2–type faults are classified correctly so that P_e will likely be reduced if if2–type faults are considered. It means that P_e should be slightly greater than the actual probability of incorrect classification. This was confirmed by our simulation. For a similar reason, the classification errors resulting from the $\text{ITT}_{\text{ratio}}$ test are also ignored since very rarely the classification would have to depend on the outcome of the $\text{ITT}_{\text{ratio}}$ test.

If retry for the current fault ω_i is successful and the detection syndrome of the next fault ω_{i+1} is identical to that of ω_i, then according to the IT_{ratio} test, ω_i will be classified as a transient fault if $Td_{i+1} - Td_i > L_c$, and as an intermittent fault if $Td_{i+1} - Td_i \leq L_c$, where

$$
L_c = \frac{1}{(\lambda_{ib} - \lambda_i)} \ln \left[\frac{\lambda_{ib}\lambda_i e^{\lambda_{ib} Tr_i}}{h(S_i)(\lambda_p + \lambda_i + \lambda_t)\lambda_t}\right].
$$

Let T_D denote the diagnosis time following an unsuccessful retry and let T_R denote the maximum retry period determined by the adopted retry policy for ω_{i+1}. A transient fault will be classified incorrectly (i) as a permanent fault if its active period is greater than T_D, or (ii) as an intermittent fault if its active period is smaller than T_R and the next fault happens to have the same detection syndrome and occur within the period of L_c. So, P_e^t can be computed as in Eq. 4.2 where Tr_i is approximated by the mean active period of a transient fault, $1/\lambda_{ta}$.

An intermittent fault will be classified incorrectly (i) as a transient or permanent fault if its active period is greater than T_R (thus resulting in an unsuccessful retry), or (ii) as a transient fault if its active period is smaller than T_R and the next fault exhibits the same detection syndrome but occurs outside the period of L_c. Hence, P_e^i is computed as in Eq. 4.3 where Tr_i is approximated by the mean active period of an intermittent fault, $1/\lambda_{ia}$.

To assess the accuracy of P_e, its calculated values are compared to the results obtained from the following simulation. Consider a system with two identical processing modules. Tasks are executed on one module while using the other module as a spare; the former is referred to as the *running module* and the latter the *spare module*. When a fault is detected in the running module, retry is applied for a fixed maximum retry period. If retry fails, the current task will be restarted on the spare module which then becomes the running module. System reconfiguration follows either an unsuccessful retry or the end of a task when $I_{\text{flag}} = 1$. It is assumed that the faulty module can be repaired quickly and returns to the system as the spare module. There are two possible causes for an unsuccessful retry in the simulation. The first cause is that the retry period is shorter than the fault's active period. The second cause is that another fault occurs during retry, which usually happens when an intermittent fault is resident in the running module for a long period.

We assume that an unlimited number of tasks with the same execution time are waiting to be executed so that the running module may always be kept busy. The simulated classification accuracy is defined as the percentage of those detected faults which have been classified correctly, whereas the estimated accuracy is calculated as $1 - P_e$.

The simulation is run for a period of 1,000,000 under a basic fault generation process with the following fault parameters:

$1/\lambda_p$	$1/\lambda_t$	$1/\lambda_i$	$1/\lambda_{ta}$	$1/\lambda_{ia}$	$1/\lambda_{ib}$
2500	500	3500	2	3	4

The inverses of rate parameters are listed above because they are easier to compare with other time-related parameters. For example, $1/\lambda_p$ is the mean time between two permanent faults, $1/\lambda_{ta}$ is the mean active period of a transient fault, etc. So during one simulation run, we can collect samples for about 400 permanent faults, 2000 transient faults, and 300 intermittent faults. The execution time for each task is 200. The maximum diagnostic time T_D is 100, i.e., any fault with an active period greater than 100 will be classified as a permanent fault by our fault classification scheme. The maximum retry period T_R is 3. All these variables have the same time unit and thus are listed without specifying units. It is assumed that there are 50 different detection symptoms and their occurrences are uniformly distributed with $h(S_i) = 0.02$.

