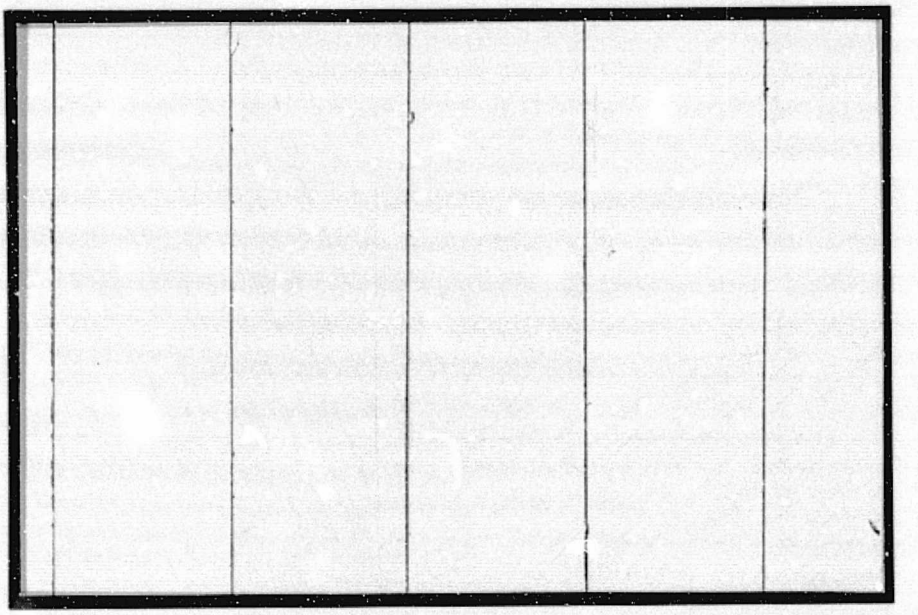
ELECTRICAL

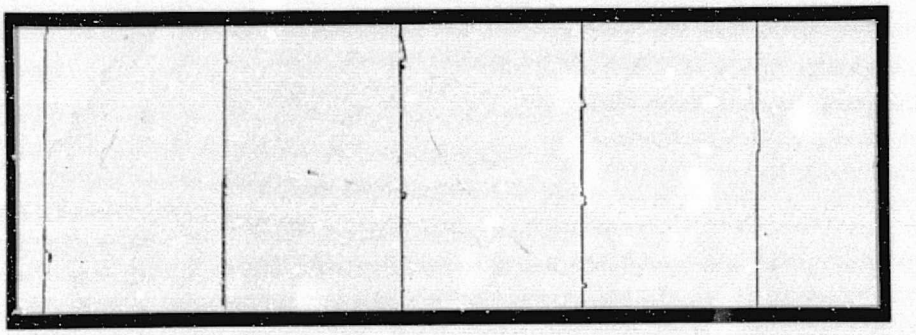
ENGINEERING

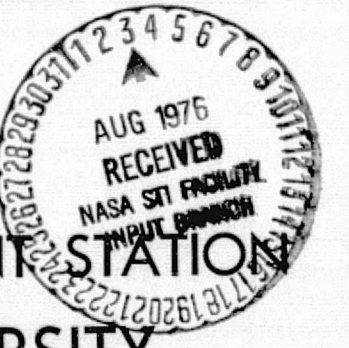


(NASA-CR-149942) RELIABILITY AND COVERAGE ANALYSIS OF NON-REPAIRABLE FAULT-TOLERANT MEMORY SYSTEMS Final Technical Report (Auburn Univ.) 106 p HC $5.50 CSCL 09B

N76-27885

Unclas
G3/60 45852





ENGINEERING EXPERIMENT STATION

AUBURN UNIVERSITY

AUBURN, ALABAMA

FINAL TECHNICAL REPORT

RELIABILITY AND COVERAGE ANALYSES OF NON-REPAIRABLE FAULT-TOLERANT MEMORY SYSTEMS

Prepared by

G. W. Cox and
B. D. Carroll
Electrical Engineering Department
Auburn University
Auburn, Alabama 36830

Contract NAS8-26930

George C. Marshall Space Flight Center
National Aeronautics and Space Administration
Marshall Space Flight Center, Alabama 35812

July 1, 1976

TABLE OF CONTENTS

ABSTRACT

A method was developed for the construction of probabilistic state-space models for non-repairable systems. This method allows the construction of system models with considerably fewer states than the model resulting from more traditional approaches. Models were developed for several systems which achieved reliability improvement by means of error-coding, modularized sparing, massive replication and other fault-tolerant techniques.

From the models developed, sets of reliability and coverage equations for the systems were developed. Comparative analyses of the systems were performed using these equation sets. In addition, the effects of varying subunit reliabilities on system reliability and coverage were described. The results of these analyses indicated that a significant gain in system reliability may be achieved by use of combinations of modularized sparing, error coding and software error control. For sufficiently reliable system subunits, this gain may far exceed the reliability gain achieved by use of massive replication techniques, yet result in a considerable saving in system cost.

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

As the field of computing system design has developed, the need for reliable computers has become crucial. Advances in the aerospace area in particular have necessitated the design of computing systems that are highly reliable and capable of operation in a non-repairable environment. In many other system applications, while repair may be possible, an interruption in system operation is unacceptable.

Due to the large number of components which it contains, the main memory has typically been the most unreliable subunit of the computing system [1]. Since this subunit contributes a high percentage of total system size and weight and many systems must operate within limitations in these areas, massive replication techniques for memory reliability improvement are often not applicable. Thus, much research has been performed to find methods of memory reliability improvement by other means.

Several methods of improvement have been utilized. One such method is the development of error-control codes for use in the memory array. Also, modular memory organizations have been designed in an attempt to limit the possible ways that stored-word errors can occur and to ease system reconfiguration problems. The example systems of this paper utilize both coding and modular design for improved system reliability. These systems are described in Chapter II.

A method is presented in this paper for calculating the reliability

and coverage of these systems. This method allows the construction of system state diagrams with fewer states than occur in many state-space approaches. The method used is described in Chapter III with example results shown in Chapter V.

Background

Replication on the membry system level [2, 3] has been used as a solution for the ultra-reliable memory problem. Substantial increase in memory reliability has resulted in many cases. System cost, however, has increased linearly with the number of duplicated systems. Other limiting factors, such as system weight and size, have prevented the use of massive replication techniques in many applications.

A number of proposed and actual systems [1, 4, 5, 6, 7, 8] have utilized a modular concept of memory arrangement, usually in conjunction with error coding. In addition, a number [9, 10, 11, 12] of burst-error correcting codes have been developed. These codes are well suited for use in word-slice oriented memories in which a majority of the word errors may be expected to occur within groups of word bits.

Several articles [13, 14, 15] have developed reliability calculation procedures for the fault-tolerant memory problem. Many others [16, 17, 18] have shown calculation procedures for fault-tolerant systems in general. When a state-space approach to system modeling has been taken, the time allowed for state transitions to occur is generally $\lim_{\Delta t \to 0} \Delta t$. Typically, only one system event is allowed to occur in this transition interval. Multiple states are then necessary to represent all possible combinations of conditions of system subunits resulting in large numbers of states for highly complex systems.

II. FAULT TOLERANT MEMORY DESCRIPTION

In this chapter, several fault-tolerant memory systems are described. The first section describes a system which is taken as a basis for the comparison of related systems. Several related systems are described in the second section. Reliability and coverage computations for these systems will be examined in following chapters.

Basic System

The basic computer system to be analyzed has been designed for use in extended aerospace missions. It was desirable to implement the computer memory in a manner so as to be within weight, size, and economic limitations, yet be highly fault-tolerant.

A modular design approach has been undertaken in which the memory array is made up of memory slices, each of which contains the same bit location of all memory words. If n words are contained in the memory and each word is k bits long, then there must be k memory modules and each module must contain n bits. These modules will be referred to as on-line bit planes.

In addition to the bit planes already discussed, the system contains identically-sized spare bit planes which may be switched to replace any failed on-line bit plane. The arrangement of on-line and spare bit planes is shown in Figure 1. The functional orientation of memory words is shown in Figure 2.

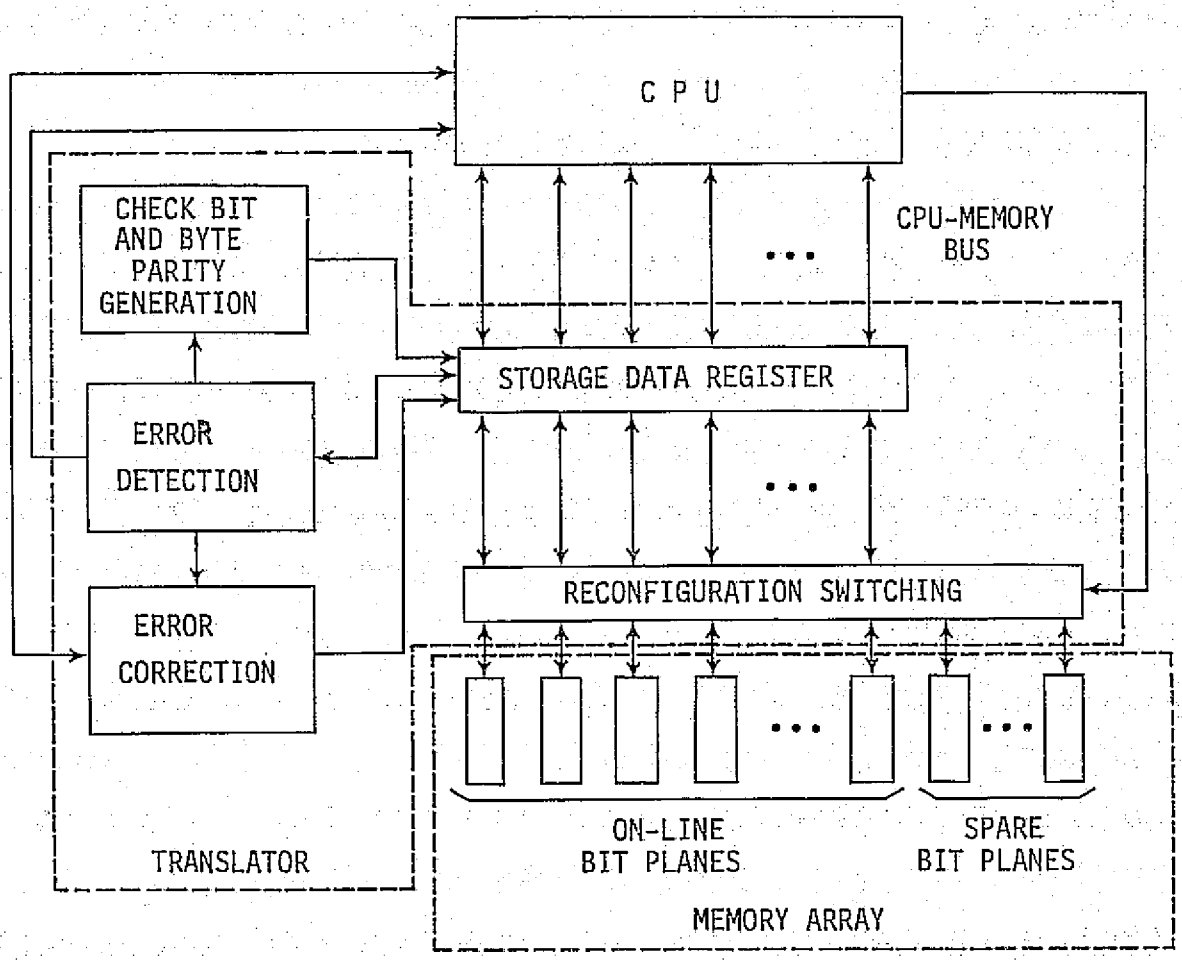


FIGURE 1. Functional Representation of Basic System

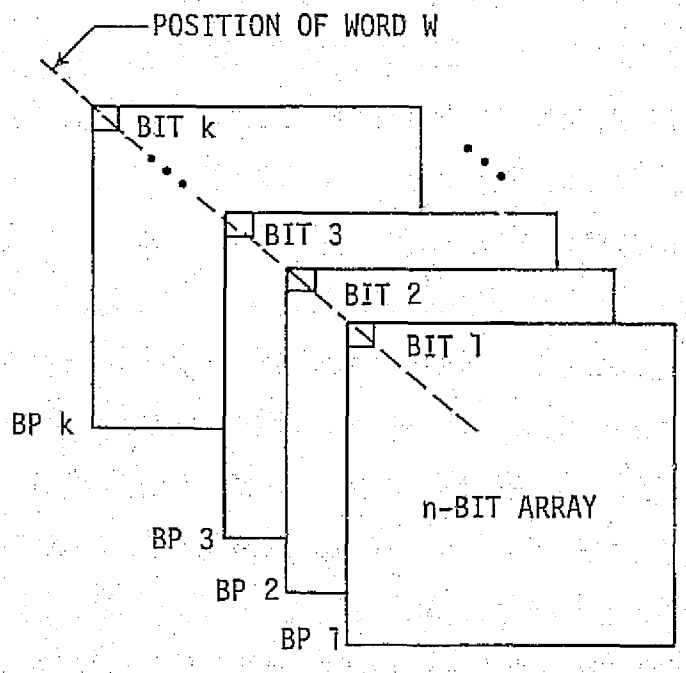


FIGURE 2. Functional Orientation of a Typical Memory Word

A single-error-correcting/double-error-detecting (22, 16) code [19] is used for memory data word encoding. This code has the property that any odd number of errors in a codeword will produce an odd-weighted error syndrome. Double errors will produce a non-zero even-weighted error syndrome and higher numbers of even errors will produce even-weighted (including all zero) error syndromes. These features of the code will be further discussed in a later section.

External to the memory, data words are encoded using only 2 byte parity bits. For this reason, circuitry which translates between the two codes is necessary for use in memory write and read cycles. This function is performed by the memory translator. In addition, the translator contains circuitry for the correction of single bit errors and detection of multiple bit errors in memory words, and control of the reconfiguration switching circuitry which directs each word bit to the appropriate bit plane. These functions will now be examined.

For a memory write operation, the translator accepts a byte-parity encoded word from the CPU-memory bus. The byte parity bits are saved and the check bits for the SEC/DED code are generated. A validity check is then made by a comparison of the saved byte parity bits with the generated check bits. If no error is found, the data word with SEC/DED check bits appended is stored in the memory. If an error is found, a program interrupt is sent to the CPU.

For a memory read operation, the requested encoded word is read from the memory array and placed in the storage data register (SDR). The error syndrome for the word is formed from the encoded word and if a zero

(no error) syndrome is signaled, the byte parity bits for the data word are formed and the word is transmitted on the data bus.

An odd-weight (odd error) syndrome signal causes a bit inversion to be made by the single error correction circuitry. The error syndrome for the corrected word is then generated. If no error is signalled, then it is assumed that there was a single error in the encoded word. The byte parity bits are generated and the word is transmitted on the data bus. If an error is signaled, a program interrupt is generated.

When the translator receives the information that a certain designated spare bit plane is to replace an on-line bit plane, it must reconfigure the memory array input and output switching to reflect this change. Memory input switching is reconfigured first. Each memory word is then read from the on-line array, corrected if necessary, and re-written in the on-line array with the spare bit plane replacing the designated on-line bit plane. After all memory words have been read and restored, the memory array output switching is reconfigured appropriately.

The decision to replace an on-line bit plane may be arrived at by use of various switching strategies. It is assumed for the basic system that the reconfiguration signal is issued by the CPU as a result of error signals received from the translator. It is also assumed that the switching strategy is to replace a bit plane as soon as it is detected that the bit plane contains an error. Another switching strategy will be discussed in a following section.

In the basic system, there is assumed to be no facility available for the correction of multiple errors. If system failure is defined to

be the occurrence of a non-correctable error, then the occurrence of more than one error in a single memory word will constitute failure for this system. For purposes of system modeling, the occurrence of simultaneous failures in multiple bit planes is assumed to be equivalent to the occurrence of multiple errors in a single memory word. System failure, then, will occur when more than one on-line bit plane has failed.

Spare bit planes are assumed to operate in a mode identical to the on-line bit planes prior to their insertion into the on-line array. Spare bit planes, then, fail with the same characteristics as the on-line units. It is also assumed that after a bit plane has been removed from the on-line array, it is never re-inserted. A bit plane which has been replaced is called an unavailable spare. A spare bit plane which has not been inserted into the on-line array and which may or may not be failed is an available spare.

The system, then, may be divided into subsystems by function. These subsystems are:

1) The on-line memory array consisting of a number of bit planes,
2) The spare bit plane array including both available and unavailable spares,
3) The error detection circuitry of the translator,
4) The error correction circuitry of the translator,
5) The reconfiguration switching array, and
6) The encoding and decoding subsystems of the translator.

References will be made to these subsystems in following sections.

Alternate Designs

Several fault-tolerant memory systems which are related to the basic system have been studied. Four of these systems will be described in this section.

The non-spared system is identical to the basic system except that no spare bit planes are provided. In addition, no reconfiguration switching circuitry is included, since such circuitry would have no use in this system. Comparisons made between this system and the basic system will show the relative improvement to be gained by the use of the spare bit plane approach.

The TMR system consists of three systems of the non-spared type in a triple modular redundant configuration. The functional operation of this system may be described as follows:

1) For a memory write operation, SEC/DED-encoded word is stored in the same logical location in all three memories.

2) For a memory read operation, the requested memory location is read in all three memories. Single error correction is performed independently by the systems and byte parity bits are generated in each case. The three byte-parity encoded words are then voted on by majority logic in a bit-by-bit fashion. The output word is constructed by using the majority vote for each bit. If the constructed word is still a codeword, it is transmitted on the data bus. If it is not a codeword, an error program interrupt is generated.

This system, then, will produce the correct output word as long as at least two of the three memories can construct the correct word. A functional depiction of this system is shown in Figure 3.

The duplicated system is composed of two identical non-spared subsystems. Data to be loaded is stored in the same logical location in

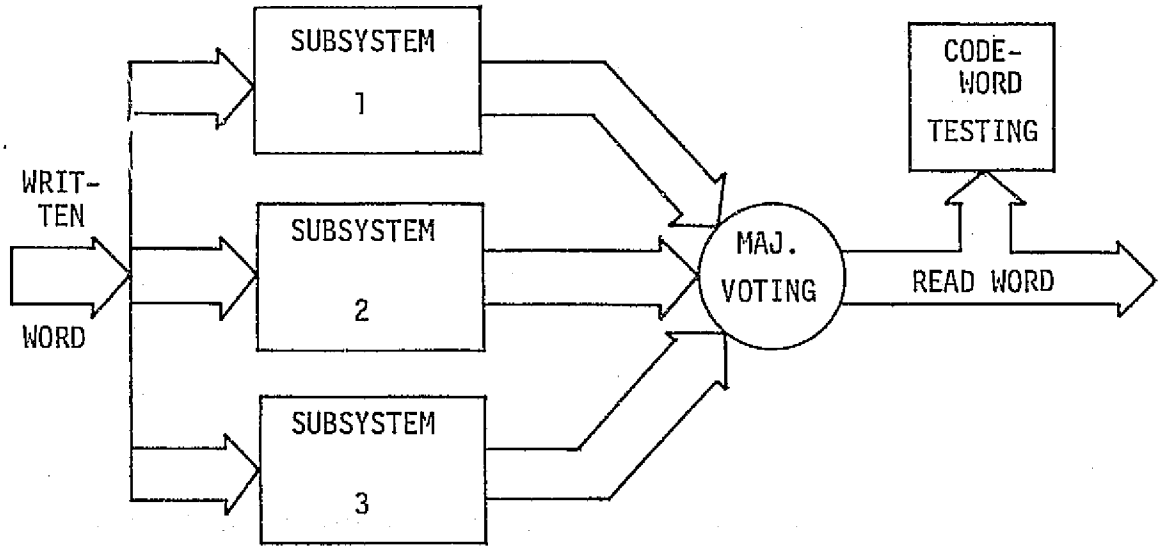


FIGURE 3. Functional Representation of TMR System

both subsystems. Data read from the system is read from only one memory. If a non-correctable error is signalled by the on-line unit, output bussing is switched to the other unit and the data is read from the same location. If both subsystems signal a non-correctable error in the same memory word, an error program interrupt is generated.

