https://ntrs.nasa.gov/search.jsp?R=19870018001 2020-03-20T09:34:49+00:00Z

SEMI-ANNUAL PROGRESS REPORT

NASA RESEARCH GRANT NSG 1442

Performance Analysis of a Generalized Upset Detection Procedure

by

Douglas M. Blough and Gerald M. Masson

Performance Analysis of a Generalized Upset Detection Procedure*

Douglas M. Blough and Gerald M. Masson

Department of Computer Science
The Johns Hopkins University
Baltimore, MD 21218

Abstract - In this paper, a general procedure for upset detection in complex systems, called the *data block capture and analysis upset monitoring process*, is described and analyzed. The process consists of repeatedly recording a fixed amount of data from a set of predetermined observation lines of the system being monitored (i.e. capturing a block of data), and then analyzing the captured block in an attempt to determine whether the system is functioning correctly. The algorithm which analyzes the data blocks can be characterized in terms of the amount of time it requires to examine a given length data block to ascertain the existence of features/conditions that have been predetermined to characterize the upset-free behavior of the system. The performance of linear, quadratic, and logarithmic data analysis algorithms is rigorously characterized in terms of three performance measures: (i) the probability of correctly detecting an upset, (ii) the expected number of false alarms, and (iii) the expected latency in detecting upsets.

1. Introduction

In this paper we consider an approach to upset detection in complex systems. The approach is based on a data block capture and analysis monitoring process which can be modeled in very general terms, thereby rendering our results broadly applicable. The process consists of recording signals on a set of observation lines of the system to form (capture) a data block and then analyzing this block of data. On the basis of this analysis, a determination is to be made as to whether the system is functioning correctly or has been upset by some fault condition.

One possible implementation of the data block capture and analysis process could be a concurrent monitoring device. Figure 1 is a schematic which illustrates this perspective of concurrent monitoring.

A key aspect of the data block capture and analysis monitoring process is that its implementation, whatever form it may take, should be relatively simple when compared with the complex system being monitored. In other words, the process is not meant to duplicate the performance of the system. Accordingly, the analysis that is performed on a captured block of data will not be an explicit comparison of results with the system being monitored. Rather, the analysis should be thought of as an examination of the captured data block in which an algorithm is executed in an attempt to ascertain the existence of features or conditions that have been predetermined to characterize the upset-free behavior of the system.

In order to characterize the performance of the data block capture and analysis monitoring process, we must consider the four possible outcomes which could result from execution of the analysis algorithm on a captured block of data:

Assuming the system was upset during the capture of a data block, there are two possible outcomes:

* This work was supported by NASA under Grant NSG-1442.

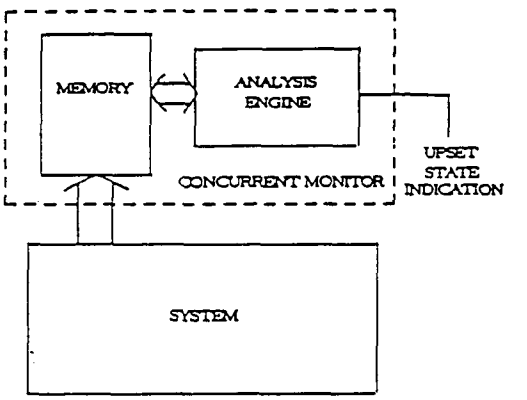

Figure 1 - A Concurrent Monitor Employing the Data Block Capture and Analysis Process

the data analysis algorithm could perform *correct detection* by indicating that the system was in the upset state

or

the data analysis algorithm could perform *incorrect rejection* by indicating that the system was in the no-upset state.

Assuming the system was not upset during the capture of the data block, there are two possible outcomes:

the data analysis algorithm could generate a *false alarm* by indicating that the system was in the upset state

or

the data analysis algorithm could perform *correct rejection* by indicating that the system was in the no-upset state.

We will want to theoretically analyze the upset detection capabilities of the data block capture and analysis process in terms of the *single monitoring period error probabilities*, the probabilities of false alarm and incorrect rejection resulting from a captured block of data. These error probabilities characterize the capability of the data analysis algorithm on a single block of data. To analyze the overall performance of the process, we will introduce measures of upset detection capability called the *probability of correct detection* [1], (or more succinctly, the *detection probability*), the *expected number of false alarms per upset*, and the *expected detection latency*.