To justify the inequality $N_{\rm uf} \leq 2$, we also simulated the proposed classification scheme with $N_{\rm uf} \leq 1$ and $N_{\rm uf} \leq 3$. It is found that under almost all circumstances, the classification error percentages for $N_{\rm uf} \leq 2$ and $N_{\rm uf} \leq 3$ are the same, indicating that extending $N_{\rm uf}$ to 3 will not improve the accuracy of the proposed classification scheme. However, a small improvement was found when we switch from $N_{\rm uf} \leq 1$ to $N_{\rm uf} \leq 2$.

The first comparison is made with $1/\lambda_t$ varying from 300 to 1500, and the results are plotted in Fig. 5. The dashed line represents the estimated classification accuracy, 1 - P_e, which was calculated using Eq. (4.1). The solid line represents the simulated classification accuracy, defined as the percentage of faults being classified correctly during the simulation. It is evident that the estimated accuracy is always lower than the simulated one, but their difference is shown to be quite small. So, the estimated accuracy can be viewed as a close lower bound of the actual accuracy. The trend of these curves is also interesting: the greater the transient fault's occurrence rate, the higher the classification accuracy. This was expected because transient faults are less likely to be misclassified, as compared to intermittent faults.

A second comparison is made with the number of detection symptoms varying from 1 to 100, and the results are plotted in Fig. 6. Generally, the accuracy improves with the number of detection symptoms. However, the improvement becomes very insignificant when the number of detection symptoms gets larger than 20. When no detection symptom is available to use, the estimated accuracy is uncharacteristically higher than the simulated accuracy. In such a case, the assumption in deriving P_e that the classification error resulting from recurrences of intermittent faults is negligible is no longer valid. However, even if there is only one detection syndrome, the estimated accuracy becomes very close to the simulated accuracy, which is itself quite high (approximately 93%).

5 CONCLUSION

In this paper, we have proposed a new fault classification scheme based on a Bayesian approach so that fault parameters can be estimated on-line using the most up-to-date information via detection and retry mechanisms. This scheme will eventually classify every detected fault as one of the four temporal fault types: permanent fault, transient fault, first occurrence and recurrence of an intermittent fault. The simulation results show that if the number of different detection symptoms is greater than 5, the accuracy of our fault classification is very high (around 96%). Since it is impossible to devise a perfect classification scheme, an expression for the probability of incorrect fault classification by our scheme is also derived, providing a close estimate of the simulation or true results.

A direct application of the proposed fault classification scheme is the on-line determination of the "optimal" retry period. Another application is the prompt identification of intermittent faults. Early knowledge of the existence of an intermittent fault in the system can facilitate the selection of a proper countermeasure. For example, if we want to limit the damage of an intermittent fault to a small well-defined area, but still want fast recovery for the current task, the following strategy can be used. Whenever an intermittent fault is first identified, an optimal retry period is determined and then applied to all future recurrences of the intermittent fault until the current task is completed, and then, the unit containing the intermittent fault will be switched out — even if the fault is not active — to prevent any further damages to the system. Other strategies against intermittent faults can also be designed based on the proposed fault classification scheme. All of these are a matter of our future research.

References

[1] K. G. Shin and Y.-H. Lee, "Error detection process – model, design, and its impact on computer performance," *IEEE Trans. Computers*, vol. C-33, pp. 529–540, June 1984.

[2] J. C. Laprie, "Dependable computing and fault tolerance: Concepts and terminology," *Digest of papers, FTCS-15*, pp. 2–11, June 1985.

[3] T. Anderson and P. A. Lee, *Fault Tolerance Principles and Practice.* London: Prentice-Hall International, 1981.

[4] D. P. Siewiorek and R. S. Swarz, *The Theory and Practice of Reliable System Design.* Bedford, MA: Digital Equipment Corporation, 1982.

[5] D. P. Siewiorek, V. Kini, H. Mashburn, S. R. McConnel, and M. M. Tsao, "A case study of c.mmp, cm*, and c.vmp: Part i – experiences with fault tolerance in multiprocessor systems," *Proceedings of the IEEE*, vol. 66, pp. 1178–1199, Oct. 1978.