The double-error-correcting system is a modified version of the basic system which will correct double errors and detect a triple error which produces a single error syndrome. The additional features are achieved by the use of software routines [20] which are CPU implemented. Since double errors are correctable in this system, a reconfiguration switching strategy is assumed in which an on-line bit plane is replaced only if it contains an erroneous bit position of a word which has two or more errors. This system will be more fully discussed in a later chapter.

III. RELIABILITY MODEL DEVELOPMENT

In this chapter, a generalized method for the computation of reliability, the probability of satisfactory operation, and coverage, the probability of recovery if a failure occurs, for a system is described. This method is applied to form sets of reliability and coverage equations for the basic system described in the preceeding chapter. Computer implementation of these equations is examined in the last section.

General Techniques

Prior to the development, it is appropriate that certain notation be defined. A listing of notation used is shown in Table 1.

For the purpose of reliability computation, the performance of a device may often be represented as a set of states and state transitions. Suppose, for example, that a certain non-repairable device has three possible modes of operation:

1) Satisfactory operation,

2) Degraded operation caused by event A which occurred while the device was operating satisfactorily, and

3) Unsatisfactory operation caused by event B which occurred while the device was operating satisfactorily or by event C which occurred while the device was operating in its degraded mode.

These three modes of operation form three natural states for the device.

TABLE 1. Definition of Notation

Notation	Meaning
$P(x)$	Probability of the occurrence of event x
$P(x,r)$	Probability of the occurrence of events x and r
$P(x \text{ or } r)$	Probability of the occurrence of event x or event r or both
$P(x/r)$	Probability of the occurrence of event x given that event r has occurred
$P_x(t,\Delta t)$	Probability of the occurrence of event x in the time period from t to $t + \Delta t$
$P_{i,j}(t,\Delta t/i)$ or $P_{i,j}(t,\Delta t)$	Probability of the occurrence of a transition from state i to state j in the time period from t to $t + \Delta t$ given that the state at time t is i
$P_i(t)$	Probability that the system is in state i at time t
$P_i(t + \Delta t,j)$	Probability that the system is in state i at time $t + \Delta t$ and that it was in state j at time t
$P_i(t + \Delta t/j)$	Probability that the system is in state i at time $t + \Delta t$ given that it was in state j at time t
$r_j(t)$	Probability that component j is non-failed at time t
$r(t)$	Probability that a generalized component is non-failed at time t

If the assumption is made that the system is operating in state 1 (satisfactory operation) at time t, then the probability that the system will be in state 2 (degraded operation) at time $t + \Delta t$, a small interval of time later, is the probability that event A occurred in the time period from t to $t + \Delta t$, in equation form

$$P_{1,2}\ (t,\Delta t/1) = P_A\ (t,\Delta t)$$

where $P_{1,2}\ (t,\Delta t/1)$ is the probability of a transition from state 1 to state 2 in time t to $t + \Delta t$ given that the system was in state 1 at time t and $P_A\ (t, \Delta t)$ is the probability of the occurrence of event A in the same time period.

In a similar manner, the transition probabilities into state 3 (the failed state) are

$$P_{1,3}\ (t,\Delta t/1) = P_B\ (t, \Delta t), \text{ and}$$

$$P_{2,3}\ (t,\Delta t/2) = P_C\ (t, \Delta t).$$

This state model can be described graphically by a state diagram as shown in Figure 4.

An equivalent form of device state representation is a matrix $\overline{T}$ which has as its i,j entry $P_{i,j}\ (t,\Delta t/i)$ for $i \neq j$ and $1 - \sum_{\substack{k=1 \\ k\neq i}}^{N} T[i,k]$ for $i = j$, where N is the number of device states. The $\overline{T}$ matrix for the example device is given below.

$$\overline{T} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 1-P_A(t,\Delta t) & P_A(t,\Delta t) & P_B(t,\Delta t) \\ -P_B(t,\Delta t) & & \\ 0 & 1-P_C(t,\Delta t) & P_C(t,\Delta t) \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

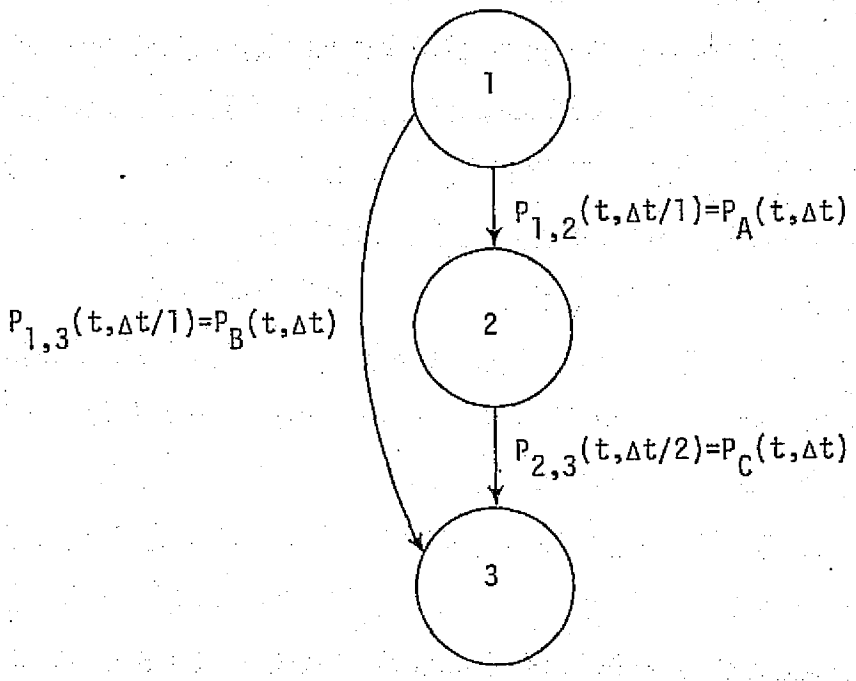


FIGURE 4. State Diagram and Transition Probabilities for Example Device

Deleting the $(t,\Delta t)$ subscripts yields

$$\overline{T} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 1-P_A-P_B & P_A & P_B \\ 0 & 1-P_C & P_C \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

The probability that the system is in any given state at time $t + \Delta t$ may be expressed in terms of the transition probabilities and the distribution of state probabilities at time t. These equations may be obtained by assuming that the system is operating in a state i at time t and by computing the probability of the occurrence of the transition event to state j in the time period from t to Δt.

For the development, the following notational convention will be used.

$$P(\text{system operating in state } i \text{ at } t + \Delta t \text{ given that the system was in state } j \text{ at time } t) = P_i(t + \Delta t/j).$$

To obtain the equation for $P_1(t + \Delta t/1)$, it must be considered that for the system to be in state 1 at time $t + \Delta t$, no state transition out of state 1 may occur between t and $t + \Delta t$. Then the complement of the two state transitions out of state 1 must be combined as follows:

$$\begin{aligned} P_1\,(t + \Delta t/1) &= (1 - P_{1,2}\,(t,\Delta t/1))(1 - P_{1,3}\,(t,\Delta t/1)) \\ &= 1 - P_{1,2}\,(t,\Delta t/1) - P_{1,3}\,(t,\Delta t/1) \\ &\quad + P_{1,2}\,(t,\Delta t/1)\,P_{1,3}\,(t,\Delta t/1) \end{aligned}$$

If Δt is defined to be a period of time which is small enough to allow only one state transition to take place, the last term in this equation becomes negligible since it defines the probability of more than one state transition occurring in time t to $t + \Delta t$. Then,

$$P_1 (t + \Delta t/1) = 1 - P_{1,2} (t,\Delta t/1) - P_{1,3} (t,\Delta t/1)$$
$$= 1 - P_A (t,\Delta t) - P_B (t,\Delta t)$$

Recalling that

$$P(X/Y) = \frac{P(X,Y)}{P(Y)} \qquad [25],$$

then

$$\frac{P_1(t + \Delta t,1)}{P_1(t)} = 1 - P_A (t,\Delta t) - P_B (t,\Delta t)$$

$$P_1 (t + \Delta t,1) = P_1(t)(1 - P_A(t,\Delta t) - P_B(t,\Delta t)).$$

Since there are no transition paths into state 1, the event "the system is in state 1 at $t + \Delta t$" implies the event "the system is in state 1 at time t." Then,

$$P_1(t + \Delta t, 1) = P_1(t + \Delta t)$$

So,

$$P_1(t + \Delta t) = P_1(t)(1 - P_A(t,\Delta t) - P_B(t,\Delta t)).$$

There are two ways for the system to be in state 2 at time $t + \Delta t$. Either the system was operating in state 1 at time t and the transition

from state 1 to state 2 occurred in the time period from t to t + Δt, or the system was operating in state 2 at time t and no transition event out of state 2 occurred in t to t + Δt.

The equation for $P_2(t + \Delta t)$ may then be formed as follows:

$$\begin{aligned} P_2(t + \Delta t) &= P_2(t + \Delta t, 1) + P_2(t + \Delta t, 2) \\ &= P_{1,2}(t,\Delta t/1)P_1(t) + (1 - P_{2,3}(t,\Delta t/2))P_2(t) \\ &= P_A(t,\Delta t)P_1(t) + (1 - P_C(t,\Delta t))P_2(t). \end{aligned}$$

By similar reasoning,

$$\begin{aligned} P_3(t + \Delta t) &= P_3(t + \Delta t,1) + P_3(t + \Delta t,2) + P_3(t + \Delta t,3) \\ &= P_{1,3}(t,\Delta t/1)P_1(t) + P_{2,3}(t,\Delta t/2)P_2(t) + P_3(t + \Delta t,3) \\ &= P_B(t,\Delta t)P_1(t) + P_C(t,\Delta t)P_2(t) + P_3(t + \Delta t,3). \end{aligned}$$

Since there are no transition paths out of state 3, the probability that the system is in state 3 at t + Δt and that it was in state 3 at time t is the probability of the latter condition, or

$$P_3(t + \Delta t,3) = P_3(t).$$

By substitution, the equation for $P_3(t + \Delta t)$ becomes

$$P_3(t + \Delta t) = P_B(t,\Delta t)P_1(t) + P_C(t,\Delta t)P_2(t) + P_3(t).$$

In general, the state probability equation for state i is

$$P_i(t + \Delta t) = \sum_{\substack{j=1 \\ j\neq i}}^{n} P_{j,i}(t,\Delta t/j)P_j(t) + (1 - \sum_{\substack{k=1 \\ k\neq i}}^{n} P_{i,k}(t,\Delta t/i))P_i(t)$$

where n is the number of system states. The first summation in this equation represents the sum of the probabilities of all possible transitions into state i from another state. The coefficient of $P_i(t)$ is the probability that no transition out of state i will occur in t to t + Δt given that the system was in state i at time t.

Since for each term of the form $P_{u,v}(t,\Delta t/u)$, the u inside the parentheses is redundant, this probability may be represented as $P_{u,v}(t,\Delta t)$ where the deleted u is understood.

The general state probability equation then becomes

$$P_i(t + \Delta t) = \sum_{\substack{j=1 \\ j\neq i}}^{n} P_{j,i}(t,\Delta t)P_j(t) + \left(1 - \sum_{\substack{k=1 \\ k\neq i}}^{n} P_{i,k}(t,\Delta t)\right)P_i(t). \qquad (3\text{-}1)$$

If vectors $\underline{P}(t + \Delta t)$ and $\underline{P}(t)$ are defined by

$$\underline{P}(t + \Delta t) = \begin{vmatrix} P_1(t + \Delta t) \\ P_2(t + \Delta t) \\ \vdots \\ P_n(t + \Delta t) \end{vmatrix}, \quad \underline{P}(t) = \begin{vmatrix} P_1(t) \\ P_2(t) \\ \vdots \\ P_n(t) \end{vmatrix}$$

then equation (3-1) may be represented in matrix form as

$$\underline{P}(t + \Delta t) = \overline{T}^T \times \underline{P}(t) \qquad (3\text{-}2)$$

where $\overline{T}$ is previously defined and $\overline{T}^T$ is the transpose of $\overline{T}$.

In a complex system, the events which cause state transitions may be composed of many subevents which must occur for the transition event to occur. It may be more desirable to work with these subevent probabilities

than to attempt to determine the probability of the overall event. For this reason, it is necessary to analyze the possible types of subevents and to be able to calculate the probability of occurrence of each type.

For any transition event i,j with probability of occurrence $P_{i,j}(t,\Delta t)$ it is possible to place any subevent in exactly one of the following six event classes:

1. The failure event of a system component or component group prior to time $t + \Delta t$.
2. The non-failure event of a system component or component group prior to time $t + \Delta t$.
3. The failure event of a system component or component group in the time period from t to $t + \Delta t$.
4. The non-failure event of a system component or component group in the time period from t to $t + \Delta t$.
5. The failure event of a system component or component group in the time period from t to $t + \Delta t$ given its non-failure prior to t.
6. The non-failure event of a system component or component group in the time period from t to $t + \Delta t$ given its non-failure prior to t.

In order to compute the probability of events in each of these classes, it is necessary to first examine the basis for the computation of failure probabilities.

Each system component or component group has associated with it a failure probability density function, f(t). In the general case,

$$f(t) = -\frac{dr(t)}{dt} \quad \text{and} \quad \int_0^\infty f(t)dt = 1. \quad [23]$$

The apriori probability of component (group) failure in the time period from t_1 to t_2 may be expressed as

$$P_F = \int_{t_1}^{t_2} f(t)dt, \text{ and}$$

the apriori reliability of the component (group) at time t_3 is

$$r(t_3) = 1 - \int_0^{t_3} f(t)dt$$

$$= \int_{t_3}^{\infty} f(t)dt.$$

If the assumption is made that, at time $t_1 \neq \emptyset$, a particular component (group) is non-failed then the probability of failure prior to this time is $\emptyset$ and the probability of failure after t_1 is 1. Then,

$$\int_{t_1}^{\infty} f'(t)dt = 1.$$

From [26], for f(t) exponential, $f'(t)=f(t-t_1)$. For this and following developments, all failure density functions will be assumed to be of the exponential type.

If the failure probability of this component in the time period from t_1 to $t_1 + \Delta t$ is of concern, then

$$\int_{t_1}^{t_1 + \Delta t} f'(t)dt = \int_{t_1}^{t_1 + \Delta t} f(t-t_1)dt$$

$$= \int_0^{\Delta t} f(t)dt$$

which is the probability that the component (group) will fail in the interval from t to t + Δt given that it is non-failed prior to time t.

By use of these concepts, the subevent probabilities for each class may now be computed as follows:

$$\text{Class 1.} \quad P_1 = \int_0^{t+\Delta t} f(t)dt$$

$$= 1 - \int_{t+\Delta t}^{\infty} f(t)dt$$

$$= 1 - r(t + \Delta t).$$

$$\text{Class 2.} \quad P_2 = 1 - \int_0^{t+\Delta t} f(t)dt$$

$$= 1 - P_1$$

$$= r(t + \Delta t).$$

$$\text{Class 3.} \quad P_3 = \int_t^{t+\Delta t} f(t)dt$$

$$= \int_t^{\infty} f(t)dt - \int_{t+\Delta t}^{\infty} f(t)dt$$

$$= r(t) - r(t + \Delta t).$$

$$\text{Class 4.} \quad P_4 = 1 - \int_t^{t+\Delta t} f(t)dt$$

$$= 1 - P_3$$

$$= 1 - r(t) + r(t + \Delta t).$$

$$\text{Class 5.} \quad P_5 = \int_0^{\Delta t} f(t)dt$$

$$= 1 - r(\Delta t).$$

$$\text{Class 6.} \quad P_6 = 1 - \int_0^{\Delta t} f(t)dt$$

$$= 1 - P_5$$

$$= r(\Delta t).$$

To completely specify the state probabilities, it is necessary to select a base time, t_{base}. In general, t_{base} may be any time at which all state probabilities are known. The following discussion will assume that t_{base} is 0. It is common to denote one system state, m, as the starting state and assume that

$$P_m(t = t_{base} = 0) = 1, \text{ and}$$

$$P_n(t = t_{base} = 0) = 0 \text{ for all } n \neq m.$$

The state probabilities may be computed for any $t > 0$ if:

1. All state transition equations are known, and
2. All system component (group) reliability equations are known.

To obtain a closed-form solution for each probability equation, it is common to rearrange each equation into its differential form and solve the equation set simultaneously. By making simplifying assumptions, the equation set may be approximated by a set of linear differential equations. For systems with a large number of states, however, the simultaneous solution problem may become quite involved. In addition

if the analysis of a related system is desired, only a slight difference in architecture or operation may necessitate the re-derivation of all state equations.

If computer evaluation of state probabilities is possible, however, the open form of the state probability equations may yield satisfactory results at considerable savings in effort. In addition, no simplifying assumptions need be made to assure equation linearity. State probability equations to be derived in this paper will remain in this open form.

Reliability Equations

For the basic memory system, the insertion of each spare bit plane on-line performs a natural partitioning of system states. By determining the number of available spares it is possible to define the state of the system. If the basic system has k bits per memory word and s spare bit planes initially available, the system state diagram may be constructed as shown in Figure 5.

For each state i ($1 \leq i \leq s+1$) in this diagram, the system is operating with exactly $s - i + 1$ spare bit planes available, and no failed bits in any word (no failed bit planes on-line). In state s+2, the system has suffered a single bit plane failure but there are no available spare bit planes to replace the failed on-line bit plane. The system must use the single-error-correction circuitry to correct one error in each memory word in this state. The FAIL state is the system state when an uncorrectable error has occurred.

The development of transition and state probability equations for this system will now be shown.