Heretofore, evaluations of an upset monitor's performance could only be achieved by implementing or simulating the monitor and then experimentally determining its upset detection capabilities [2],[3]. Our measures will permit an analytical evaluation of the performance of the data block capture and analysis process based on key parameters of the process.

2. Preliminaries

An upset is a disruption of correct system behavior. Upsets are caused by faults or underlying failure mechanisms in the system. The faults can be permanent or intermittent/transient in nature. In general, these faults occur at such a low level in the sys-

0731-3071/87/0000/0218$01.00 © 1987 IEEE

tem that they are, unto themselves, of no use relative to concurrent monitoring. It is the effect of the faults on observable signal lines of the system with which we must be concerned. In order for correct detection of an upset to take place, a data block would have to be captured at a time when the signal activity on the observation lines was showing evidence of the upset. However, we will assume that the entire captured data block need not indicate an upset for correct detection to take place. Rather, in general, if any portion of the captured data block shows evidence of an upset, we will assume that it is possible for the process implementation to perform correct detection. Furthermore, the implementations are not expected to be infallible. Even if the entire captured data block showed evidence of an upset occurrence, it would still be possible for an incorrect rejection to take place. Likewise, if the captured data block showed no evidence of an upset, there would still be the possibility of a false alarm occurrence.

Given our perspective of upsets and their detection, we can model the occurrence of upsets in the system as a two-state continuous-time Markov process, moving between the states *upset* and *no-upset*. This is illustrated in Figure 2. In this model, both the time of occurrence and the duration of the upsets are random. The infinitesimal matrix, A, characterizes a continuous-time Markov process [4]. For our model,

$$A = \begin{bmatrix} -\lambda & \lambda \\ \mu & -\mu \end{bmatrix}$$

where the no-upset state is state 0 and the upset state is state 1. It is implied by this model that the time the upset condition does not exist is exponentially distributed with parameter λ, and the time the upset condition is active is exponentially distributed with parameter μ. A continuous-time Markov model such as this has been used to model both individual fault conditions within digital systems [5], as well as system-level upsets [6],[7]. An upset will be said to *occur* at the time that the system enters the upset state.

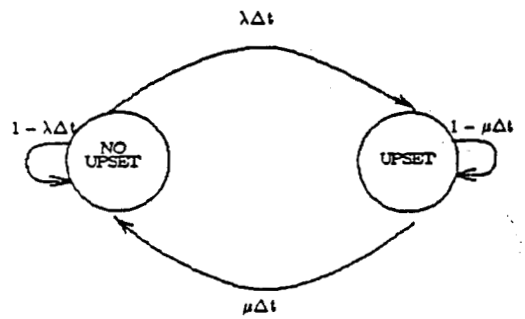

Figure 2 - A Continuous Markov Model

3. The Data Block Capture and Analysis Process

The data block capture and analysis process consists of repeated applications of a two-phase function:

In the *data capture phase* of operation, a sequence (block) of data signals (cycle-by-cycle relative to the system clock) on the selected observation lines is stored (captured). We will denote the length of the captured data block by the parameter n.

In the *data analysis phase* of operation, an algorithm, which examines the most recently captured data block for evidence of an upset, is executed.

We will refer to the total amount of time spent by the process in the data capture phase and the data analysis phase as the *monitoring period*.

Implementation of the data analysis phase of the process requires execution of a data analysis algorithm which examines the captured data block for variations of predetermined features/conditions of signal activity. These features/conditions could be based on properties of the functions being computed [8]; or they could result from a use of coding theory [9], or an embedding of signatures in the signal/data flow [10]; or they simply could be experimentally extracted using pattern recognition techniques [11],[12]. These techniques encompass both *specification based* [8], as well as *symptom based* [13], diagnosis.

3.1. The Data Analysis Algorithm

Certainly, the upset detection capability of the data block capture and analysis process depends on the amount of time allotted to the analysis of captured data. More time spent examining the captured data intuitively corresponds to a more thorough analysis of the block and a higher probability of correctly detecting (the more subtle) evidence of an upset. We can generally characterize this analysis time in terms of the complexity of the data analysis algorithm employed by the process implementation. Since n is the length of the data block captured, we will consider data analysis algorithms that require a number of execution steps that are linear, quadratic, and logarithmic functions of n. For each such data analysis algorithm, the time required for execution will be assumed to be proportional to the complexity, and we will denote this analysis time by the function T(n). The following table lists the data analysis algorithms which we will consider in the remainder of the paper.