[6] O. Tasar and V. Tasar, "A study of intermittent fault in digital computers," *Proc. Nat. Comput. Conf.*, pp. 807–811, June 1977.

[7] Y.-H. Lee and K. G. Shin, "Optimal design and use of retry in fault-tolerant computer systems," *J. ACM*, vol. 35, pp. 45–69, Jan. 1988.

[8] M. Berg and I. Koren, "On switching policies for modular redundancy fault-tolerant computing systems," *IEEE Trans. Computers*, vol. C-36, pp. 1052–1062, Sept. 1987.

[9] I. Koren, Z. Koren, and S. Y. H. Su, "Analysis of a class of recovery procedures," *IEEE Trans. Computers*, vol. C-35, pp. 703–712, Aug. 1986.

[10] J. O. Berger, *Statistical Decision Theory, Foundations, Concepts, Methods.* New York: Springer-Verlag, 2nd ed., 1985.

[11] M. H. DeGroot, *Optimal Statistical Decisions.* New York: McGraw-Hill Book Company, 1970.

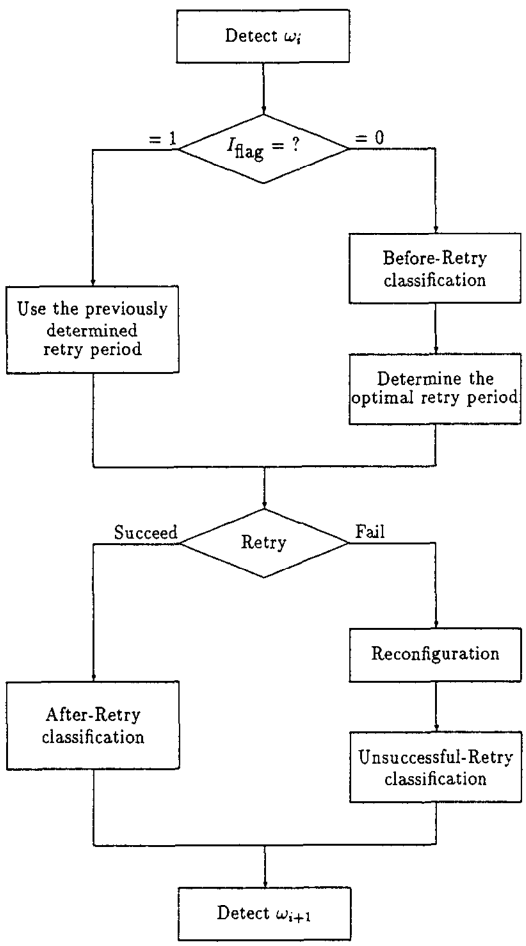

Figure 1: The procedure of a retry recovery.