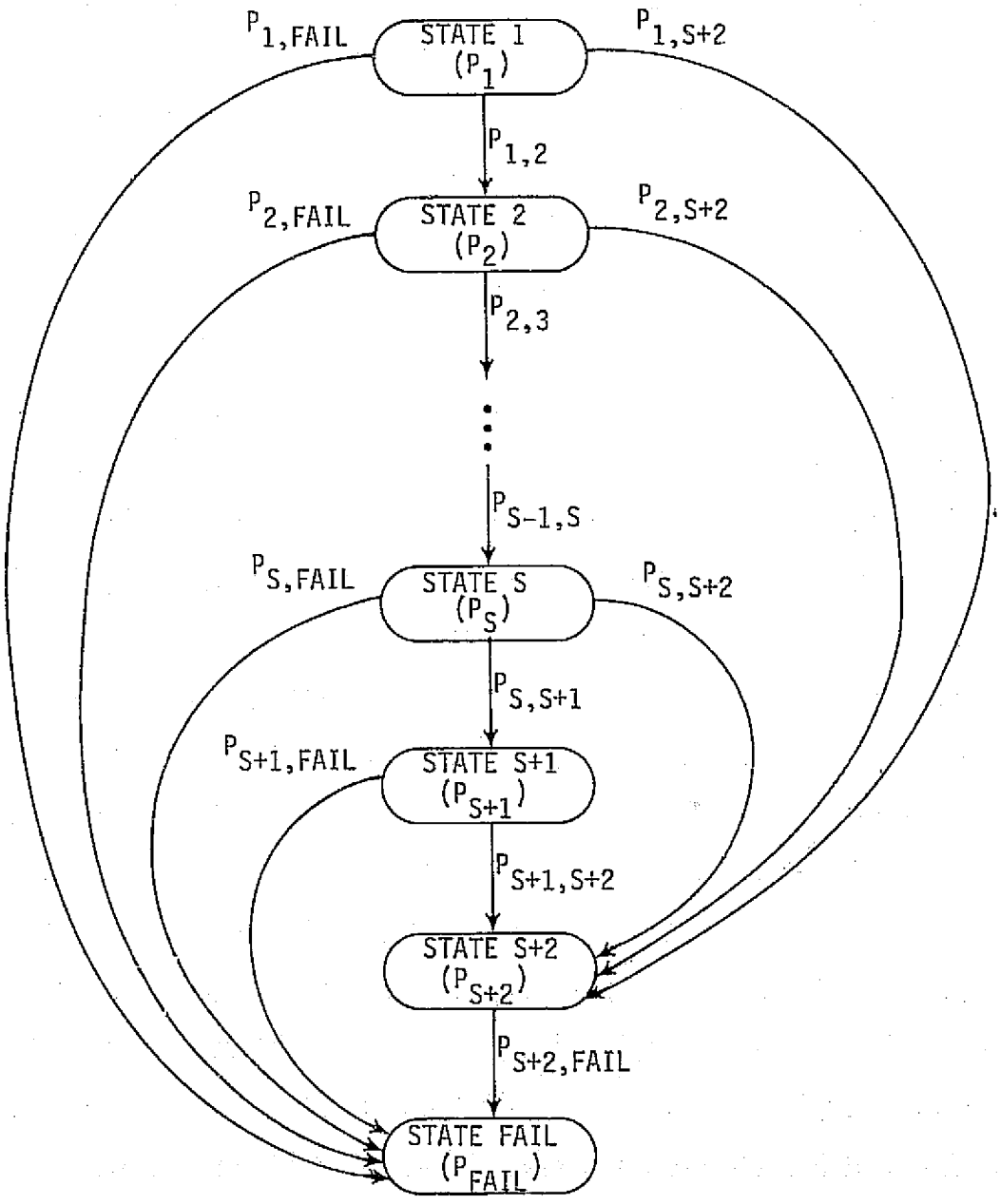


FIGURE 5. State Diagram for Basic System

For the transition event to occur from state i to state i+1 ($1 \leq i \leq s$) in the time period from t to $t + \Delta t$, exactly four subevents must occur. These subevents are:

E_1: The failure of exactly one on-line bit plane in the time period from t to $t + \Delta t$ given the non-failure of all on-line bit planes prior to t,

E_2: The non-failure of the system error detector (group) prior to time $t + \Delta t$,

E_3: The non-failure of the system reconfiguration switching circuitry (group) prior to time $t + \Delta t$, and

E_4: The non-failure of at least one available spare bit plane prior to time $t + \Delta t$.

These subevents belong to classes 5, 2, 2, and 2, respectively.

The subevent probabilities may be computed as:

$$P_{E_1}(t,\Delta t) = \binom{k}{1}(r(\Delta t))^{(k-1)}(1-r(\Delta t))$$

$$= k\,(r(\Delta t))^{(k-1)}(1-r(\Delta t))$$

$$P_{E_2}(t,\Delta t) = r_d(t + \Delta t)$$

$$P_{E_3}(t,\Delta t) = r_s(t + \Delta t)$$

$$P_{E_4}(t,\Delta t) = 1-(1 - r(t + \Delta t))^{(s - i + 1)} .$$

where all symbols are as defined in Tables 1 and 2.

TABLE 2. Definition of Reliability Symbols for Basic System

Symbol	Meaning
$r(t)$	Reliability of an on-line or available spare bit plane at time t.
$r_d(t)$	Reliability of the system error detector (group) at time t.
$r_c(t)$	Reliability of the system error corrector (group) at time t.
$r_s(t)$	Reliability of the system reconfiguration switching circuitry (group) at time t.

Assuming subevent independence, then

$$P_{i,i+1}(t,\Delta t) = P_{E_1}(t,\Delta t)\, P_{E_2}(t,\Delta t)\, P_{E_3}(t,\Delta t)\, P_{E_4}(t,\Delta t)$$

$$= k\,(r(\Delta t))^{(k-1)}(1-r(\Delta t)) \cdot r_d(t+\Delta t)\, r_s(t+\Delta t)(1-(1-r(t+\Delta t))^{(s-i+1)})$$

$$\text{For } 1 \leq i \leq s.$$

Denote $r_m(t)$ by r_m and $r_m(t+\Delta t)$ by r_m'. Then

$$P_{i,i+1}(t,\Delta t) = k(r(\Delta t))^{(k-1)}(1-r(\Delta t)) \cdot r_d' r_s'(1-(1-r')^{(s-i+1)})$$

$$\text{For } 1 \leq i \leq S.$$

The state transition event from state i to state s+2 ($1 \leq i \leq s+1$) represents a transition of the system from a condition in which no on-line bit planes are failed to a condition in which exactly one on-line bit plane is failed and no non-failed spare is available for replacement.

The subevents composing this transition event are:

$$E_1,\ E_2,\ E_5 \text{ and } [E_6 \text{ or } (E_4 \text{ and } E_7)]$$

where E_1, E_2, and E_4 are as previously defined and E_5, E_6, and E_7 are as described below.

E_5: The non-failure of the system error correction (group) prior to time $t + \Delta t$.

E_6: The failure of all s - i + 1 available spare bit planes prior to time t + Δt.

E_7: The failure of the system reconfiguration switching circuitry (group) prior to time t + Δt.

The state transition probability may now be formed as

$$P_{i,s+2}(t,\Delta t) = k(r(\Delta t))^{(k-1)}(1-r(\Delta t))\, r_d' r_c' \cdot [(1-r')^{(s-i+1)} + (1-(1-r')^{(s-i+1)})(1-r_s')],$$

which reduces to

$$P_{i,s+2}(t,\Delta t) = k\,(r(\Delta t))^{(k-1)}(1-r(\Delta t))\, r_d' r_c' \cdot [1 - r_s'(1-(1-r')^{(s-i+1)})]$$

$$\text{For } 1 \leq i \leq s+1.$$

Define for each state i in the system state diagram a probability

$$P_{i,i}(t,\Delta t)$$

which is the probability that, if the system is in state i at time t, no transition out of state i will occur before time t + Δt. Then, for the basic system,

$$P_{i,i}(t,\Delta t) + P_{i,i+1}(t,\Delta t) + P_{i,s+2}(t,\Delta t) + P_{i,FAIL}(t,\Delta t) = 1$$

$$\text{for } 1 \leq i \leq s,$$

$$P_{s+1,s+1}(t,\Delta t) + P_{s+1,s+2}(t,\Delta t) + P_{s+1,FAIL}(t,\Delta t) = 1,$$

$$P_{s+2,s+2}(t,\Delta t) + P_{s+2,FAIL}(t,\Delta t) = 1, \text{ and}$$

$$P_{FAIL,FAIL}(t,\Delta t) = 1.$$

The formulation of the equation for $P_{i,i}(t,\Delta t)$, then, uniquely specifies the equation for $P_{i,FAIL}(t,\Delta t)$. Since, for this system, the non-transition event involves fewer subevents than the transition event to the failed state, these non-transition equations will be developed.

For states 1 through s+1, the only event occurrence which is necessary for the non-transition event to occur in time t to t + Δt is the non-failure of the k on-line bit planes in the same time interval given that all were non-failed at time t. Then

$$P_{i,i}(t,\Delta t) = (r(\Delta t))^k \qquad \text{for } 1 \leq i \leq s + 1.$$

The operation of the system error detector and corrector is required for the system to be in state s+2 at time t. The non-transition event for this state, then contains the subevents

E_8: The non-failure of the system error detector (group) in the time period from t to t + Δt, given its non-failure prior to t, and

E_9: The non-failure of the system error corrector (group) in the time period from t to t + Δt, given its non-failure prior to t.

In addition, none of the k-1 on-line operating bit planes may fail from t to t + Δt. Then

$$P_{s+2,s+2}(t,\Delta t) = (r_d(\Delta t))(r_c(\Delta t))(r(\Delta t))^{(k-1)}$$

For any state i, $(1 \leq i \leq s)$, $P_{i,FAIL}(t,\Delta t)$ may now be computed as

$$P_{i,FAIL}(t,\Delta t) = 1-P_{i,i}(t,\Delta t) - P_{i,s+2}(t,\Delta t) - P_{i,i+1}(t,\Delta t)$$

$$= 1 - (r(\Delta t))^k - k(r(\Delta t))^{(k-1)}(1-r(\Delta t))\, r_d' r_c'$$

$$\cdot [1 - r_s' (1-(1-r')^{(s-i+1)})]$$

$$- k(r(\Delta t))^{(k-1)}(1-r(\Delta t)) r_d' r_s' (1-(1-r')^{(s-i+1)})$$

$$= 1 - (r(\Delta t))^k - k(r(\Delta t))^{(k-1)}(1-r(\Delta t))$$

$$\cdot [r_d' r_c' - r_d' r_c' r_s' (1-(1-r')^{(s-i+1)})$$

$$+ r_d' r_s' (1-(1-r')^{(s-i+1)})]$$

$$= 1 - (r(\Delta t))^k - k(r(\Delta t))^{(k-1)}(1-r(\Delta t)) r_d'$$

$$\cdot [r_c' + r_s' (1-r_c')(1-(1-r')^{(s-i+1)})]$$

For $1 \leq i \leq s$.

For states s + 1 and s + 2,

$$P_{s+1,FAIL}(t,\Delta t) = 1 - P_{s+1,s+1}(t,\Delta t) - P_{s+1,s+2}(t,\Delta t)$$

$$= 1 - (r(\Delta t))^k - k(r(\Delta t))^{(k-1)}(1-r(\Delta t)) r_d' r_c',$$

and

$$P_{s+2,FAIL}(t,\Delta t) = 1 - P_{s+2,s+2}(t,\Delta t)$$

$$= 1 - r_d(\Delta t)\, r_c(\Delta t)(r(\Delta t))^{(k-1)}.$$

The state probability equation for state 1 may be obtained by use of equation (3-1), the general probability equation. The resultant equation is

$$P_1(t + \Delta t) = (1-P_{1,2}(t,\Delta t) - P_{1,s+2}(t,\Delta t) - P_{1,FAIL}(t,\Delta t))P_1(t)$$

$$= P_{1,1}(t,\Delta t)P_1(t) \qquad (3\text{-}3)$$

$$= (r(\Delta t))^k P_1(t).$$

The state probability equation for states 2 through s may be obtained as

$$P_i(t + \Delta t) = P_{i-1,i}(t,\Delta t)\ P_{i-1}(t) + (1-P_{i,i+1}(t,\Delta t)$$

$$- P_{i,s+2}(t,\Delta t) - P_{i,FAIL}(t,\Delta t))\ P_i(t)$$

$$= P_{i-1,i}(t,\Delta t)\ P_{i-1}(t) + P_{i,i}(t,\Delta t)\ P_i(t)$$

$$= k[(r(\Delta t))^{(k-1)}(1-r(\Delta t))r_d'r_s'(1-(1-r')^{(s-i)}]P_{i-1}(t)$$

$$+ (r(\Delta t))^k\ P_i(t)$$

For $2 \leq i \leq s$.

For state s+1, the state probability equation is

$$P_{s+1}(t + \Delta t) = P_{s,s+1}(t,\ \Delta t)\ P_s(t) + P_{s+1,s+1}(t,\Delta t)\ P_{s+1}(t)$$

$$= k\ [(r(\Delta t))^{(k-1)}(1-r(\Delta t))\ r_d'r_s'(1-(1-r'))]P_s(t)$$

$$+ (r(\Delta t))^k\ P_{s+1}(t)$$

So for all states i where $2 \leq i \leq s + 1$,

$$P_i(t + \Delta t) = k\,[(r(\Delta t))^{(k-1)}(1-r(\Delta t))r_d'r_s'(1-(1-r')^{(s-i)})]$$

$$\cdot P_{i-1}(t) + (r(\Delta t))^k P_i(t), \qquad (3\text{-}4)$$

$$\text{for } 2 \leq i \leq s + 1$$

The state probability equation for state s+2 is

$$P_{s+2}(t + \Delta t) = \sum_{j=1}^{s+1} P_{j,s+2}(t,\Delta t)\, P_j(t) + P_{s+2,s+2}(t,\Delta t)\, P_{s+2}(t)$$

$$= \sum_{j=1}^{s+1} k\,(r(\Delta t))^{(k-1)}(1-r(\Delta t))\, r_d'r_c'$$

$$\cdot [1 - r_s'(1-(1-r')^{(s-j+1)})]\, P_j(t)$$

$$+ (r_d(\Delta t))(r_c(\Delta t))(r(\Delta t))^{(k-1)}P_{s+2}(t).$$

Reduction of this equation yields

$$P_{s+2}(t + \Delta t) = k\,(r(\Delta t))^{(k-1)}(1-r(\Delta t))\, r_d'r_c'$$

$$\cdot \sum_{j=1}^{s+1} (1-r_s'(1-(1-r')^{(s-j+1)})\, P_j(t)$$

$$+ (r_d(\Delta t))(r_c(\Delta t))(r(\Delta t))^{(k-1)}P_{s+2}(t). \qquad (3\text{-}5)$$

The state probability equation for state FAIL is

$$P_{FAIL}(t+\Delta t) = \sum_{k=1}^{s+2} P_{k,FAIL}(t,\Delta t)\, P_k(t) + P_{FAIL}(t)$$

$$= \sum_{k=1}^{s} [1-(r(\Delta t))^k - k(r(\Delta t))^{(k-1)}(1-r(\Delta t))r_d'$$

$$\cdot [r_c' + (1-r_c')\, r_s'\, (1-(1-r')^{(s-k+1)})]]\, P_k(t)$$

$$+ (1-(r(\Delta t))^k - k(r(\Delta t))^{(k-1)}(1-r(\Delta t))r_d' r_c' P_{s+1}(t)$$

$$+ (1-(r_d(\Delta t))(r_c(\Delta t))(r(\Delta t))^{(k-1)} P_{s+2}(t)$$

$$= \sum_{k=1}^{s} P_k(t) + P_{s+1}(t) + P_{FAIL}(t) + P_{s+2}(t)$$

$$- \sum_{k=1}^{s+1} [(r(\Delta t))^k + k(r(\Delta t))^{(k-1)}(1-r(\Delta t))r_d'$$

$$\cdot [r_c' + (1-r_c')\, r_s'\, (1-(1-r')^{(s-k+1)})]]\, P_k(t)$$

$$- [r_d(\Delta t)\, r_c(\Delta t)(r(\Delta t))^{(k-1)}]\, P_{s+2}(t)$$

$$= 1 - \sum_{k=1}^{s+1} [(r(\Delta t))^k + k\,(r(\Delta t))^{(k-1)}(1-r(\Delta t))\, r_d'$$

$$\cdot [r_c' + (1-r_c')\, r_s'\, (1-(1-r')^{(s-k+1)})]] P_k(t)$$

$$- [r_d(\Delta t)\, r_c(\Delta t)(r(\Delta t))^{(k-1)}]\, P_{s+2}(t). \qquad (3\text{-}6)$$

The system reliability may now be computed as the summation of probabilities of being in any state other than the failed state. Then:

$$R(t) = \sum_{i=1}^{s+2} P_i(t) = 1-P_{FAIL}(t) \qquad (3\text{-}7)$$

where the P_i's are obtained from equations (3-3) through (3-6).

It is only necessary then, to compute the probability of the occurrence of state FAIL at any time t to determine the system reliability at that time.

Coverage Equations

System coverage (C) is defined to be the probability that the system will recover given that a failure has occurred [21]. This probability is useful in reliability calculations and provides an indication of the effectiveness of a fault-tolerant system. Hence, a derivation of coverage equations for the basic system will now be shown.

If the system's states are examined, it is evident that a failure in the time period from t to $t + \Delta t$ may be grouped into 1 of 3 classes dependent upon the failure's effect on the system state at time $t + \Delta t$. These classes are

1. The failure causes no change in system state,
2. The failure causes a transition to another system state which is not the failed state, and
3. The failure causes a transition to the failed state.

The occurrence of class 1 and 2 failures contribute to system coverage while the occurrence of class 3 failures does not. Denoting the probability of the occurrence of class L - type failures given that a failure has occurred in the time period from t to $t + \Delta t$ by P(L), then

$$C(t) = P(1) + P(2)$$

But $P(1) + P(2) + P(3) = 1$

so $$C(t) = 1-P(3)$$

$$= 1-P(\text{Class 3 failure/a failure has occurred in t to } t+\Delta t).$$

In general, however, the subevents which constitute the class 3 failure event are dependent on the current system configuration (or state). To overcome this difficulty, the state coverage, $C_i(t)$ (system coverage given that the system state at time t is i) is introduced, where

$$C_i(t) = 1 - P(\text{state i, class 3 failure/a failure has occurred in t to t + } \Delta t \text{ where the system is in state i at time t}),$$

and a state i, class 3 failure is a component failure which causes a transition from state i to the failed state.

Now, by Bayes' Theorem,

$$P(A_j/B) = \frac{P(B/A_j)P(A_j)}{P(B/A_1)P(A_1) + \cdots + P(B/A_n)P(A_n)}$$

where

$$P(A_j, A_r) = \emptyset \text{ for } 1 \leq j, r \leq n$$

and

$$P(A_1 \text{ or } A_2 \text{ or } \ldots \text{ or } A_n) = 1.$$

The following events are considered

A_1: No failure has occurred in t to t + Δt;

A_2: Occurrence of a state i, class 1 failure in t to t + Δt;

A_3: Occurrence of a state i, class 2 failure in t to t + Δt;

A_4: Occurrence of a state i, class 3 failure in t to t + Δt;

B: Occurrence of a failure in t to t + Δt where the system state at time t is i.

Then

$$C_i(t) = 1 - P(A_4/B)$$

$$= 1 - \frac{P(B/A_4)P(A_4)}{P(B/A_1)P(A_1) + P(B/A_2)P(A_2) + P(B/A_3)P(A_3) + P(B/A_4)P(A_4)}$$

But $P(B/A_2) = P(B/A_3) = P(B/A_4) = 1$, and $P(B/A_1) = 0$, so

$$C_i(t) = 1 - \frac{P(A_4)}{P(A_2) + P(A_3) + P(A_4)}$$

$$= 1 - \frac{P(\text{Occurrence of a state i, class 3 failure t to t+}\Delta t)}{P(\text{Occurrence of a failure in t to t+}\Delta t/\text{state at t is i})}.$$

Since each occurrence of a state i, class 3 failure results in a transition from state i to the failed state and no other conditions cause this transition, it follows that

$$P(\text{Occurrence of a state i, class 3 failure in t to t + }\Delta t)$$

$$= P(\text{transition from state i to the failed state in t to t + }\Delta t)$$

$$= P_{i,FAIL}(t,\Delta t).$$

To compute the probability of a failure in the time period from t to t + Δt, a hypothetical series system S, which contains all system components for state i, may be constructed.