Data Analysis Algorithm	Execution Time	Constraints
Linear	$T_L(n) = an$	$a > 0$
Quadratic	$T_Q(n) = an^2+bn+c$	$a > 0$ $b \geq 0$ $c \geq 0$
Logarithmic	$T_{LOG}(n) = an \cdot \log n$	$a > 0$

Table 1

3.2. Error Probabilities

We are led by the above discussion to consider a means of assessing the single monitoring period error probabilities of the data block capture and analysis process. These error probabilities describe the extent to which the data analysis algorithm produces the incorrect outcomes (incorrect rejection or false alarm) during a single data analysis phase, and are defined to be:

$p = Pr\{\text{algorithm indicates no-upset} \mid \text{system in upset state}\}$

and

$q = Pr\{\text{algorithm indicates upset} \mid \text{system in no-upset state}\}$

We will consider four classes of p functions, each of which in some way quantifies the notion that as n increases, the sensitivity of the process implementation to upsets will also increase, and p will correspondingly decrease. The characteristics of these classes are summarized in Table 2.

We will assume that increasing the data which is available to the analysis algorithm enhances the performance of the algorithm relative to false alarms as well as incorrect detections. Thus, we will allow q functions to belong to the same four classes proposed for p functions.

Class	Function	Comments
Inverse Logarithmic	$\frac{k}{\ln n}$	poor sensitivity to features/conditions, large p offset at high n, low detection probability
Decaying Exponential	e^{-kn}	moderate sensitivity in short blocks, high sensitivity in longer blocks, sometimes requires several indications to correctly detect an upset
Inverse Linear	$\frac{k}{n}$	moderately high sensitivity for all block lengths, does not benefit from extra indications in longer blocks, hence some upsets in longer blocks not detected
Inverse Quadratic	$\frac{k}{n^2}$	excellent sensitivity in all blocks, most rapid decrease, almost complete detection for large n

Table 2

4. The Probability of Correct Detection

The primary performance goal of an upset detection procedure is to detect as many of the upsets which occur in the system as possible. The *probability of correct detection* [1], or simply the *detection probability*, is defined to be

$$\Pr\{\text{an upset detected} \mid \text{an upset occurs}\}$$

The detection probability is a natural measure of upset detection capability. Clearly, this quantity depends on the time of occurrence of the upset. To remove such short term inconsistencies, it is natural to consider upset occurrences over a long period of time. If we define the detection probability over an interval, $D([t_1, t_2))$, to be

$$D([t_1,t_2)) = \Pr\{\text{upset detected} \mid \text{upset occurs in } [t_1,t_2)\}$$

then this can be accomplished by considering intervals with $t_2 \gg t_1$.

Specifically, for the data block capture and analysis process, we will consider the interval detection probability over a large number of monitoring periods. Thus, we will define the detection probability, D, in the following way:

$$D = \lim_{l\to\infty} D([0, i(n+T)))$$

where $[0, n+T)$ is a single monitoring period of the process. If we let X represent the time of occurrence of the upset we can express the detection probability as

$$D = \lim_{l\to\infty} \int_0^{l(n+T)} \Pr\{\text{upset detected} \mid X = x\} \cdot f(x \mid \text{upset occurs in } [0, i(n+T))) \, dx$$