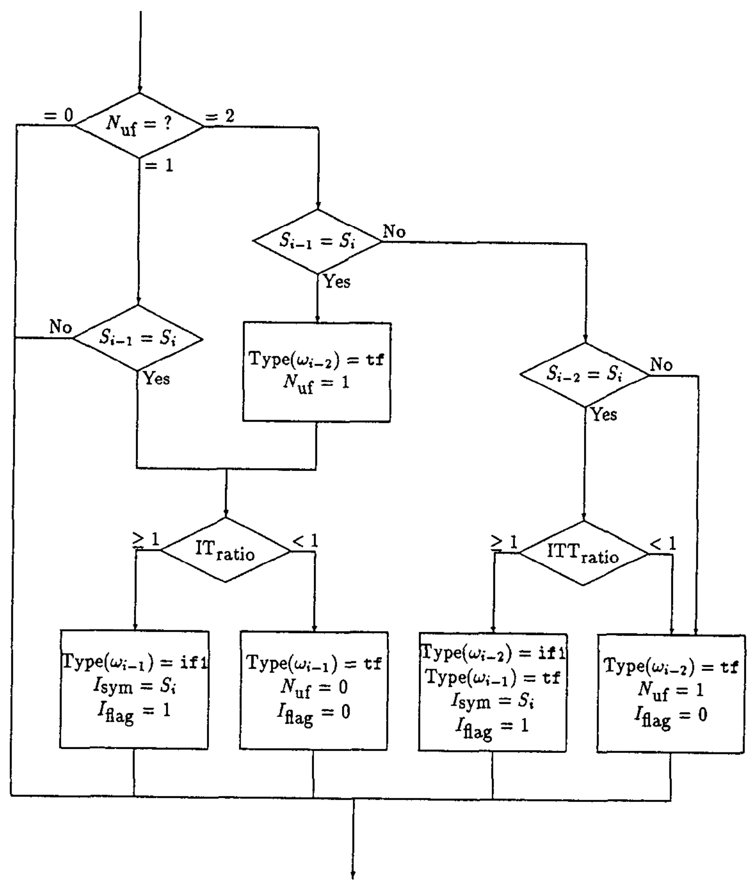

Figure 2: The procedure of before-retry classification.

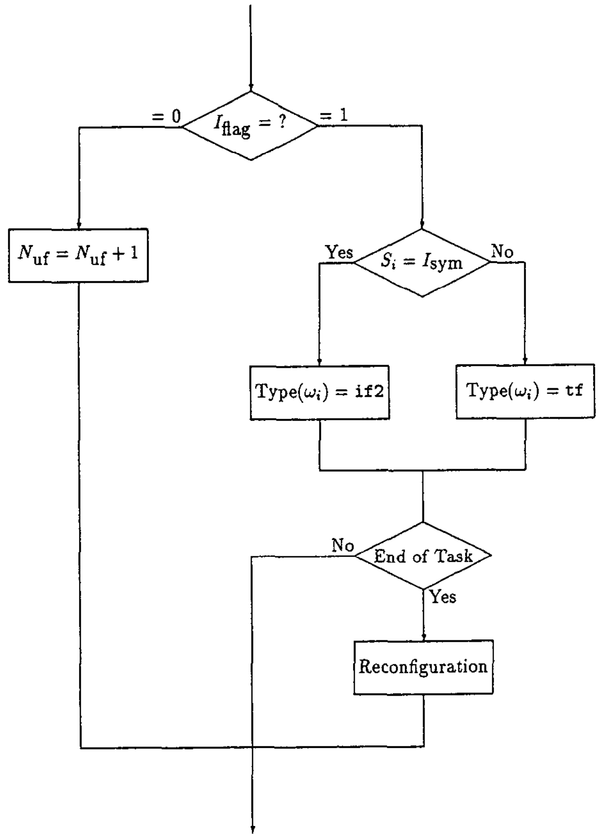

Figure 3: The procedure of after-retry classification.

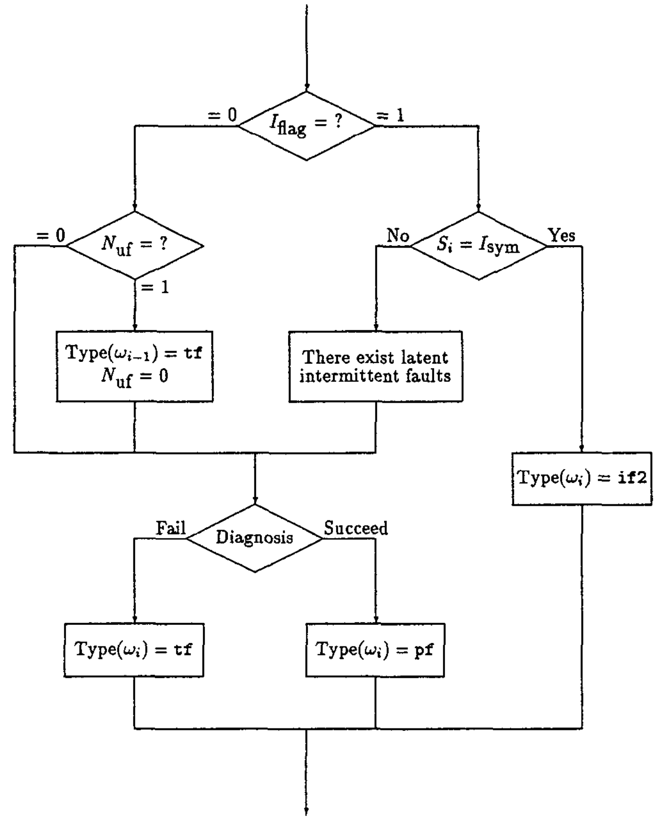

Figure 4: The procedure of unsuccessful-retry classification.