If the reliability, $R_s(t)$, of this system is computed, then the failure density function of the system may be obtained as

$$f_s(t) = -\frac{d\,R_s(t)}{dt}.$$

The probability of system S failure in the time period from t to t + Δt is

$$P(\text{Failure of S in t to t} + \Delta t) = \int_{t}^{t+\Delta t} f_s(t)\, dt$$

$$= R_s(t) - R_s(t + \Delta t)$$

as was shown in a previous section.

The reliability of a series system is the product of all system component reliabilities, so

$$P(\text{failure of S in t to t} + \Delta t) = \prod_{j=1}^{n_i} r_j(t) - \prod_{j=1}^{n_i} r_j(t + \Delta t),$$

where n_i is the number of components in S and $r_j(t)$ is the reliability of the j^{th} system component at time t.

Since the failure event for a series system occurs when any system component or combination of components fails and since S contains all components of interest for state i of the original system, then

$$P(\text{Failure of S in t to t} + \Delta t)$$

$$= P(\text{occurrence of a failure in t to t} + \Delta t/\text{state at t is i})$$

$$= \prod_{j=1}^{n_i} r_j(t) - \prod_{j=1}^{n_i} r_j(t + \Delta t),$$

where n_i is the number of components in state i of the original system.

As was shown previously,

$$r_j(t) = 1 \text{ and } r_j(t + \Delta t) = r_j(\Delta t)$$

for a system component j which is required for operation in state i at time t. If the number of these components is m_i, then

P(occurrence of a failure in t to t + Δt/state at t is i)

$$= \prod_{j=1}^{n_i-m_i} r_j(t) - \prod_{q=1}^{m_i} r_q(\Delta t) \prod_{k=1}^{n_i-m_i} r_k(t + \Delta t).$$

For state i ($1 \leq i \leq s+1$) of the basic system, this probability is

$$r_d r_c r_s r^{(s-i+1)} - (r(\Delta t))^k r_d' r_c' r_s' (r')^{(s-i+1)}$$

where all symbols have been previously defined.

Then

$$C_i(t) = 1 - \frac{P_{i,FAIL}(t,\Delta t)}{r_d r_c r_s r^{(s-i+1)} - (r(\Delta t))^k r_d' r_c' r_s' (r')^{(s-i+1)}} \qquad (3\text{-}8)$$

for $i \leq i \leq s + 1$.

For state s+2, $C_{s+2}(t)$ may be obtained as

$$C_{s+2}(t) = 1 - \frac{P_{s+2,FAIL}(t)}{r_s - (r(\Delta t))^{(k-1)} r_d(\Delta t) r_c(\Delta t) r_s'} \qquad (3\text{-}9)$$

Recalling that

$C_i(t)$ = P(system will recover/a failure occurs in t to t + Δt where the system is in state i at time t),

then

P[(System will recover/a failure occurs in t to t + Δt) and the system is non-failed at time t]

$$= \sum_{i=1}^{s+2} C_i(t)\, P_i(t).$$

Since, for a non-repairable system, it is meaningless to compute coverage for the system after it has failed, the total system coverage may be considered to be

$C(t)$ = P[(System will recover/a failure occurs in t to t + Δt)/ the system is non-failed at time t].

This is of the form

$$P[\ A\ /B]$$

whereas the previously derived equation is of the form

$$P[\ A\ \text{and}\ B].$$

Since

$$P[\ A\ /B] = \frac{P[\ A\ \text{and}\ B]}{P(B)},$$

then

$C(t)$ = Total system coverage

$$= \frac{\sum_{i=1}^{s+2} C_i(t)\ P_i(t)}{\sum_{i=1}^{s+2} P_i(t)}$$

$$= \frac{\sum_{i=1}^{s+2} C_i(t)\ P_i(t)}{R(t)}, \qquad (3\text{-}10)$$

where the C_i's are obtained from equations (3-8) and (3-9), the P_i's from equations (3-3) through (3-6) and R from equation (3-7).

Computer Evaluation

Three approaches to computer evaluation of equations of the type presented will be described in this section. These methods are:

1) Manual substitution of transition probability equations into the general state probability equation and evaluation of the state probability equations each Δt,

2) Evaluation of the transition probability equations and substitution of the results into the general state probability equation each Δt, and

3) Evaluation of a product of a $\overline{T}$ - type matrix and a $\overline{T}$ matrix which is updated each Δt.

Methods 1 and 2 are straightforward. Method 3 will now be discussed.

It was shown in a preceding section that

$$\underline{P}(t + \Delta t) = \overline{T}^T \times \underline{P}(t) \tag{3-2}$$

where $\underline{P}(t + \Delta t)$ and $\underline{P}(t)$ are state probability vectors and $\overline{T}$ contains $P_{i,j}(t,\Delta t)$ in its i,j location

Then

$$\underline{P}(t + 2\Delta t) = \overline{T}_1^T \times \underline{P}(t + \Delta t)$$

where $\overline{T}_1$ is $\overline{T}$ evaluated at time $t + \Delta t$. By substitution,

$$\underline{P}(t + 2\Delta t) = \overline{T}_1^T \times [\overline{T}^T \times \underline{P}(t)]$$

$$= [\overline{T}_1^T \times \overline{T}^T] \times \underline{P}(t).$$

In general,

$$\underline{P}(t + n\Delta t) = [\overline{T}_{n-1}^T \times \overline{T}_{n-2}^T \times \ldots \times \overline{T}_1^T \times \overline{T}^T] \, \underline{P}(t)$$

$$= [\overline{T} \times \overline{T}_1 \times \ldots \times \overline{T}_{n-2} \times \overline{T}_{n-1}]^T \, \underline{P}(t)$$

$$= \overline{T}_{n*}^T \, \underline{P}(t)$$

$$\text{where } \overline{T}_{n*} = [\overline{T} \times \overline{T}_1 \times \ldots \times \overline{T}_{n-2} \times \overline{T}_{n-1}]. \tag{3-11}$$

Thus, to evaluate $P(t + n\Delta t)$ when $P(t)$ is known, the following algorithm may be used.

1. Evaluate $\overline{T}$ at time t, set $\overline{T}_{i*} = \overline{T}$, i = 1.
2. Evaluate $\overline{T}$ at time t + i Δ t to obtain $\overline{T}_i$.
3. $\overline{T}_{i+1*} = \overline{T}_{i*} \times \overline{T}_i$.
4. If i < n then i = i + 1, go to 2. Otherwise, $\underline{P}(t + n\Delta t) = \overline{T}_{n*}^{T} \times \underline{P}(t)$, stop.

For a system with a small number of states and state transitions, method 1 is managable. For systems with a large number of states, however, either method 2 or 3 is more expedient. Example flowcharts for methods 2 and 3 are presented in Appendix A. Program listings may be found in [27].

The selection of a suitable Δt for use in the computer evaluation of these equations is a difficult task. This problem will now be discussed.

The time period Δt was originally defined to be a time period in which no more than one state transition is likely to occur. Since the probability of more than one state transition occurring may be represented as a product of state transition probabilities, the monitoring of these products during execution will give an indication of the appropriateness of the selected Δt.

By specifying a maximum allowable probability, p_{max}, for the occurrence of two state transitions in time Δt, and reducing Δt when this probability is exceeded, the computational error may be reduced. The following algorithm will implement this self-monitoring control for a method 3-type evaluation.

1. Evaluate $\overline{T}$ at time t, set $\overline{T}_{i*} = \overline{T}$, i = 1.

1a. Specify initial Δt, p_{max}

2. Evaluate $\overline{T}$ at time t + iΔt to obtain $\overline{T}_i$.

2a. For each non-diagonal entry $\overline{T}i(j,k)$ compute $\overline{T}i(j,k)\cdot(\overline{T}_i(k,m)$ for each m.

2b. If any of these products is greater than p_{max}, reduce Δt and go to 2.

3. $\overline{T}_{i+1*} = \overline{T}_{i*} \times \overline{T}_i$.

4. If i < n then i = i + 1, go to 2. Otherwise $\underline{P}(t + n\Delta t) = T_{n*}^{T} \times P(t)$, stop.

In general, the value selected for p_{max} is dependent on the subsystem failure rates and the computational accuracy of the computing system used. For the computations of this paper, satisfactory results were obtained by the use of p_{max} in the range from .0001 to .000001.

The magnitude of the computational error accumulated at time t may be approximated by determining the magnitude of the difference of the sum of all state probabilities and 1. In equational form,

$$|e(t)| = |1 - \sum_{i=1}^{N} P_i(t)|$$

where N is the number of system states.

The percent error in system reliability may be approximated by

$$e(t)\% = \frac{|e(t)|}{R(t)} \times 100\%.$$

IV. RELIABILITY EQUATIONS FOR ALTERNATE SYSTEMS

This chapter will show equational developments for the reliability of the non-spared, TMR, duplicated and double-error-correcting systems. A method will also be shown which allows the computation of the probability of various memory word fault patterns and the effects of these patterns on system reliability.

Non-Spared System

The non-spared system is capable of operation in only 3 states. These states correspond to states 1, s+2, and FAIL in the basic system. By substitution of 0 for s in the equations for the basic system, the state probability equations for states 1, 2, and FAIL of the non-spared system are obtained as follows:

$$P_{NS_1}(t + \Delta t) = (r(\Delta t))^k P_{NS_1}(t) \tag{4-1}$$

$$P_{NS_2}(t + \Delta t) = k(r(\Delta t))^{(k-1)}(1-r(\Delta t))\, r_d' r_c' P_{NS_1}(t) + (r_d(\Delta t))(r_c(\Delta t))(r(\Delta t))^{(k-1)} P_{NS_2}(t) \tag{4-2}$$

$$P_{NS_{FAIL}}(t + \Delta t) = 1-[(r(\Delta t))^k + k(r(\Delta t))^{(k-1)}(1-r(\Delta t))r_d' r_c'] \cdot P_{NS_1}(t) - [r_d(\Delta t) r_c(\Delta t)(r(\Delta t))^{(k-1)}]P_{NS_2}(t) \tag{4-3}$$

and

$$R_{NS}(t) = P_{NS_1}(t) + P_{NS_2}(t) = 1 - P_{NS_{FAIL}}(t) \qquad (4\text{-}4)$$

where the P_{NS_i}'s are obtained from equations (4-1) through (4-3)

TMR System

The reliability of the TMR system may be approximated from the reliability of the non-spared system by application of the classical TMR equation. From [24], this equation is

$$R_{TMR}(t) = [3(R(t))^2 - 2(R(t))^3]\, r_{VT}(t),$$

where R(t) is the unreplicated unit reliability, and $r_{VT}(t)$ is the reliability of the voting and codeword testing circuitry.

Then

$$R_{TMR}(t) = [3(R_{NS}(t))^2 - 2(R_{NS}(t))^3]\, r_{VT}(t), \qquad (4\text{-}5)$$

where $R_{NS}(t)$ is obtained by use of equation (4-4)

Duplicated System

The reliability of the duplicated system may be computed by determining the probability of the various operational modes of the system. These modes are:

1. Both non-spared units operate correctly,
2. the unit currently on-line fails, and the sense switching circuitry switches the system output to the other unit which is non-failed, and
3. the unit currently off-line fails.

The reliability of this system, then, is

$$R_D(t) = [R_{NS}(t)]^2 + [1 - R_{NS}(t)]\, R_{NS}(t)\, r_{ss}(t) + R_{NS}(t)\, [1 - R_{NS}(t)]$$

$$= R_{NS}(t) + [1 - R_{NS}(t)]\, R_{NS}(ts)\, r_{ss}(t), \qquad (4\text{-}6)$$

where $r_{ss}(t)$ is the reliability of the sense-switching circuitry and R_{NS} is obtained by use of equation (4-4).

Double-Error-Correcting (DEC) System

Carter and McCarthy [20] have described a fault-tolerant memory system of the double-error-correcting type which utilizes a software implementable double-error-correction algorithm. The algorithm is based on a concept of memory word error modeling which will now be described.

The non-operational modes of a memory word bit cell are assumed to be:

1) Stuck-at-one (s-a-1), and
2) Stuck-at-zero (s-a-Ø).

The occurrence of either of these modes is termed a fault.

The class of all faults may be partitioned into two subclasses by the effect of each fault on the correct memory word bit. If the fault is of the s-a-x type and the correct memory word bit for that location is x, then no effect on the memory bit occurs. Faults of this subclass are termed failures. If the fault is s-a-x and the correct bit is $\overline{x}$, then the fault causes an incorrect response on a memory read operation. Faults of this type are called errors.

The weight of a binary word is defined to be the number of binary digits (bits) in the word which are logic 1. By analysis of

the words of a particular code the sum of all codeword weights, W, may be obtained. An average codeword weight, $\overline{w}$ is computed by

$$\overline{w} = \frac{W}{V}$$

where V is the total number of codewords. If $\overline{w}$ is divided by N, the length in bits of each codeword, an approximation to the statistical probability of any given bit of a word being a logic 1 is obtained. In equational form,

$$P(\text{Word bit} = 1) = P_{w1} = \frac{\overline{w}}{N}, \text{ and}$$

$$P(\text{Word bit} = \emptyset) = P_{w\emptyset} \cong 1 - P_{w1} = 1 - \frac{\overline{w}}{N}.$$

A statistical analysis of faults for a memory system should isolate the following probabilities for the bit locations of a data word.

$$P(\text{Bit location s-a-1/location faulted}) = P_{S1}, \text{ and}$$

$$P(\text{Bit location s-a-0/location faulted}) = P_{S\emptyset}.$$

It is now possible to obtain the probability of a failure when it is known that a single word fault has occurred. This probability is

$$P(\text{failure/1 fault}) = P[(\text{BIT location s-a-1/location faulted}) \text{ and Word Bit} = 1] + P[(\text{Bit location s-a-}\emptyset\text{/location faulted}) \text{ and word bit} = \emptyset]$$

$$= P_{S1}\, P_{w1} + P_{S\emptyset}\, P_{w\emptyset}.$$

In a similar manner,

$$P(\text{Error/1 fault}) = P_{S1}\,P_{w\emptyset} + P_{S\emptyset}\,P_{w1}.$$

Since P(failure/1 fault) + P(error/1 fault)

$$= P_{S1}\,P_{w1} + P_{S\emptyset}\,P_{w\emptyset} + P_{S\emptyset}\,P_{w1} + P_{S1}\,P_{w\emptyset}$$

$$= (P_{S1} + P_{S\emptyset})(P_{w1} + P_{w\emptyset})$$

$$= 1 \cdot 1 = 1,$$

the binomial probability distribution may be used to compute the probability of any combination of errors and failures in a word given that a certain number of faults has occurred.
Then

$$P(\text{n failures and m errors/n + m faults})$$

$$= \binom{n+m}{n}(P_{S1}P_{w1} + P_{S\emptyset}P_{w\emptyset})^{n}(P_{S1}P_{w\emptyset} + P_{S\emptyset}P_{w1})^{m}.$$

If the binomial distribution is also used to compute the probability of n+m faults, then

$$P(\text{n failures and m errors, n + m faults in b bits})$$

$$= \binom{n+m}{n}(P_{S1}P_{w1}+P_{S\emptyset}P_{w\emptyset})^{n}(P_{S1}P_{w\emptyset}+P_{S\emptyset}P_{w1})^{m}\cdot\binom{b}{n+m}r^{b-(n+m)}(1-r)^{(n+m)},$$

where r is the reliability of a memory word bit location.

Since $\binom{b}{n+m}$ is the number of n+m-fault words which may occur and $\binom{n+m}{n}$ is the number of ways that exactly n failures may be ordered among n+m faults, then the number of distinct m+n-fault words with n failures is $\binom{n+m}{n}\binom{b}{n+m}$. The total number of distinct (with regard to number and order of failures) n+m-fault words is then

$$\left[\sum_{i=0}^{n+m}\binom{n+m}{i}\right]\binom{b}{n+m}.$$

These numbers may now be used to obtain the percentage of f-fault words which contain a given number of failures. For example, the percentage of f-fault words of b bits which contain f failures is

$$\frac{\binom{f}{f}\binom{b}{f}}{\left[\sum_{i=0}^{f}\binom{f}{i}\right]\binom{b}{f}} \times 100\% = \frac{1}{\left[\sum_{i=0}^{f}\binom{f}{i}\right]} \times 100\%,$$

a useful figure, since an f-fault word with f failures is error-free. The application of these concepts to the double-error-correcting system will be shown following a discussion of correctable error types for the system.

A fault pattern vector for a memory word is defined as

FPV = (he jf, qe nf)

where h and q are the numbers of errors in the memory word data and check bits, respectively, and j and n are the numbers of failures.

The double-error-correction algorithm discussed will always produce a valid correction when presented with memory words with FPV's of certain forms. These forms, from [20], are as follows.

(2e Øf, Øe Øf); (1e Øf, 1e Øf); (Øe Øf, 2e Øf);
(2e 1f, Øe Øf); (2e Øf, Øe 1f); (Øe Øf, 2e 1f);
(2e 1f, Øe 1f); (2e Øf, Øe 2f); (Øe Øf, 2e 2f);
(Øe Øf, 4e Øf).

For memory words with FPV's of the following forms correction may or may not be attempted and results may be invalid [20]

(1e Øf, 1e 1f); (Øe 1f, 2e Øf);
(1e Øf, 1e 2f); (Øe 1f, 2e 1f).

No error correction is attempted in the following cases [20]

(1e 1f, 1e Øf);
(2e 2f, Øe Øf); (1e 2f, 1e Øf); (1e 1f, 1e 1f);
(4e Øf, Øe Øf); (3e Øf, 1e Øf); (2e Øf, 2e Øf).

It should be noted that the preceding FPV's listed all contain an even number of errors and will produce error syndrome vectors of even weight. The computation of a syndrome of this type by the memory translator causes the invocation of this algorithm.

A second algorithm has been designed to attempt data reconstruction when an odd-weight error syndrome is computed. Since many triple-error patterns produce a single-error syndrome and a high percentage of these syndromes imply an error in a valid bit, a critical function of this algorithm is to distinguish between single and triple word errors.

This algorithm is capable of reconstructing all memory words with FPV's containing exactly one error and two or fewer failures. In addition, all memory words with FPV's containing one error and three failures are corrected with the exception of the FPV

(Øe 3f, 1e Øf)

for which no reconstruction is attempted [20].

Valid results, [20], are also produced for

(Øe Øf, 3e Øf) and (Øe Øf, 3e 1f).

Correction results are variable [20] for memory words with the following FPV's

(2e Øf, 1e Øf); (1e Øf, 2e Øf);
(2e Øf, 1e 1f); (1e Øf, 2e 1f).

No correction is attempted, [20], for the case listed above and the cases

(3e Øf, Øe Øf); (3e Øf, Øe 1f).