Calculation of the detection probability of the data block capture and analysis process by the above formula requires knowledge of the conditional distribution of time of upset occurrence. This distribution can be shown to be approximately uniform, assuming the mean time between upsets $(1/\lambda)$ is much larger than the mean upset duration $(1/\mu)$. Using this result along with the fact that the probability of an upset being detected depends only on its position within the data block capture and analysis monitoring period, we get the following:

$$\begin{aligned} D &\cong \lim_{l\to\infty} \frac{1}{i(n+T)} \int_0^{l(n+T)} \Pr\{\text{upset detected} \mid X=x\}dx \\ &= \lim_{l\to\infty} \frac{i}{i(n+T)} \int_0^{n+T} \Pr\{\text{upset detected} \mid X=x\}dx \\ &= \frac{1}{n+T} \int_0^{n+T} \Pr\{\text{upset detected} \mid X = x\}dx \end{aligned} \tag{4.1}$$

$$= D([0, n+T))$$

Thus, the detection probability of the data block capture and analysis process is simply equal to the interval detection probability over a single monitoring period. Now,

$$\begin{aligned} &\Pr\{\text{upset not detected} \mid X = x\} \\ &= \sum_{i=1}^{\infty} \Pr\{\text{upset missed in } 1^{st} \text{ i analysis phases} \mid \\ &\qquad \text{upset ends during } i^{th} \text{ monitoring period}\} \cdot \\ &\qquad \Pr\{\text{upset ends during } i^{th} \text{ monitoring period}\} \\ &= p(1 - e^{-\mu(n+T-x)}) + \sum_{i=2}^{\infty} p^i \Pr\{t<(i+1)(n+T) - x \mid \\ &\qquad t\geq i(n+T) - x\} \Pr\{t\geq i(n+T) - x\} \end{aligned}$$

if $0 \leq x < n$. By the memoryless property of the exponential distribution,

$$\begin{aligned} &\Pr\{t < (i+1)(n+T) - x \mid t \geq i(n+T) - x\} \\ &\quad = \Pr\{t < n+T\} = 1 - e^{-\mu(n+T)} \end{aligned}$$

Thus,

$$\begin{aligned} &\Pr\{\text{upset not detected} \mid X = x\} \\ &= p\left[1 - e^{-\mu(n+T-x)} + (1 - e^{-\mu(n+T)}) \sum_{i=1}^{\infty} p^i e^{-\mu[i(n+T)-x]}\right] \\ &= p\left[1 - \frac{(1-p)e^{-\mu(n+T-x)}}{1 - pe^{-\mu(n+T)}}\right] \end{aligned}$$

Thus,

$$\begin{aligned} &\Pr\{\text{upset detected} \mid X = x\} \\ &= 1 - p\left[1 - \frac{(1-p)e^{-\mu(n+T-x)}}{1 - pe^{-\mu(n+T)}}\right], \quad 0 \leq x < n \end{aligned} \tag{4.2}$$

Similarly, it can be shown that

$$\Pr\{\text{upset detected} \mid X = x\} = \frac{(1-p)e^{-\mu(n+T-x)}}{1 - pe^{-\mu(n+T)}}, \quad n \leq x < n+T \tag{4.3}$$

where the data analysis phase of the process takes place during $[n, n+T)$. It remains only to evaluate Equation 4.1. The final

ORIGINAL PAGE IS
OF POOR QUALITY

result is that the detection probability is given by

$$D = \frac{(1-p)\left[(\mu n+1)(1-pe^{-\mu(n+T)}) - (1-p)e^{-\mu T}\right]}{\mu(n+T)(1-pe^{-\mu(n+T)})}$$

5. The Occurrence of False Alarms

While the probability of correct detection is an important measure of the performance of an upset detection procedure, it should by no means be the only one. It would be trivial to implement a procedure which always claims that the system is in an upset state. Clearly, such a procedure has $D = 1$ at the expense of many false alarm occurrences. Given that false alarms will occur [14], it may be worthwhile to trade off correct detection capability and false alarm production.

The above discussion leads us to consider a measure for the extent to which false alarms are produced. Let us define a random variable, F, as

F = number of false alarms between upset occurrences

We will use E[F], the expected number of false alarms per upset, as our false alarm occurrence measure.

If a false alarm is regarded as a "success", each decision produced by the process implementation while the system is in the no-upset state is a Bernoulli trial with parameter q. Thus,

$$F = \sum_{i=1}^{N} B_i$$

where B_i represents the outcome of the i^{th} decision and N is the number of decisions made between upset occurrences. If we let

Y = sojourn time in no-upset state

where the sojourn time in a state is the length of time spent during a single visit to that state, then

$$N = \left\lfloor \frac{Y}{n+T} \right\rfloor$$

Using results from probability concerning random sums [4] we get that

$$E[F] = qE[N]$$

Now, it can be shown that

$$E[N] = \frac{e^{-\lambda(n+T)}}{1-e^{-\lambda(n+T)}}$$

and the expected number of false alarms per upset is therefore given by

$$E[F] = \frac{qe^{-\lambda(n+T)}}{1-e^{-\lambda(n+T)}}$$

To determine the optimum value of E[F] its behavior as the block length is allowed to vary must be determined. The following lemma suggests the shape of E[F] for a wide class of q functions.