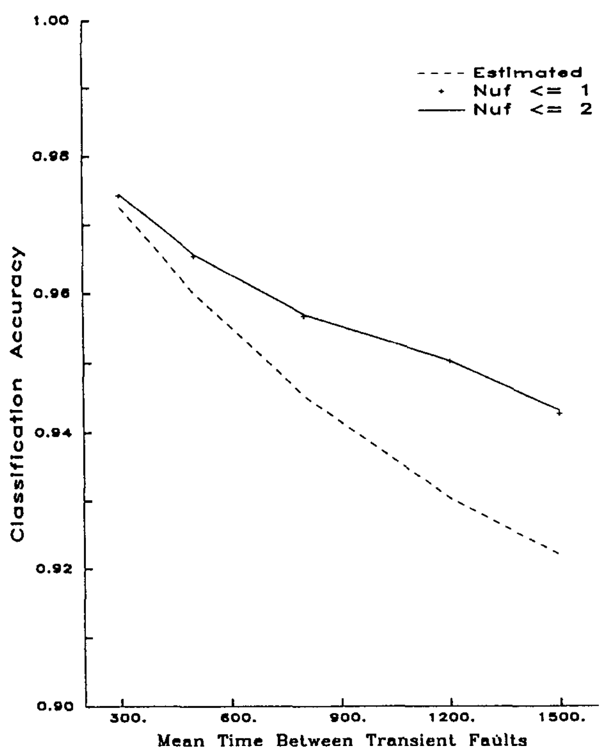

Figure 5: Classification accuracy vs. $1/\lambda_t$ when $1/\lambda_p = 2500$, $1/\lambda_i = 3500$, $1/\lambda_{ta} = 2$, $1/\lambda_{ia} = 3$, and $1/\lambda_{ib} = 4$.

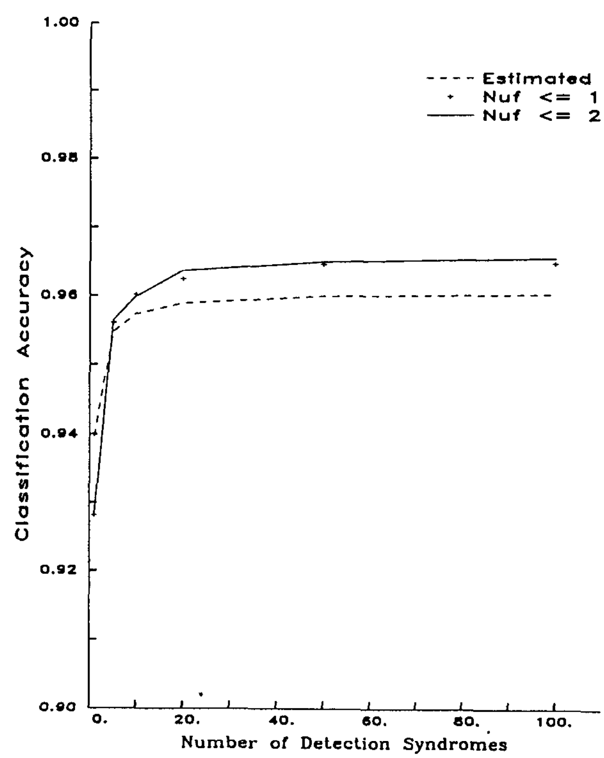

Figure 6: Classification accuracy vs. number of detection symptoms when $1/\lambda_t = 500$, $1/\lambda_p = 2500$, $1/\lambda_i = 3500$, $1/\lambda_{ta} = 2$, $1/\lambda_{ia} = 3$, and $1/\lambda_{ib} = 4$.