The listings above show that any combination of two or fewer faults in a memory word will be algorithmically corrected. For words with three faults, the percentage of words which are corrected may be computed as follows.

The number of ways in which three faults may appear in a word with k bits is

(The number of ways 3 faults can appear) +
(The number of ways 2 faults and 1 error can appear) +
(The number of ways 1 fault and 2 errors can appear) +
(The number of ways 3 errors can appear)

$$= \left[\binom{k}{3} + 3\binom{k}{3} + 3\binom{k}{3} + \binom{k}{3}\right] = 8 \quad \binom{k}{3}$$

The first term of this sum represents all 3-fault words with no errors. No correction is required for these words. In addition, the triple

error algorithm will correct all $3\binom{k}{3}$ three-fault words with only one error.

A 3-fault word containing 2 errors will not be corrected if the FPV is of the form

(1e 1f, 1e Øf).

If the number of data bits in the word is D and the number of check bits is C, then the number of 3-fault patterns of this form is

$$\binom{D}{2}\binom{2}{1}\binom{C}{1} = 2\binom{D}{2}C.$$

The number of 3-fault words with 2 errors for which correction is uncertain is

$$\binom{D}{1}\binom{2}{1}\binom{C}{2} + \binom{D}{1}\binom{C}{2} = 2\,D\binom{C}{2} + D\binom{C}{2} = 3\,D\binom{C}{2}.$$

A 3-fault word with three errors will be corrected if the FPV is of the form

(Øe Øf, 3e Øf).

The number of patterns of this form is

$$\binom{C}{3}.$$

The number of 3-fault words with three errors for which correction is uncertain is

$$\left[\binom{D}{2}\binom{C}{1} + \binom{D}{1}\binom{C}{2}\right] = \left[C\binom{D}{2} + D\binom{C}{2}\right].$$

The total number, T_3, of 3-fault words which are correctable is then bounded as follows

$$[\binom{k}{3} + 3\binom{k}{3} + 3\binom{k}{3} - 2C\binom{D}{2} + \binom{C}{3}] \leq T_3 \leq$$

$$[\binom{k}{3} + 3\binom{k}{3} + 3\binom{k}{3} - 2C\binom{D}{2} + \binom{C}{3} + 3D\binom{C}{2} + C\binom{D}{2} + D\binom{C}{2}]$$

$$\equiv [7\binom{k}{3} - 2C\binom{D}{2} + \binom{C}{3}] \leq T_3 \leq [7\binom{k}{3} - C\binom{D}{2} + \binom{C}{3} + 4D\binom{C}{2}].$$

Since there are $8\binom{k}{3}$ possible ways that 3 faults can occur, the percentage, u_3, of 3 faults words that can be corrected is

$$u_3 = \frac{T_3}{8\binom{k}{3}} \times 100\%.$$

For the (22, 16) code of the basic system, u_3 may be computed as:

$$75.96\% \leq u_3 \leq 89.45\%$$

A breakdown of double-error-correcting system correction percentages by the number of memory word faults is shown in Table 3. In this table, $u_{m,n}$ denotes n errors which are system correctable. u_m denotes the total percentage of m-fault FPV's which are correctable.

The switching strategy assumed for the double-error correcting system is as follows:

1) If a memory word is detected to have a single error, the single error correction procedure is performed.

2) If the word has two errors, one of the faulty on-line bit planes is switched out and replaced with a spare. Error correction is attempted by use of the double-error correction procedure.

TABLE 3. Percentage of Memory Word FPV's Correctable for the Double-Error-Correcting System (22, 16) Code

F # FAULTS	e # ERRORS	$u_{F,e}$ (% correctable/100%) x(% of F-fault words with e errors/100%)
0	0	$u_{0,0} = 1$
0	0	$u_0 = 1$
1	0	$u_{1,0} = .5$
1	1	$u_{1,1} = .5$
1	0,1	$u_1 = 1$
2	0	$u_{2,0} = .25$
2	1	$u_{2,1} = .5$
2	2	$u_{2,2} = .25$
2	0,1,2	$u_2 = 1$
3	0	$u_{3,0} = .125$
3	1	$u_{3,1} = .375$
3	2	$.258 \leq u_{3,2} \leq .317$
3	3	$.0016 \leq u_{3,3} \leq .078$
3	0,1,2,3	$.7596 \leq u_3 \leq .8945$

Table 3 (continued)

F # FAULTS	e # ERRORS	$u_{F,e}$ (% correctable/100%) x(% of F-fault words with e errors/100%)
4	0	$u_{4,0} = .0625$
4	1	$u_{4,1} = .25$
4	2	$.102 \leq u_{4,2} \leq .119$
4	3	$.0005 \leq u_{4,3} \leq .0395$
4	4	$u_{4,4} = .0001$
4	0,1,2,3,4	$.4151 \leq u_4 \leq .4711$

3) If the word has either three or four errors, a correction is attempted. If the correction is successful, faulty on-line bit planes are replaced with spare bit planes until either all available spares are exhausted or only one faulty bit plane remains on-line.

A $\overline{T}$ matrix may be constructed as shown in Figure 6 with the system configuration in each state as shown in Table 4.

Appendix B shows the derivation of the state transition probability equations for this system. If the notational simplifications

$$(u_{k-x+y,k-x+y})\binom{x}{y}(r(\Delta t)^{(x-y)}(1-r(\Delta t))^{y} = D(x,y),$$

$$\sum_{k=0}^{y} \binom{s-x+2}{k} (1-r')^{(s-x+2-k)}(r')^{k} = E(x,y), \text{ and } r(\Delta t) = r^*$$

are made and the reliability of the algorithmic correction procedure is denoted by r_A, then the transition equations appear as follows:

$$P_{1,2}(t,\Delta t) = D(k,1)\ r_d' r_c' r_A'.$$

$$P_{1,3}(t,\Delta t) = D(k,2)\ r_d' r_c' r_A' r_s' (1-E(2,0)).$$

$$P_{1,4}(t,\Delta t) = D(k,3)\ r_d' r_c' r_A' r_s' (1-E(2,1)).$$

$$P_{1,5}(t,\Delta t) = D(k,4)\ r_d' r_c' r_A' r_s' (1-E(2,2)).$$

$$P_{1,s+3}(t,\Delta t) = D(k,2)\ r_d' r_c' r_A' (1-r_s' + r_s' E(2,0)) + D(k,3)\ r_d' r_c' r_A' r_s' (E(2,1)-E(2,0)) + D(k,4)\ r_d' r_c' r_A' r_s' (E(2,2)-E(2,1)).$$

$$P_{1,s+4}(t,\Delta t) = D(k,3)\ r_d' r_c' r_A' (1-r_s' + r_s' E(2,0)) + D(k,4)\ r_d' r_c' r_A' r_s' (E(2,1)-E(2,0)).$$

$$P_{1,s+5}(t,\Delta t) = D(k,4)\ r_d' r_c' r_A' (1-r_s' + r_s' E(2,0)).$$

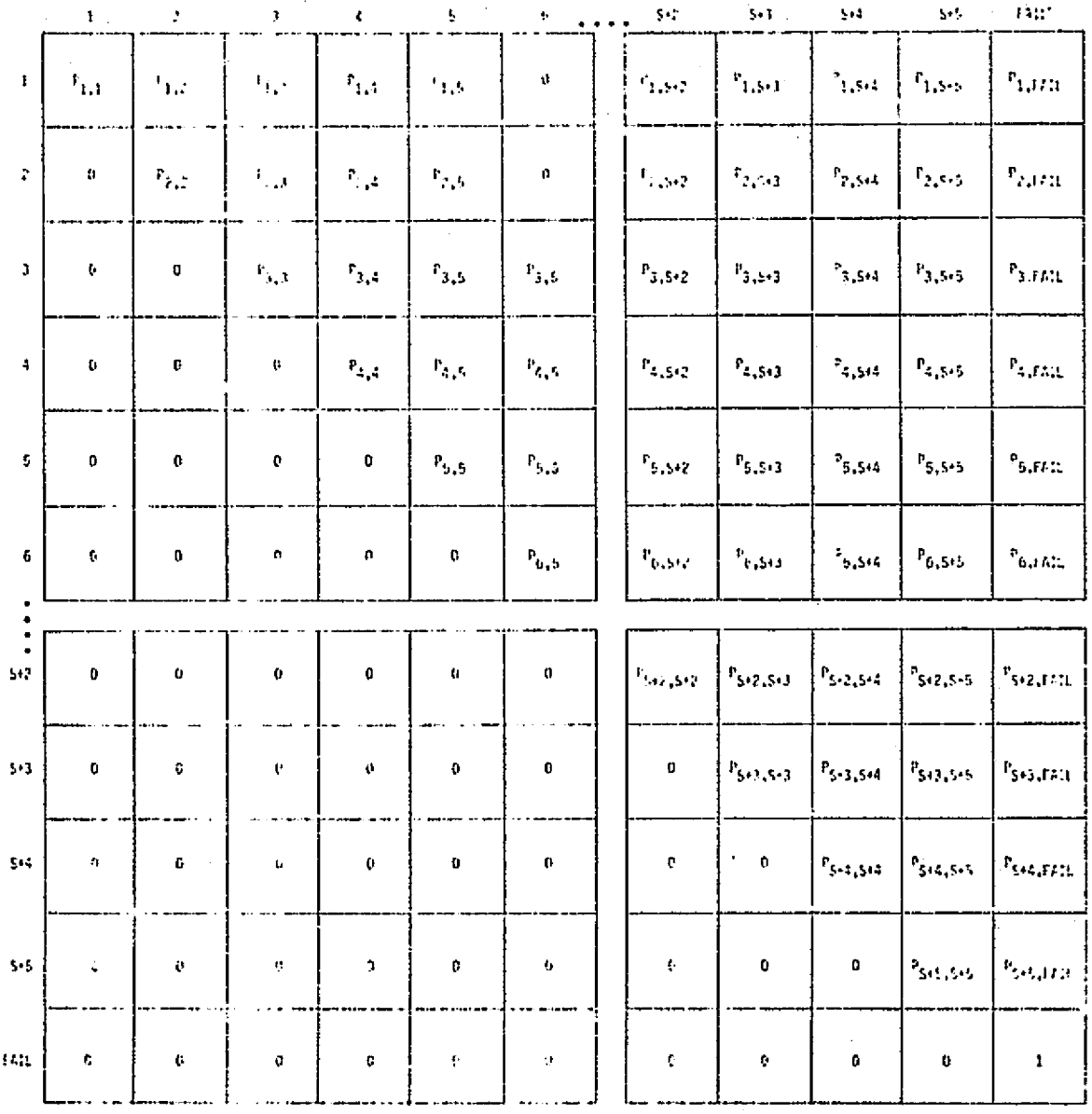


FIGURE 6. $\overline{T}$-Matrix for Double-Error-Correcting System

TABLE 4. State Configurations For Double-Error-Correcting System

State	Configuration
1	K Good bit planes on-line, S available spares
$2 \leq i \leq s + 2$	K-1 Good bit planes on-line s - i + 2 available spares
s + 3	K-2 Good bit planes on-line
s + 4	K-3 Good bit planes on-line
s + 5	K-4 Good bit planes on-line
FAIL	An uncorrectable word error exists

$$P_{1,1}(t,\Delta t) = D(k,0).$$

$$P_{i,i+1}(t,\Delta t) = D(k-1,1)\ r_d{*}r_c{*}r_A{*}r_s{'}\ (1-E(i,0))$$

for $2 \le i \le s+1$.

$$P_{i,i+2}(t,\Delta t) = D(k-1,2)\ r_d{*}r_c{*}r_A{*}r_s{'}\ (1-E(i,1))$$

for $2 \le i \le s$.

$$P_{i,i+3}(t,\Delta t) = D(k-1,3)\ r_d{*}r_c{*}r_A{*}r_s{'}\ (1-E(i,2))$$

for $2 \le i \le s-1$.

$$P_{i,s+3}(t,\Delta t) = D(k-1,1)\ r_d{*}r_c{*}r_A{*}(1-r_s{'} + r_s{'}\ E(i,0))$$
$$+ D(k-1,2)\ r_d{*}r_c{*}r_A{*}r_s{'}\ (E(i,1)-E(i,0))$$
$$+ D(k-1,3)\ r_d{*}r_c{*}r_A{*}r_s{'}\ (E(i,2)-E(i,1))$$

for $2 \le i \le s$.

$$P_{i,s+4}(t,\Delta t) = D(k-1,2)\ r_d{*}r_c{*}r_A{*}\ (1-r_s{'} + r_s{'}\ E(i,0))$$
$$+ D(k-1,3)\ r_d{*}r_c{*}r_A{*}r_s{'}\ (E(i,1)-E(i,0))$$

for $2 \le i \le s+1$.

$$P_{i,s+5}(t,\Delta t) = D(k-1,3)\ r_d{*}r_c{*}r_A{*}\ (1-r_s{'} + r_s{'}\ E(1,0))$$

for $2 \le i \le s+1$.

$$P_{i,i}(t,\Delta t) = D(k-1,0)\ r_d{*}r_c{*}r_A{*}$$

for $2 \le i \le s+2$.

$$P_{s+1,s+3}(t,\Delta t) = D(k-1,1)\ r_d{*}r_c{*}r_A{*}\ (1-r_s{'} + r_s{'}\ E(s+1,0))$$
$$+ D(k-1,2)\ r_d{*}r_c{*}r_A{*}r_s{'}\ (E(s+1,1)-E(s+1,0))$$

$$P_{s+2,s+j}(t,\Delta t) = D(k-1,j-2)\ r_d{}^*r_c{}^*r_A{}^*$$

for $3 \leq j \leq 5$.

$$P_{s+3,s+j}(t,\Delta t) = D(k-2,j-3)\ r_d{}^*r_c{}^*r_A{}^*$$

for $4 \leq j \leq 5$.

$$P_{s+4,s+5}(t,\Delta t) = D(k-3,1)r_d{}^*r_c{}^*r_A{}^*.$$

$$P_{s+j,s+j}(t,\Delta t) = D(k-j+1,0)\ r_d{}^*r_c{}^*r_A{}^*$$

for $3 \leq j \leq 5$.

$$P_{1,FAIL}(t,\Delta t) = 1-D(k,0)-r_d\,r_c\,r_A\sum_{j=1}^{4} D(k,j).$$

$$P_{i,FAIL}(t,\Delta t) = 1-r_d{}^*r_c{}^*r_A{}^*\sum_{j=0}^{3} D(k-1,j)$$

for $2 \leq i \leq s+2$.

$$P_{s+q,FAIL}(t,\Delta t) = 1-r_d{}^*r_c{}^*r_A{}^*\sum_{j=0}^{5-q} D(k-q+1,j).$$

for $3 \leq q \leq 5$.

The state probability equations for this system are also derived in Appendix B. The resultant equations are

$$P_1(t+\Delta t) = D(k,0)\ P_1(t). \tag{4-7}$$

$$P_2(t+\Delta t) = D(k,1)\ r_d\,r_c\,r_A\,P_1(t) + D(k-1,0)\ r_d{}^*r_c{}^*r_A{}^*P_2(t). \tag{4-8}$$

$$\begin{aligned} P_3(t+\Delta t) = {} & D(k,2)\ r_d\,r_c\,r_A\,r_s\,(1-E(2,0))\ P_1(t) \\ & + r_d{}^*r_c{}^*r_A{}^*\,[D(k-1,1)r_s\,(1-E(2,0))P_2(t) \\ & + D(k-1,0)\ P_3(t)]. \end{aligned} \tag{4-9}$$

$$P_4(t+\Delta t) = D(k,3)\, r_d' r_c' r_A' r_s' (1-E(2,1))\, P_1(t)$$
$$+ r_d^* r_c^* r_A^* \,[D(k-1,2)\, r_s' \,(1-E(2,1))\, P_2(t) \qquad (4\text{-}10)$$
$$+ D(k-1,1)\, r_s' \,(1-E(3,0))\, P_3(t) + D(k-1,0)\, P_4(t)].$$

$$P_5(t+\Delta t) = D(k,4)\, r_d' r_c' r_A' r_s' \,(1-E(2,2))\, P_1(t)$$
$$+ r_d^* r_c^* r_A^* \,[D(k-1,3)\, r_s' \,(1-E(2,2))\, P_2(t)$$
$$+ D(k-1,2)\, r_s' \,(1-E(3,1))\, P_3(t) \qquad (4\text{-}11)$$
$$+ D(k-1,1)\, r_s' \,(1-E(4,0))\, P_4(t) + D(k-1,0)\, P_5(t)].$$

$$P_i(t+\Delta t) = r_d^* r_c^* r_A^* \,[D(k-1,0)\, P_i(t)$$
$$+ r_s' \sum_{j=1}^{3} D(k-1,j)(1-E(i-j,j-1)\, P_{i-j}(t)] \qquad (4\text{-}12)$$

for $6 \le i \le s+2$.

$$P_{s+3}(t+\Delta t) = r_d' r_c' r_A' [D(k,2)(1-r_s' + r_s' E(2,0)$$
$$+ \sum_{j=3}^{4} D(k,j)\, r_s' \,(E(2,j-2)-E(2,j-3))]\, P_1(t)$$
$$+ r_d^* r_c^* r_A^* \sum_{k=2}^{s} [D(k-1,1)(1-r_s' + r_s' E(k,0))$$
$$+ D(k-1,2) r_s' \,(E(k,1)-E(k,0))$$
$$+ D(k-1,3)\, r_s' \,(E(k,2)-E(k,1))]\, P_k(t)$$
$$+ [D(k-1,1)(1-r_s' + r_s' E(s+1,0))$$
$$+ D(k-1,2)\, r_s' \,(E(s+1,1)-E(s+1,0))]\, P_{s+1}(t)$$
$$+ D(k-1,1)\, P_{s+2}(t) + D(k-2,0)\, P_{s+3}(t). \qquad (4\text{-}13)$$

$$P_{s+4}(t+\Delta t) = r_d' r_c' r_A' [D(k,3)(1-r_s' + r_s' E(2,0)) + D(k,4)\, r_s' (E(2,1) - E(2,0))]\, P_1(t) + r_d^* r_c^* r_A^* \sum_{j=2}^{s+1} [D(k-1,2)(1-r_s' + r_s' E(j,0)) + D(k-1,3)\, r_s' (E(j,1)-E(j,0))]\, P_j(t) + D(k-1,2)\, P_{s+2}(t) + D(k-2,1)\, P_{s+3}(t) + D(k-3,0)\, P_{s+4}(t). \quad (4\text{-}14)$$

$$P_{s+5}(t+\Delta t) = D(k,4)\, r_d' r_c' r_A' (1-r_s' + r_s' E(2,0))\, P_1(t) + r_d^* r_c^* r_A^* \sum_{j=2}^{s+1} D(k-1,3)(1-r_s' + r_s' E(j,0)) P_j(t) + D(k-1,3)\, P_{s+2}(t) + D(k-2,2)\, P_{s+3}(t) + D(k-3,1)\, P_{s+4}(t). \quad (4\text{-}15)$$

$$P_{FAIL}(t+\Delta t) = 1 - D(k,0) + r_d' r_c' r_A' \sum_{j=1}^{4} D(k,j)\, P_1(t) + r_d^* r_c^* r_A^* \left[\sum_{k=2}^{s+2} \left(\sum_{m=0}^{3} D(k-1,j)\, P_k(t)\right) + \sum_{n=3}^{5} \left(\sum_{q=0}^{5-n} D(k-n+1,q)\, P_{s+n}(t)\right)\right] . \quad (4\text{-}16)$$

It should be noted that these equations are developed for a double-error-correcting system with s+2 greater than 5 (equivalently, more than 3 spare bit planes). If s+2 = i where $2 \leq i \leq 5$, then the equations involving state j where $i \leq j \leq 5$ should be modified to delete this state. This modification will involve only the deletion of the appropriate equations.