Lemma 1: If q is a non-increasing function of n, then E[F] is a strictly decreasing function of n.

Proof: $e^{-\lambda(n+T)}$ is a strictly decreasing function and $1-e^{-\lambda(n+T)}$ is a strictly increasing function. Thus, if q does not increase with n, it must be that $\frac{qe^{-\lambda(n+T)}}{1-e^{-\lambda(n+T)}}$ is a strictly decreasing function.

□

6. Detection Latency

Another important characteristic of an upset detection process is the amount of time between an upset's occurrence and its subsequent detection. We will refer to this time as the *detection latency* of the process.

Assuming an upset is correctly detected, we will define the *detection latency*, L, to be the time between the system entering the upset state and the time at which the upset is correctly detected. For the data block capture and analysis process, we will assume that the data analysis algorithm indicates whether or not the system was found to be in the upset state only at the end of a monitoring period. This forces the detection latency to be at least T, the length of the data analysis phase. As with the detection probability, we will consider upsets occurring in a single monitoring period, due to the periodicity of the data block capture and analysis process. Hence, we will define the *expected detection latency* of the process in the following way:

$$E[L] = E[L \mid \text{upset occurs in } [0,n+T) \text{ and is detected}]$$

where, as before, [0, n+T) represents a single monitoring period of the data block capture and analysis process. If we let X represent the time of upset occurrence, the expected latency can be rewritten as:

$$E[L] = \int_0^{n+T} E[L \mid X=x]\, f(x \mid \text{upset occurs in } [0,n+T) \text{ and is detected})\, dx \qquad (6.1)$$

Given that $X = x$, L is a discrete random variable with sample space Ω, given by

$$\Omega = \{n+T-x,\ 2(n+T)-x,\ 3(n+T)-x, \ldots\}$$

Furthermore, the conditional distribution of L is given by

$$\Pr\{L = i(n+T)-x \mid X = x\} = \begin{cases} (1-p)p^{i-2}, & \text{if } n \le x < n+T \\ (1-p)p^{i-1}, & \text{if } 0 \le x < n \end{cases}$$

From this the conditional expected value of L can easily be shown to be

$$E[L \mid X = x] = \begin{cases} \frac{n+T}{1-p} - x, & \text{if } 0 \le x < n \\ \frac{n+T}{1-p} - x + n + T, & \text{if } n \le x < n+T \end{cases} \qquad (6.2)$$

Now, f(x | upset occurs in [0,n+T) and is detected) can be calculated from quantities derived in Section 4. Evaluation of Equation 6.1 yields the following expression for the expected detection latency:

$$E[L] = \frac{\left[\frac{(\mu n+1)(n+T)}{1-p} - \frac{(\mu n)^2-2}{2\mu}\right](1-pe^{-\mu(n+T)}) - \left[2(n+T)+(1-p)\left(\frac{1}{\mu}-n\right)\right]e^{-\mu T}}{\left[(\mu n+1)(1-pe^{-\mu(n+T)}) - (1-p)e^{-\mu T}\right]} \qquad (6.3)$$

7. Evaluation of Performance Measures

In this section we will consider in detail the performance of the data block capture and analysis process. The behavior of our performance measures will be examined as the length of the cap-

tured data block is allowed to vary.

7.1. Linear Data Analysis Algorithms

If we let

$$p_1 = 1 - p\left[1 - \frac{(1-p)(e^{-\mu T} - e^{-\mu(n+T)})}{\mu n(1 - pe^{-\mu(n+T)})}\right]$$

and

$$p_2 = \frac{(1-p)(1-e^{-\mu T})}{\mu T(1 - pe^{-\mu(n+T)})}$$

then the detection probability is given by

$$D = \frac{np_1 + Tp_2}{n+T} \tag{7.1}$$

Since $T_{LIN} = an$, the detection probability for a linear data analysis algorithm is

$$D = \frac{p_1 + ap_2}{a+1} \tag{7.2}$$

The asymptotic behavior of a linear data analysis algorithm can be derived using Equation 7.2 and is stated in the following lemma.