If the system is defined to be operating satisfactorily in states 1 through s+5 and $P_1(t=0) = 1$, $P_i(t=0)=0$, $i \neq 1$, then the system reliability may be completely specified as

$$R_{DEC}(t) = \sum_{i=1}^{s+5} P_i(t), \tag{4-17}$$

where the P_i's are obtained from equations (4-7) through (4-16).

V. ANALYSIS RESULTS

In this chapter, typical results of analyses performed on the five systems previously described will be discussed. Comparative reliabilities of each system are shown and the effect of varying several system parameters is described.

The base variable values assumed [22] for the system analyses are shown in Table 5. For each analysis performed, the system variables are fixed at the base value unless otherwise noted.

A comparative reliability analysis of the five subject systems was performed by use of equations (3-7), (4-4), (4-5), (4-6), and (4-17). The results of this analysis are displayed in Figure 7. This figure shows the reliability of the TMR, non-spared, duplicated, basic, and double-error-correcting systems for mission lengths of four years or less. Also shown is the reliability of a simplex system with no error-detecting or correcting capabilities. This system consists of 16 on-line bit planes and has reliability $(e^{-\lambda_{BP}t})^{16}$ where λ_{BP} is obtained from Table 5. It may be seen from this figure that, for missions of 1/2 year or less, all of the systems except the non-spared and simplex systems have reliability greater than .99. For greater mission lengths, however, the reliability of the non-spared, duplicated, and TMR systems decrease rapidly. For a 3-year mission, probably only the basic or double-error-correcting systems would be acceptable.

TABLE 5. Base Values for System Variables

# On-line Bit Planes	22
# Spare Bit Planes	4
Bit Plane Failure Rate	$2.6384/10^6$ HR
Detector Failure Rate	$.900/10^6$ HR
Reconfiguration Switch Failure Rate	$.583/10^6$ HR
Corrector Failure Rate	$.027/10^6$ Hr
DEC Algorithm Failure Rate	0
Mission Length	3 Years
Memory Size	16k Words
Failure Distribution	Exponential
4K-Bit Subplane Failure Rate	$.5596/10^6$ Hr
Peripheral Bit Plane Circuitry Failure Rate	$.3/10^6$ Hr

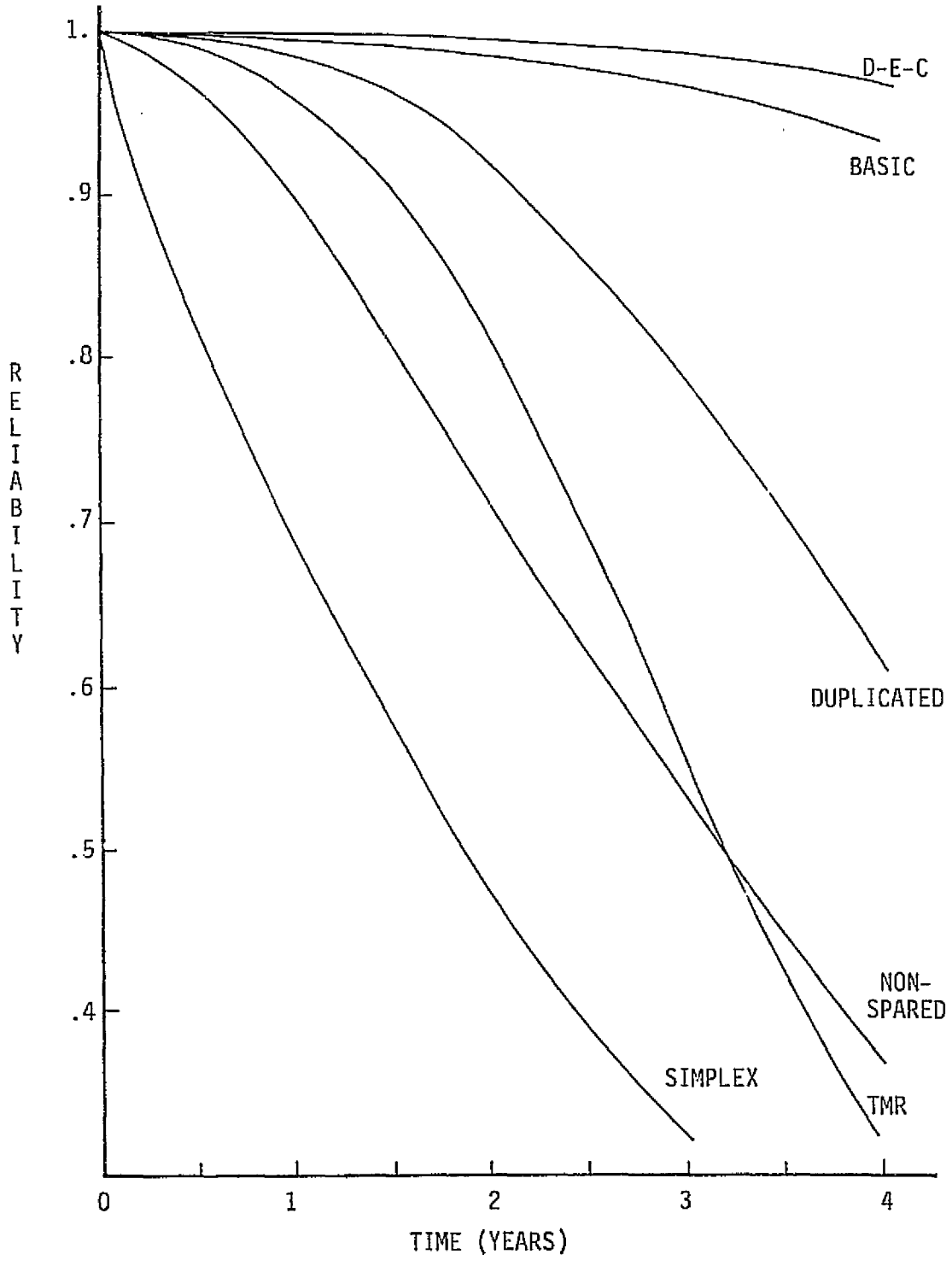


FIGURE 7. Reliability of Subject Systems

Comparison of the curves for the double-error-correcting and Basic Systems shows the reliability improvement to be expected from the use of the software algorithms of the double-error-correcting system. For 1/2-year missions, this improvement is negligible. For missions of greater lengths, however, the reliability improvement gained by the use of this system becomes important.

It is interesting to note that, while the duplicated and TMR systems represent a doubling and tripling of memory bit planes over the non-spared system, the basic and double-error-correcting systems result in much higher system reliabilities with an addition of only 4 bit planes to the non-spared system.

Figure 8 shows the results of a reliability analysis performed on the basic system for various numbers of spare bit planes. The corresponding curves for the double-error-correcting system are shown in Figure 9. Comparison of these two figures shows that the same degree of reliability achieved by the basic system with 4 spare bit planes may be reached by a double-error-correcting system with 3 spares and a sufficiently reliable double-error-correction algorithm. The need for one spare bit plane may thus be aleviated by the use of software error correction.

The reliability of the software error correction algorithms used in the double-error-correcting system is highly important to system success. The effects on the double-error-correcting system reliability made by varying a hypothetical failure rate for the CPU hardware which implements these algorithms is shown in Figure 10.

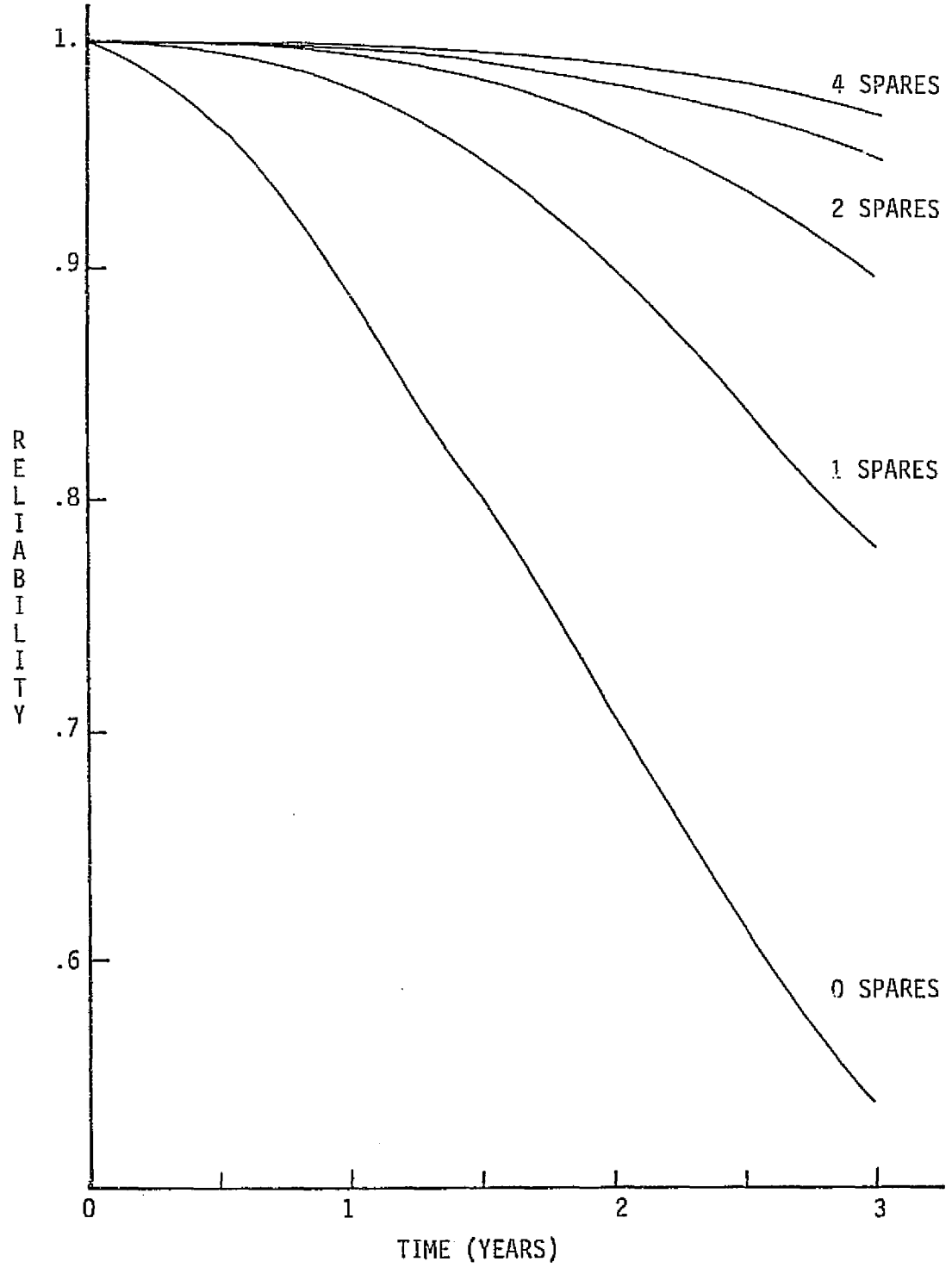


FIGURE 8. Reliability of Basic System
for Various Numbers of Spares

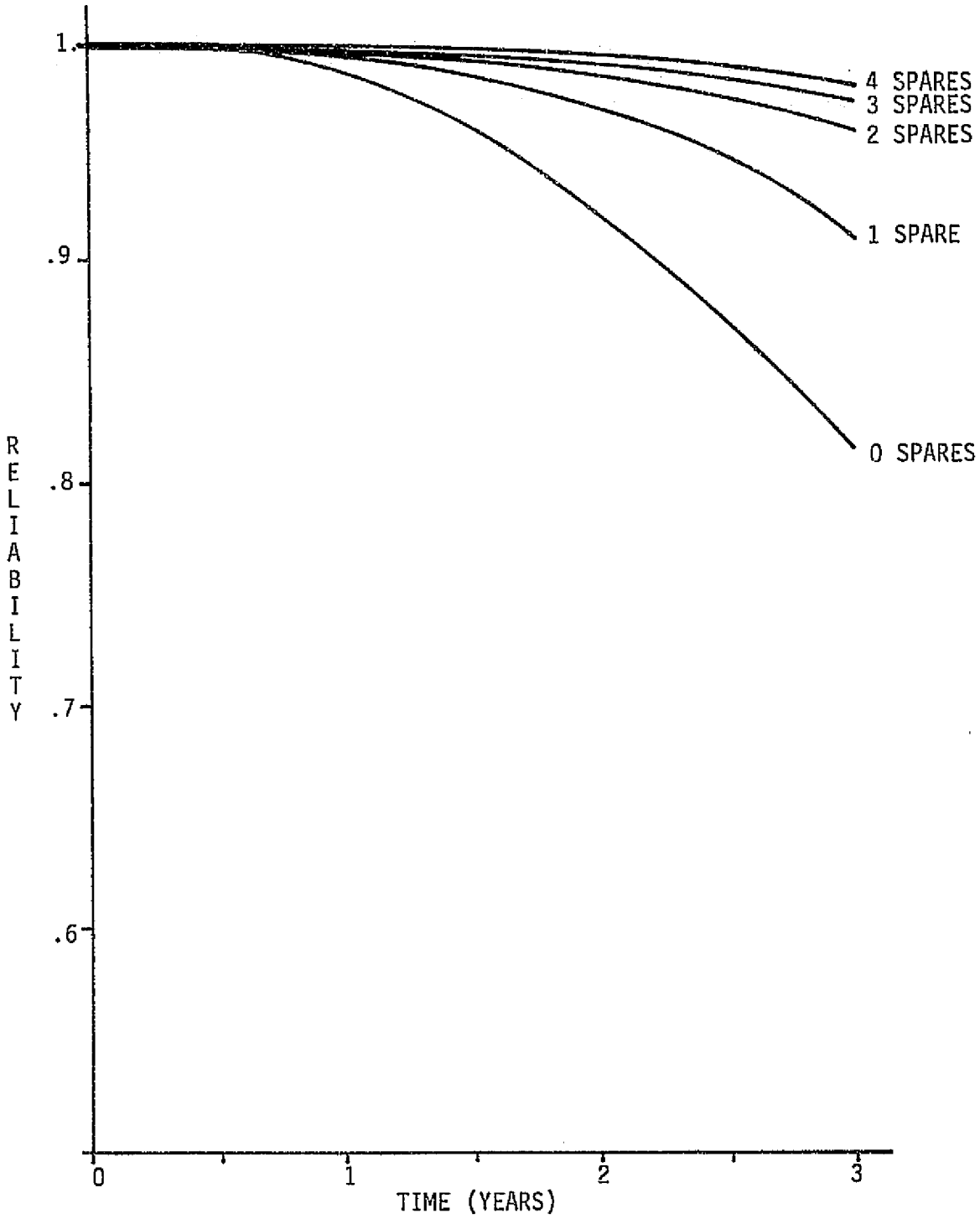


FIGURE 9. Reliability of Double-Error-Correcting System for Various Numbers of Spares

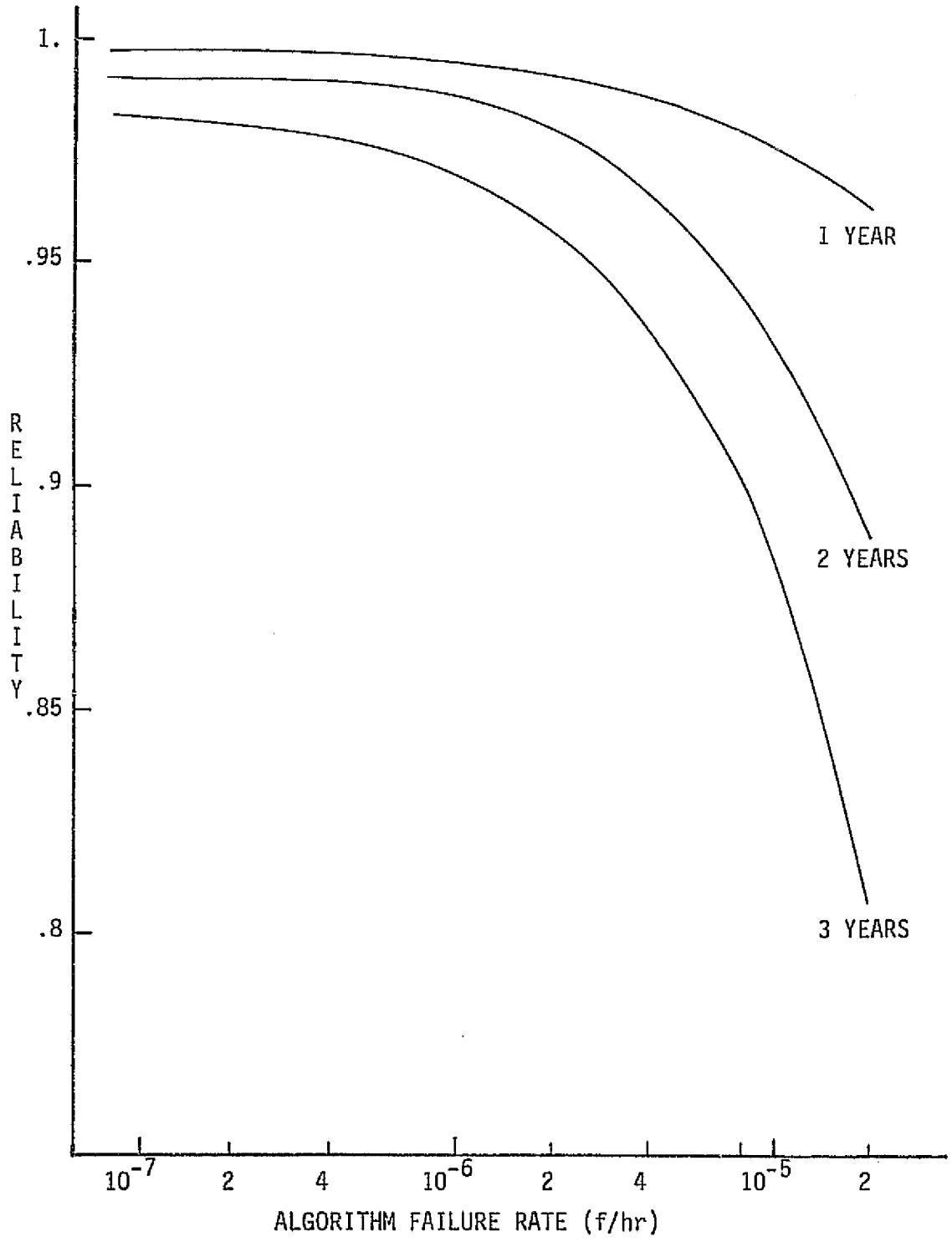


FIGURE 10. Reliability of Double-Error-Correcting System vs. Algorithm Failure Rate

Also essential to overall system reliability is the failure rate of the detector. The effects of varying this failure rate are shown in Figure 11.

The reliability of double-error-correcting systems with various memory capacities is shown in Figure 12. The major effect of memory size on the reliability of a system of this type is in the bit plane failure rate. Also affected are failure rates of memory size-related components such as address decoder circuits, however, only the bit plane failure rates are considered in this figure. The failure rates used were obtained by assuming that each bit plane is composed of 4K-bit sub-planes and peripheral circuitry, each with a failure rate as shown in Table 5.

The results of the memory capacity analysis show that for missions of 1 year or less, double-error-correcting type memories containing up to 64K words will achieve high reliability. Greater mission lengths show a reliability decrease for the larger capacity memories with a dramatic decrease for memories larger than 32K words and a three-year mission length.

The coverage of the basic system for various numbers of spare bit planes is shown in Figure 13. Coverage may be defined as the probability that the system will continue to function given that a failure occurs. As such, the coverage of a system is useful in analyzing the system's behavior after component failures of a nature not predictable by system failure rates.

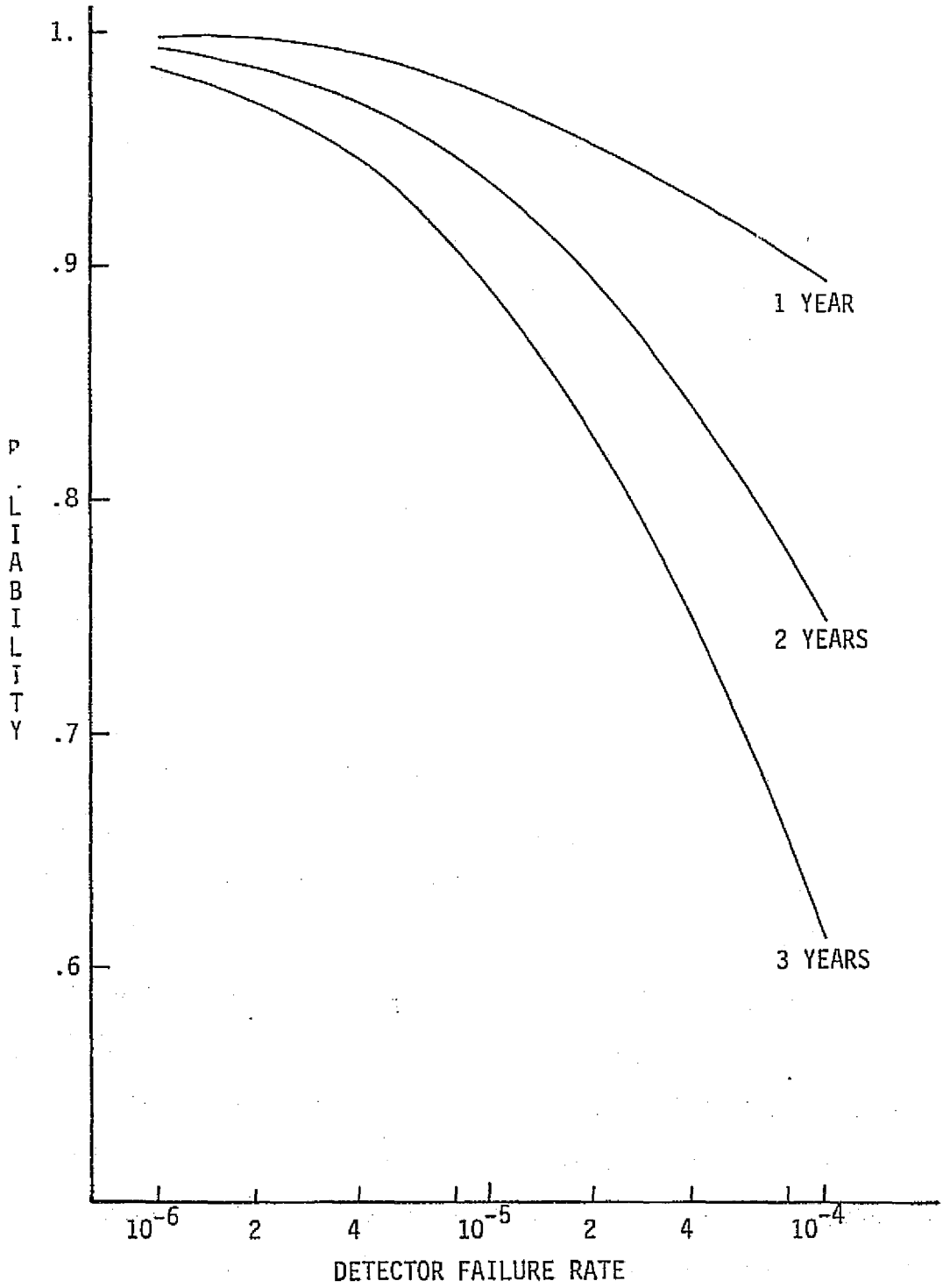


FIGURE 11. Reliability of Double-Error-Correcting System vs. Detector Failure Rate

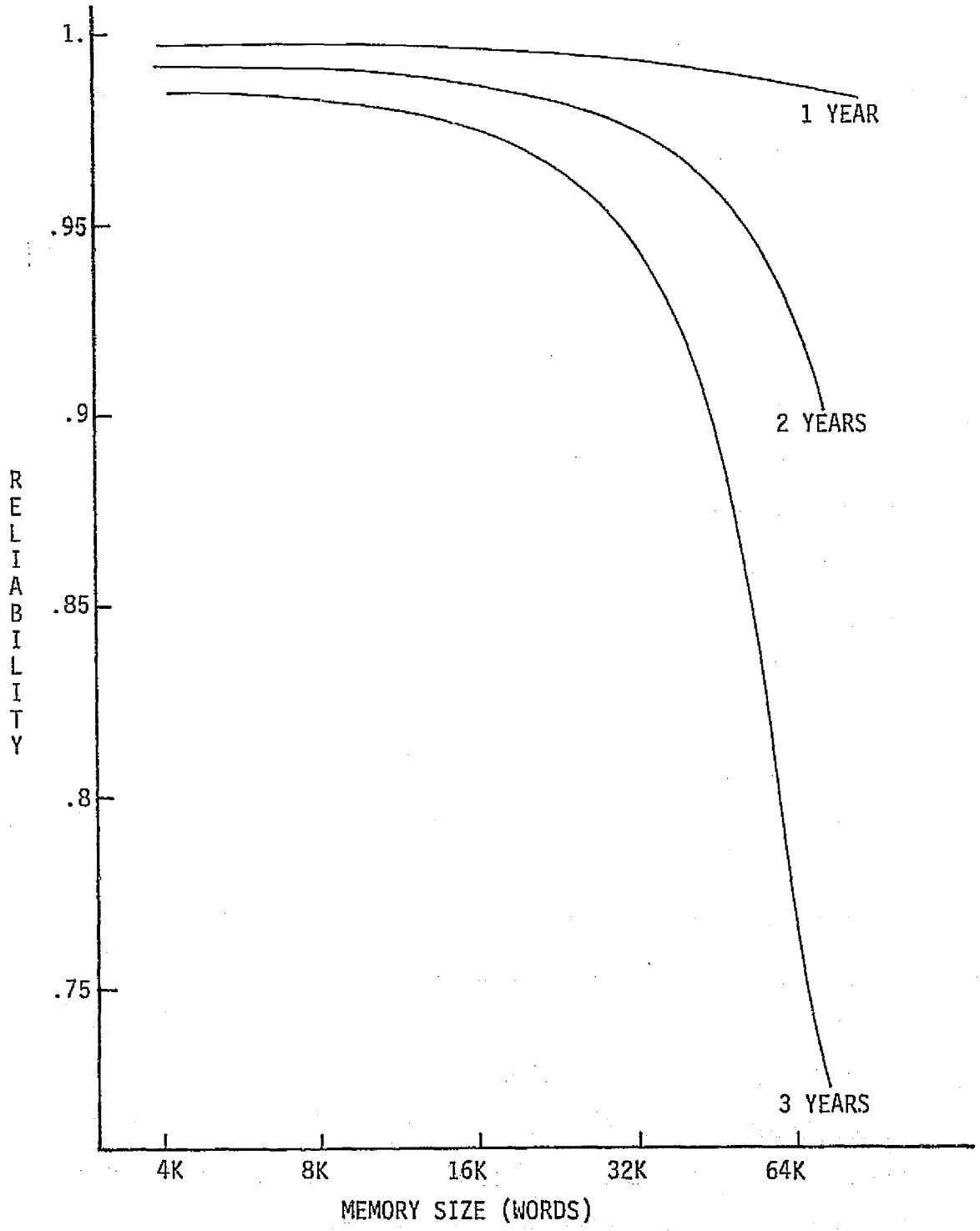


FIGURE 12. Reliability of Double-Error-Correcting System vs. Memory Size.

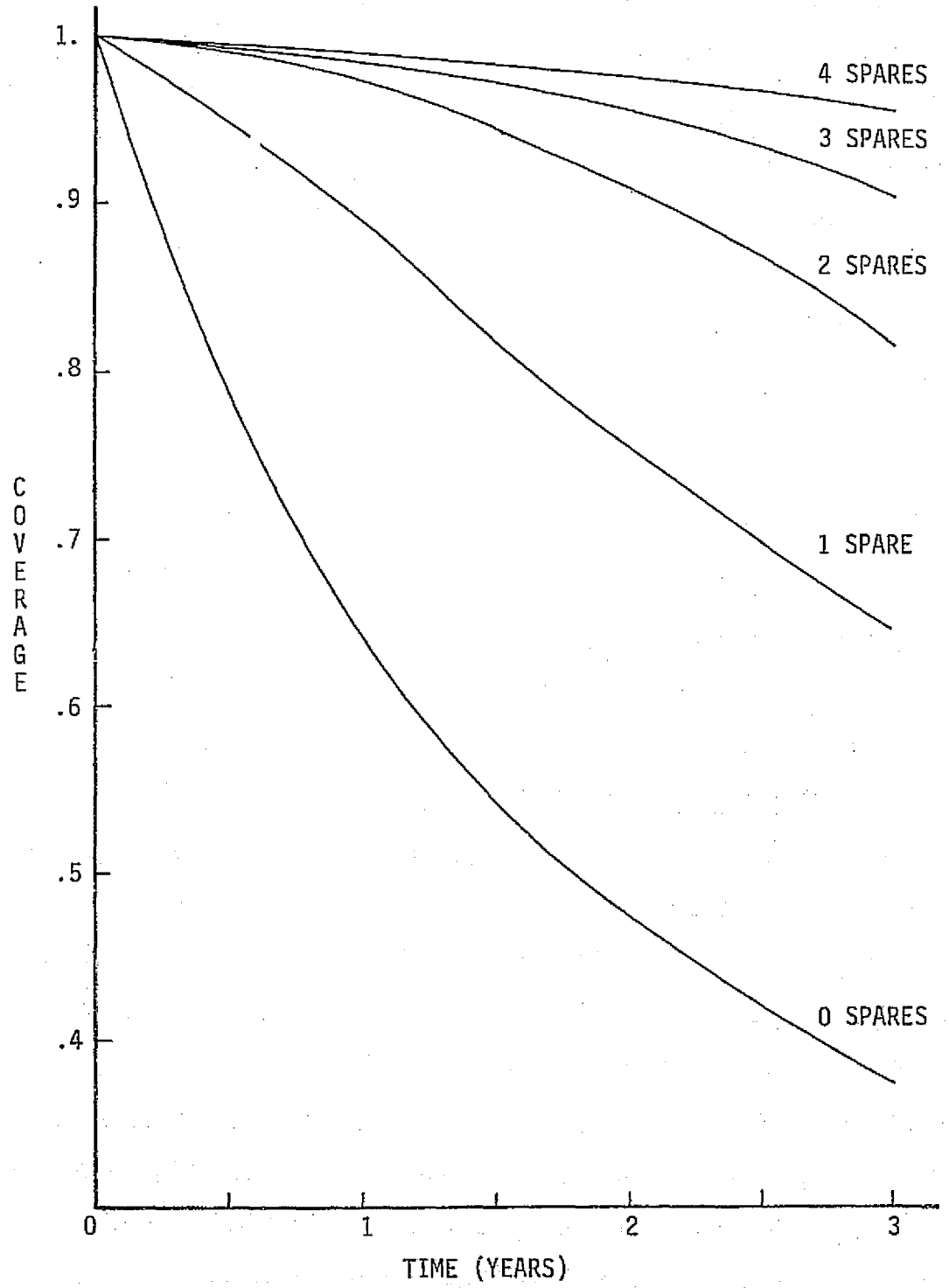


FIGURE 13. Coverage for Basic System

It may be seen from this figure that a basic system with no spares is highly vulnerable to system component failures. As the number of spare bit planes increases, however, this vulnerability decreases rapidly until, in the system with 4 spare bit planes, there is a probability of .96 or greater of successful operation after a failure for missions of 3 years or less.

Overall results of the analyses performed show that a high degree of system reliability may be obtained by a judicious combination of coding, modular sparing, and software error correction. Substantial reliability improvement over massive replication techniques is achieved with relatively low cost. While some sensitivity is shown to the reliability of system control components, fault-tolerant techniques applied to these components should assure high system reliability.

VI. CONCLUSION

A technique for the development of reliability and coverage equations for a class of non-repairable fault-tolerant memory systems has been presented. The methods discussed have been applied to several systems and typical results have been shown.

The basic and double-error-correcting fault-tolerant memory systems have been shown to achieve high reliability at minimal cost. These systems make efficient use of the spare bit-planes provided and the error-correction capabilities of the code. By use of software correction, the double-error-correcting system adds an additional level of error control and may reduce the need for one of the spare bit planes.

A major advantage of the calculation methods presented here over more traditional reliability calculation methods is the allowance of a finite Δt for state transition occurrence. The use of this finite time increment allows multiple system events to occur during any state transition. The need for separate states to represent these events is then diminished. The result is a state diagram with a reduced number of states with probability equations that are easily computer-implemented.

A disadvantage of this method is the lack of a closed form solution which is easily obtainable by use of other methods. Because of the dependency of the state probabilities at time $t + \Delta t$ on the conditions at time t, small errors in computation at one time may cause large errors at succeeding times. A closed form solution should eliminate this problem.

Further work in this area could include the following:

1. Development of a closed-form solution from the equations of this method,

2. Research into the effect of un-powered spares on system modeling, and

3. Application of these methods to the repairable system problem.

REFERENCES

[1] Goldberg, J., Levitt, K. N., and Wensley, J. H., "An Organization for a Highly Survivable Memory," IEEE Trans. on Computers, Vol. C-23, July 1974, pp. 693-705.

[2] Downing, R. W., Nowak, J. S., and Thomenoksa, L. S., "No. 1 ESS Maintenance Plan," Bell System Tech. J., Vol. 43, September 1964, pp. 1961-2019.

[3] Dickinson, et. al., "Saturn V Launch Vehicle Digital Computer and Data Adapter," 1964 Fall Joint Comput. Cong., AFIPS Conf. Proc., 1964, Vol. 26, pp. 501-516.

[4] McCarthy, C. E., Carter, W. C., and White, J. B., "A Memory System Which Can Tolerate Multiple Storage Array Faults," Proceedings of 1975 Southeastern Symposium on System Theory, Auburn, Alabama, pp. 172-178, March 20 - 21, 1975.

[5] Patel, A. M., Hsiao, M. Y., "An Adaptive Error Correction Scheme for Computer Memory System," 1972 Fall Joint Computer Conf., AFIPS Conf. Proc., Vol 41, 1972, pp. 83-85.

[6] Szygenda, S. A. and Flynn, M. J., "Coding Techniques for Failure Recovery in a Distributive Modular Memory Orgainization," 1971 Spring Joint Computer Conf., AFIPS Conf. Proc., Vol. 38, 1971. pp. 459-466.

[7] Szygenda, S. A. and Flynn, M. J., "Failure Analysis of Memory Organizations for Utilization in a Self Repair Memory System," IEEE Trans. on Reliability, Vol. R-20, No. 2, May 1971, pp. 64-70.

[8] Szygenda, S. A. and Flynn, M. J., "Self-Diagnosis and Self-Repair in Memory: An Integrated System Approach," IEEE Trans. on Reliability, Vol. R-22, No. 1, April 1973, pp. 2-12.

[9] Abramson, N. M., "A Class of Systematic Codes for Non-Independent Errors," IRE Transactions on Information Theory, IT-5, No. 4 December 1959, pp. 150-157.

[10] Elspas, B. and Short, R. A., "A Note on Optimum Burst Error-Correcting Codes, IRE Trans. on Info. Theory, IT-8, No. 1, January 1962, pp. 39-42.

[11] Srinivasan, C. V., "Codes for Error Correction in High-Speed Memory Systems: Part II: Correction of Temporary and Catastrophic Errors," IEEE Trans. on Computers, Vol. C-20, No. 12, Dec. 1971, pp. 1514-1520.

[12] Bossen, D. C., "b-Adjacent Error Correction," IBM J. Res Develop., Vol. 14, July 1970, pp. 402-408.

[13] Graham, M., "Error Correction in Batch-Fabricated Memories." IEEE Trans. on Comps. (Corresp.), Vol. C-18, No. 6, June 1969, pp. 566-567.

[14] Rao, T. R. N., "Use of Error Correcting Codes on Memory Words for Improved Reliability," IEEE Trans. on Reliability, Vol. R-17, No. 2 June 1968, pp. 91-96.

[15] Bouricius, W. G., et.al., "Modeling of a Bubble Memory Organization with Self-Checking-Transistors to Achieve High Reliability," IEEE Trans. on Comps., Vol. C-22, No. 3, March 1973, pp. 269-275.

[16] Bricker, J. L., "A Unified Method for Analyzing Mission Reliability for Fault Tolerant Computer Systems," IEEE Trans. on Rel., Vol. R-22, No. 2, June 1973, pp. 72-77.

[17] Jagannathan, T., "General Expressions for Reliability of Redundant Systems," IEEE Trans on Rel., Vol. R-21, No. 2, May 1972, pp. 119.

[18] Benning, C. J., "Reliability Prediction Formulas for Stand By Redundant Structures," (Corresp.) IEEE Trans. on Rel., Vol. R-16, No. 3, December 1967, pp. 136-137.

[19] Hsiao, M. Y., "A Class of Optimal Minimum Odd-Weight-Column SED/DED Codes," IBM J. Res. Dev., Vol. 14, July 1970, pp. 395-401.

[20] Carter, W. C. and McCarthy C. E., "Implementation of an Experimental Fault Tolerant Memory System," IBM Research RC5514 (#23976), July 1975.

[21] Arnold, T. F., "The Concept of Coverage and Its Effect on the Reliability Model of a Repairable System," IEEE Trans. Comps., Vol. C-22, No. 3, March 1973, pp. 251-254.

[22] "Solar Electric Propulsion Stage (SEPS) Command Computer Subsystem Utilizing Space Ultrareliable Modular Computer (SUMU): Vol. 1, Technical," IBM no. 73W-00260, August 1973, pp. 4, 17-4, 19.

[23] Bazovsky, I., Reliability Theory and Practice, Prentice-Hall, Inc., Englewood Cliffs. New Jersey, 1961, pp. 29-30.

[24] Bouricius, W. et.al., "Reliability Modeling for Fault-Tolerant Computers," IEEE Trans. Comps., Vol. C-20, No. 11, November 1971, pp. 1306-1311.