Lemma 2: For a linear data analysis algorithm, if $p \to 0$ as $n \to \infty$, then $\lim_{n\to\infty} D = \frac{1}{a+1}$.

Proof: If $p \to 0$ as $n \to \infty$, then $p_1 \to 1$ and $p_2 \to 0$. We can then see from Equation 7.2 that $D \to \frac{1}{a+1}$.

□

Typically, $a > 1$. In this situation, $\lim_{n\to\infty} D < 1/2$. Hence, capturing arbitrarily long blocks does not yield a high detection probability. The p functions being considered are all equal to, or arbitrarily close to, 1 for some small value of n. With $p = 1$, $D = 0$, so arbitrarily small block lengths yield a low detection probability, as well. We would, therefore, like to examine the block length spectrum between $n = 0$ and $n = \infty$ to determine whether a global maximum exists or whether the detection probability is bounded by its limiting value, $\frac{1}{a+1}$. In order to accomplish this, specific choices of p and q functions must be considered.

A relatively conservative linear data analysis algorithm might have single monitoring period error probability functions given by $p = \frac{k}{n}$ and $q = \frac{k'}{n}$. Figure 3 shows plots of the detection probability, expected number of false alarms per upset, and the expected detection latency of such an algorithm for $a = 2$, $k = 10$, and $k' = 2 \times 10^{-6}$. A value of $\lambda = 10^{-10}$ is used throughout this section in calculating the expected number of false alarms per upset. The maximum detection probability, with $\mu = 0.005$, occurs at $n = 32$ and has a value of 0.775. Thus, almost 80% of all upsets are detected by such a scheme. Furthermore, an interesting tradeoff can be achieved by noting that the expected number of false alarms per upset decreases very rapidly with n while the detection probability decreases slowly and the expected detection latency increases slowly. Because of this, small sacrifices in detection probability and expected detection latency yield a large payoff in terms of the expected number of false alarms per upset. For example, at $n = 32$, $E[L] = 152.65$ and $E[F] = 6.510$. When $n = 92$, $D = 0.707$, $E[L] = 333.70$, and $E[F] = 0.788$. Thus, the number of false alarms which occur is decreased by a factor greater than 8 while the expected detection latency increases by a factor of about 2 and the detection probability decreases by less than 10%.

It is important to determine if a given algorithm in a particular fault environment can achieve a detection probability greater than the limiting value given by Lemma 2. The following theorem, stated without proof, gives necessary and sufficient conditions on the existence of a maximum as shown in Figure 3, for arbitrary values of a and k.

Theorem 1: For a linear data analysis algorithm, with $p = \frac{k}{n}$, $\max_{all\ n} D > \frac{1}{a+1}$ if and only if $\mu < \frac{1}{k}$.

A less conservative linear data analysis algorithm should be able to detect more upsets by allowing more false alarms to occur. Such an algorithm might have $p = \frac{k}{n^2}$ and $q = \frac{k'}{\ln n}$. Examination of such an algorithm for $a = 2$, $k = 100$, and $k' = 5 \times 10^{-7}$ shows that the maximum detection probability has increased to a value of 0.874 at $n = 26$, while the expected number of false alarms per upset has increased to 19.675 at the same block length. Thus, by using a more aggressive algorithm (which as a side-effect allows an increased number of false alarms to occur) it has been possible to detect almost 90% of all upsets in the system. Such a tradeoff would be beneficial in many cases.

For any linear data analysis algorithm with a p function approaching 0 faster than $1/n$ the global maximum exists independently of the choice of μ, as is shown by the following theorem, also stated without proof.

Theorem 2: For a linear data analysis algorithm, with $p = \frac{k}{f(n)}$, where $\lim_{n\to\infty} \frac{n}{f(n)} = 0$, $\max_{all\ n} D > \frac{1}{a+1}$, for all μ.

7.2. Higher Order Data Analysis Algorithms

It may be possible to decrease the probabilities of incorrect rejection and false alarm in a single monitoring period by spending more time analyzing each data block. If the time required to analyze a block grows too quickly, however, large portions of system activity become invisible to the process, leading to poor overall performance. For any data analysis algorithm which has running time of order greater than n, the following limit result holds.

Lemma 3: For any data analysis algorithm with $n = o(T)$, $\lim_{n\to\infty} D = 0$.

Proof: From Equation 7.1, $D = \frac{np_1 + Tp_2}{n+T}$. Note that $p_2 \to 0$ as $n \to \infty$, regardless of p. Hence, $D \to \frac{np_1}{n+T}$. Since $p_1 \leq 1$ and $n = o(T)$, we have that $D \to 0$.

□

With higher order data analysis algorithms, since D is arbitrarily close to 0 for small enough n and $\lim_{n\to\infty} D = 0$, clearly a global maximum between 0 and ∞ exists, independent of p and μ. We will mainly be interested, then, in the value and position of this maximum, along with the behavior of the expected detection latency and the expected number of false alarms per upset.

Use of a quadratic data analysis algorithm should produce small probabilities of incorrect rejection and false alarm over a single monitoring period. Examination of such an algorithm with $p = \frac{k}{n^2}$, and $q = \frac{k'}{n^2}$, with parameter values $a = 0.2$, $b = c = 0$, $k = 100$, and $k' = 5 \times 10^{-5}$, seems to show that quadratic data analysis algorithms are feasible only if used on very short data blocks. At $n = 17$, the maximum detection probability is achieved with a value of 0.773. $E[L] = 132.70$ and $E[F] = 23.130$ at the same block length. For $n > 17$, there is a rapid decrease in the detection probability and a rapid increase in

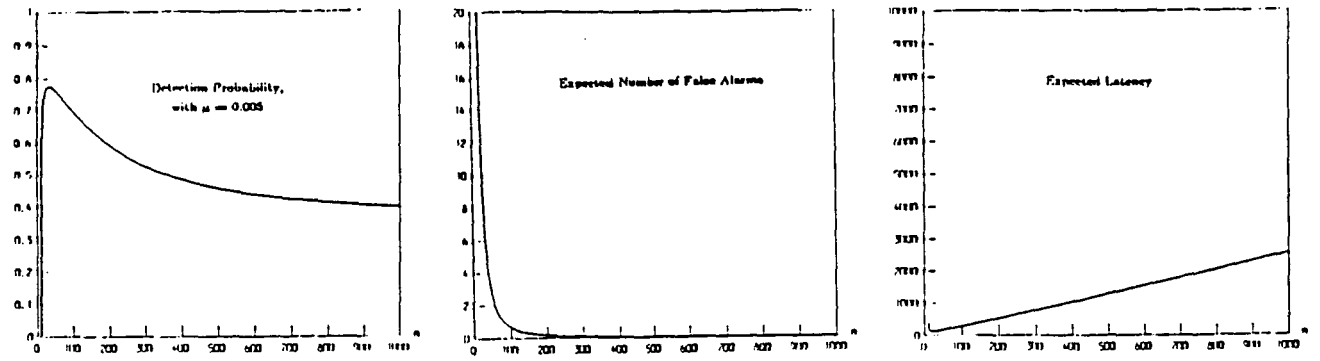

Figure 3

the expected detection latency. For example, when $n = 100$, $D = 0.141$ and $E[L] = 2237.82$. The increase in performance over a single monitoring period is overwhelmed by the decrease in overall performance caused by the rapid growth rate of the non-visible fraction of system operation.

It may be possible to achieve improved single monitoring period performance with logarithmic data analysis algorithms, which do not have a growth rate as rapid as that of quadratic algorithms. Such an algorithm with $p = \frac{k}{n^2}$, $q = \frac{k'}{n}$, and $a = 2.3$, $k = 0.02$, and $k' = 2\times10^{-6}$, is not burdened by the rapid detection probability decrease and expected detection latency increase characteristic of a quadratic data analysis algorithm. The maximum detection probability occurs at $n = 21$, where $D = 0.833$, $E[L] = 114.21$, and $E[F] = 12.444$. When $n = 100$, $D = 0.541$ and $E[L] = 573.77$ showing the more gradual degradation of performance inherent in the logarithmic algorithm. Hence, such algorithms seem to be potentially useful over a wide range of block lengths.

For each data analysis algorithm studied, the expected detection latency appears to increase at the same rate as the amount of time spent analyzing a captured data block, after some initial deviation. This is indeed true, as shown by the following theorem, stated without proof.

Theorem 3: For any data analysis algorithm with analysis time T, such that $n = O(T)$, and $p \to 0$, $E[L] \sim cT$, for some constant c.

8. Conclusions

A new procedure for upset detection in complex systems, called the data block capture and analysis process, has been presented. The upset detection capability of this process has been characterized in terms of process parameters by deriving the probability of correct detection, expected number of false alarms per upset, and expected latency of the process. It has been shown that the detection probability can be maximized by a proper choice of parameters. In addition, it has been shown that improvement in any one of our detection measures can be achieved at the expense of one or both of the other measures. In particular, the detection probability can be increased by allowing a greater number of false alarms to occur. Examples have been shown where the process can detect almost 90% of all upsets while maintaining reasonable false alarm production and detection latency. These analytical results suggest that the data block capture and analysis monitoring process is a very effective means of providing upset detection in complex systems.

References

1. L.F. Pau, *Failure Diagnosis and Performance Monitoring*, pp. 5-7. New York: Marcel Dekker, Inc., 1981.
2. M. Schmid, R. Trapp, A. Davidoff, and G.M. Masson, "Upset Exposure by Means of Abstraction Verification", in *Proc. 12th Fault-Tolerant Computing Symposium*, IEEE Comp. Soc. Publ., pp. 237-244, 1982.
3. M. Schuette, J. Shen, D. Siewiorek, and Y. Zhu, "Experimental Evaluation of Two Concurrent Error Detection Schemes", in *Proc. 16th Fault-Tolerant Comp. Symp.*, IEEE Comp. Soc. Publ., pp. 138-143, 1986.
4. Howard Taylor and Samuel Karlin, *An Introduction to Stochastic Modeling*. Orlando: Academic Press, Inc., 1984.
5. Daniel Siewiorek and Robert Swarz, *The Theory and Practice of Reliable System Design*, pp. 246-255. Bedford, MA: Digital Press, 1982.
6. J. Lala and A. Hopkins, "Survival and Dispatch Probability Models for FTMP Computer", in *Proc. 8th Fault-Tolerant Comp. Symp.*, IEEE Comp. Soc. Publ., pp. 37-43, 1978.
7. D. Miller, "Reliability Calculation Using Randomization for Markovian Fault-Tolerant Computing Systems", in *Proc. 13th Fault-Tolerant Comp. Symp.*, IEEE Comp. Soc. Publ., pp. 284-289, 1983.
8. W.K. Fuchs, "A Specification-Based Approach to Concurrent Structure Verification in Multiprocessor Systems," in *Proc. Internat'l Conf. on Comp. Design*, IEEE Comp. Soc. Publ., pp. 375-378, 1986.
9. A. Avizienis, "Fault Tolerance by means of External Monitoring of Computer Systems", in *National Comp. Conf. Proc.*, AFIPS Press, pp. 27-40, 1981.
10. J. Eifert and J. Shen, "Processor Monitoring Using Asynchronous Signatured Instruction Streams", in *Proc. 14th Fault-Tolerant Comp. Symp.*, IEEE Comp. Soc. Publ., pp. 394-399, 1984.
11. L.F. Pau, "Applications of Pattern Recognition to the Diagnosis of Equipment Failures", in *Pattern Recognition Journal*, Vol. 6, No. 3, pp. 3-11, August 1974.
12. R. Maxion, "Distributed Diagnostic Performance Reporting and Analysis", in *Proc. of the Internat'l Conf. on Comp. Design*, IEEE Comp. Soc. Publ., pp. 362-365, 1986.
13. T. Lin and D. Siewiorek, "Towards On-Line Diagnosis and Trend Analysis", in *Proc. of the Internat'l Conf. on Comp. Design*, IEEE Comp. Soc. Publ., pp. 370-374, 1986.
14. Capt. G. Montgomery, Conf. Intro. *Proc. of the Air Force Workshop on Artificial Intelligence Applications for Integrated Diagnostics*, Center for Appl. Artif. Intell., Univ. of Colorado at Boulder, 1986.