[25] Papoulis, A., Probability, Random Variables, and Stochastic Processes, McGraw-Hill, New York, New York, 1965, p. 37.

[26] Bazovsky, I., Reliability Theory and Practice, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1961, pp. 43-49.

[27] Cox, G. W., "Reliability and Coverage Analyses of Non-Repairable Fault-Tolerant Memory Systems," Master's Thesis, Auburn University, Auburn, AL, 1976.

APPENDIX A

Flowcharts for Computational Algorithms

Three methods for computer evaluation of the equations of this paper were outlined in Chapter III. Flowcharts for evaluation by use of Methods 2 and 3 are shown here.

Figure 14 shows a typical implementation of evaluation Method 2. For this flowchart, t_{BASE} is selected to be 0 and the system starting state is state 1. TMAX is the mission length of interest.

After initialization, all transition probabilities are calculated for the current time (T) and Δt. Where a two-state transition is possible, the product of the two single-state transitions involved is formed. If this product is greater than PMAX, the maximum allowable two-state transition probability, the Δt is reduced.

The amount of this reduction is arbitrary. If Δt and T have units of hours, then a convenient method of reduction is to multiply Δt by .9 and set the new Δt equal to the greatest integral number of hours less than this number. When this method is used, however, a test must be performed to assure that Δt is not 0 since this condition would prevent any further processing.

If all the two-state transition probabilities are less than PMAX, the state probabilities for time $T + \Delta t$ are computed by substitution of the transition probabilities and state probabilities for time T into equation (3-1), the general state probability equation. If T is less

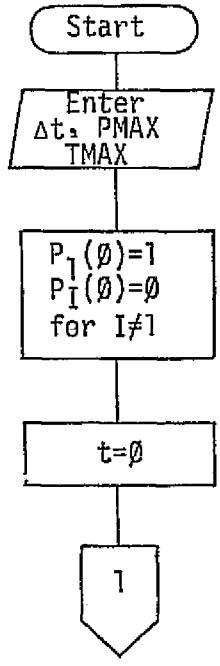


Figure 14. Flowchart for Reliability Computations by Method 2.

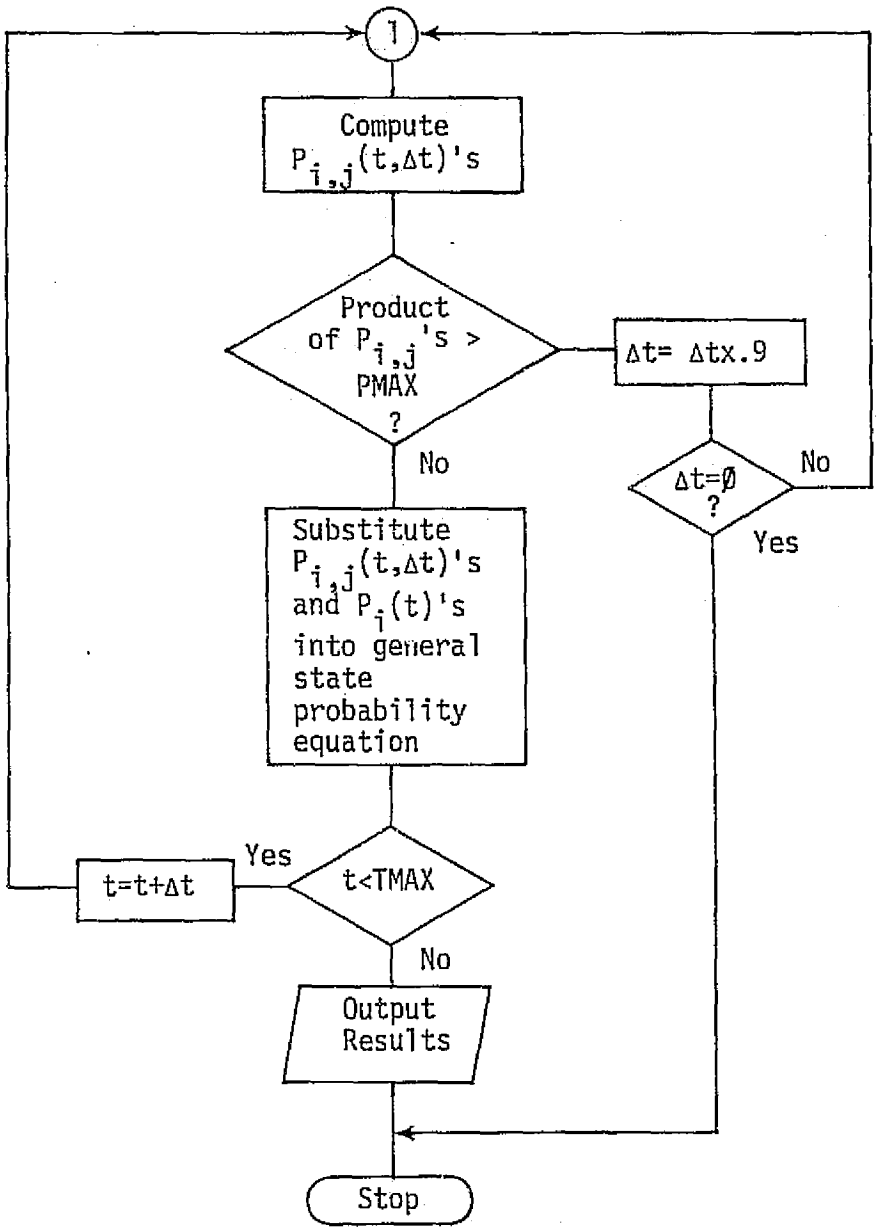


Figure 14. Continued

than TMAX, T is incremented by Δt and processing continues. Otherwise, the system reliability is formed as a suitable sum of state probabilities, results are output, and processing terminates.

Figure 15 shows an implementation of a Method 3 evaluation. This flowchart follows the steps outlined in the second computational algorithm of Chapter III.

It should be noted from equation (3-11) that if the base computation time is $\emptyset$ and the system starting state is state i so that $P_i(\emptyset) = 1$ then $\overline{T}_{n^*}$ contains the state probabilities for state j in its (i,j) location. For this case, then, the multiplication by $\underline{P}(t)$ to obtain $\underline{P}(t + n\Delta t)$ is unnecessary since the state probabilities may be determined directly.

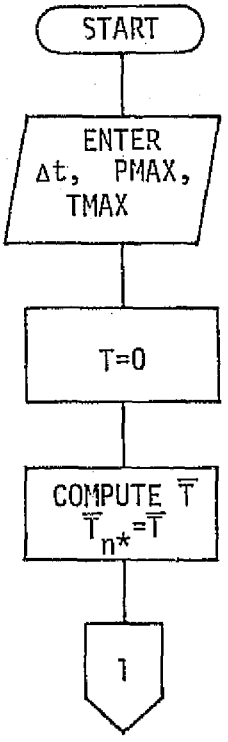


FIGURE 15. Flowchart for Reliability Computations by Method 3

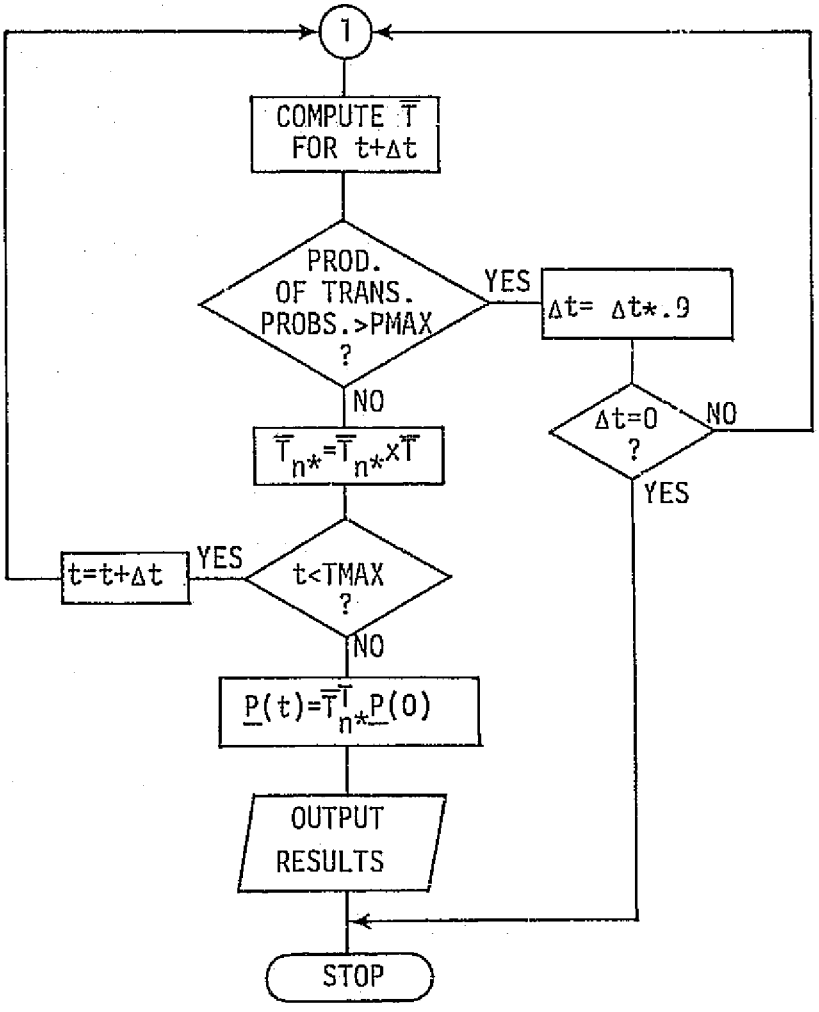


FIGURE 15. Continued

APPENDIX B

Development of Equations for the Double-Error-Correcting System

A listing of transition events and subevents causing the transitions is shown in Table 6. In this table, the success of the detector, corrector, correction algorithm and switch prior to time $t + \Delta t$ are represented by D', C', A', and W', respectively. Success in the time interval from t to $t + \Delta t$ is denoted by a "*" superscript. The non-success event is denoted by a subtraction of the appropriate symbol from 1.

For the derivation of the transition and state probability equations, the following notation will be used.

$$D(x,y) = P(\text{y correctable on-line bit plane errors out of x possible on-line bit planes given all were good at time t})$$

$$= \binom{x}{y}(r(\Delta t))^{(x-y)}(1-r(\Delta t))^{y}$$

$$E(x,y) = P(\text{y or fewer good spare bit planes out of s - x + 2 available})$$

$$= \sum_{k=0}^{y} \binom{s-x+2}{k}(1-r')^{(s-x+2-k)}(r')^{k}$$

$$r_m^* = r_m(\Delta t)$$

The double-error-correcting system transition probability equations may now be specified as

TABLE 6. Events and Subevents for Double-Error-Correcting System

TRANSITION	SUBEVENTS CAUSING TRANSITION	
	# on-line correctable B-P Errors / # possible bits	Other Subevents
1,2	1/K	D~C~A~
1,3	2/K	D~C~A~W~(at least 1 good spare)
1,4	3/K	D~C~A~W~(at least 2 good spares)
1,5	4/K	D~C~A~W~(at least 3 good spares)
1,s+3	2/K	D~C~A~((1-W~) or W~(No good spares))
		or
	3/K	D~C~A~W~(exactly 1 good spare)
		or
	4/K	D~C~A~W~(exactly 2 good spares)
1,s+4	3/K	D~C~A~((1-W~) or W~(no good spares))
		or
	4/K	D~C~A~W~(exactly 1 good spare)
1,s+5	4/K	D~C~A~((1-W~) or W~(no good spares))
1,1	Ø/K	-
i,i+1	1/K-1	D*C*A*W~(at least 1 good spare)
$2 \leq i \leq s+1$		

TABLE 6. continued

i,i+2	2/K-1	D*C*A*W'(at least 2 good spares)
$2 \leq i \leq s$		
i,i+3	3/K-1	D*C*A*W'(at least 3 good spares)
$2 \leq i \leq s-1$		
i,s+3	1/K-1	D*C*A* ((1-W') or W'(no good spares)
$2 \leq i \leq s$		or
	2/K-1	D*C*A*W'(exactly 1 good spare)
		or
	3/K-1	D*C*A*W'(exactly 2 good spares)
i,s+4	2/K-1	D*C*A*((1-W') or W'(no good spares))
$2 \leq i \leq s+1$		or
	3/K-1	D*C*A*W'(exactly 1 good spare)
i,s+5	3/K-1	D*C*A*((1-W') or W'(no good spares))
$2 \leq i \leq s+1$		
i,i	0/K-1	D*C*A*
$2 \leq i \leq s+2$		
s+1,s+3	1/K-1	D*C*A*((1-W') or W'(no good spares))
		or
	2/K-1	D*C*A*W'(exactly 1 good spare)
s+2,s+j	j-2/K-1	D*C*A*
$3 \leq j \leq 5$		

TABLE 6. continued

s+3,s+j 4≤j≤5	j-3/K-2	D*C*A*
s+4,s+5	1/K-3	D*C*A*
s+j,s+j 3≤j≤5	0/K-j+1	D*C*A*

The double-error-correcting system transition probability equations may now be specified as:

$$P_{1,2}(t,\Delta t) = D(k,1)\ r_d' r_c' r_A'.$$

$$P_{1,3}(t,\Delta t) = D(k,2)\ r_d' r_c' r_A' r_s' (1-E(2,0)).$$

$$P_{1,4}(t,\Delta t) = D(k,3)\ r_d' r_c' r_A' r_s' (1-E(2,1)).$$

$$P_{1,5}(t,\Delta t) = D(k,4)\ r_d' r_c' r_A' r_s' (1-E(2,2)).$$

$$P_{1,s+3}(t,\Delta t) = D(k,2)\ r_d' r_c' r_A' (1-r_s' + r_s' E(2,0)) + D(k,3)\ r_d' r_c' r_A' r_s' (E(2,1)-E(2,0)) + D(k,4)\ r_d' r_c' r_A' r_s' (E(2,2)-E(2,1)).$$

$$P_{1,s+4}(t,\Delta t) = D(k,3)\ r_d' r_c' r_A' (1-r_s' + r_s' E(2,0)) + D(k,4)\ r_d' r_c' r_A' r_s' (E(2,1)-E(2,0)).$$

$$P_{1,s+5}(t,\Delta t) = D(k,4)\ r_d' r_c' r_A' (1-r_s' + r_s' E(2,0)).$$

$$P_{1,1}(t,\Delta t) = D(k,0).$$

$$P_{i,i+1}(t,\Delta t) = D(k-1,1)\ r_d * r_c * r_A * r_s' (1-E(i,0))$$

for $2 \le i \le s+1$.

$$P_{i,i+2}(t,\Delta t) = D(k-1,2)\ r_d * r_c * r_A * r_s' (1-E(i,1))$$

for $2 \le i \le s$.

$$P_{i,i+3}(t,\Delta t) = D(k-1,3)\ r_d * r_c * r_A * r_s' (1-E(i,2))$$

for $2 \le i \le s-1$.

$$P_{i,s+3}(t,\Delta t) = D(k-1,1)\ r_d{*}r_c{*}r_A{*}\ (1-r_s' + r_s'\ E(i,0))$$
$$+ D(k-1,2)\ r_d{*}r_c{*}r_A{*}r_s'\ (E(i,1)-E(i,0))$$
$$+ D(k-1,3)\ r_d{*}r_c{*}r_A{*}r_s'\ (E(i,2)-E(i,1))$$

for $2 \le i \le s$.

$$P_{i,s+4}(t,\Delta t) = D(k-1,2)\ r_d{*}r_c{*}r_A{*}\ (1-r_s' + r_s'\ E(i,0))$$
$$+ D(k-1,3)\ r_d{*}r_c{*}r_A{*}r_s'\ (E(i,1)-E(i,0))$$

for $2 < i < s+1$.

$$P_{i,s+5}(t,\Delta t) = D(k-1,3)\ r_d{*}r_c{*}r_A{*}\ (1-r_s' + r_s'\ E(i,0))$$

for $2 \le i \le s+1$.

$$P_{i,i}(t,\Delta t) = D(k-1,0)\ r_d{*}r_c{*}r_A{*}$$

for $2 \le i \le s+2$.

$$P_{s+1,s+3}(t,\Delta t) = D(k-1,1)\ r_d{*}r_c{*}r_A{*}\ (1-r_s' + r_s'\ E(s+1,0))$$
$$+ D(k-1,2)\ r_d{*}r_c{*}r_A{*}r_s'\ (E(s+1,1)-E(s+1,0)).$$

$$P_{s+2,s+j}(t,\Delta t) = D(k-1,j-2)\ r_d{*}r_c{*}r_A{*}$$

for $3 \le j \le 5$.

$$P_{s+3,s+j}(t,\Delta t) = D(k-2,j-3)\ r_d{*}r_c{*}r_A{*}.$$

for $4 \le j \le 5$.

$$P_{s+4,s+5}(t,\Delta t) = D(k-3,1)\ r_d{*}r_c{*}r_A{*}.$$

$$P_{s+j,s+j}(t,\Delta t) = D(k-j+1,0)\ r_d{*}r_c{*}r_A{*}$$

for $3 \le j \le 5$.

Since

$$P_{i,FAIL}(t,\Delta t) = 1 - \sum_{j=1}^{s+5} P_{i,j}(t,\Delta t),$$

the following equations may be developed:

$$P_{1,FAIL}(t,\Delta t) = 1 - P_{1,2}(t,\Delta t) - P_{1,4}(t,\Delta t) - P_{1,5}(t,\Delta t) - P_{1,s+3}(t,\Delta t) - P_{1,s+4}(t,\Delta t) - P_{1,1}(t,\Delta t)$$

$$\begin{aligned} P_{1,FAIL}(t,\Delta t) &= 1 - D(k,0) - D(k,1)\, r_d' r_c' r_A' \\ &\quad - D(k,2) r_d' r_c' r_A' [r_s'(1-E(2,0)+1-r_s' + r_s' E(2,0)] \\ &\quad - D(k,3) r_d' r_c' r_A' [r_s'(1-E(2,1))+r_s'(E(2,1)-E(2,0)) \\ &\quad + 1-r_s' + r_s'\, E(2,0)] - D(k,4)\, r_d' r_c' r_A' \\ &\quad \cdot [r_s'\,(1-E(2,2)) + r_s'\,(E(2,2)-E(2,1)) \\ &\quad + r_s'(E(2,1)-E(2,0)) + 1-r_s' + r_s'\, E(2,0)] \\ &= 1 - D(k,0) - D(k,1)\, r_d' r_c' r_A' - D(k,2)\, r_d' r_c' r_A' \\ &\quad - D(k,3)\, r_d' r_c' r_A' - D(k,4)\, r_d' r_c' r_A' \\ &= 1 - D(k,0) - r_d' r_c' r_A' \sum_{j=1}^{4} D(k,j). \end{aligned}$$

$$P_{i,FAIL}(t,\Delta t) = 1 - P_{i,i+1}(t,\Delta t) - P_{i,i+2}(t,\Delta t) - P_{i,i+3}(t,\Delta t) - P_{i,s+3}(t,\Delta t) - P_{i,s+4}(t,\Delta t) - P_{i,s+5}(t,\Delta t) - P_{i,i}(t,\Delta t)$$

for $1 \le i \le s-1$

$$P_{i,FAIL}(t,\Delta t) = 1 - D(k-1,0)\, r_d^* r_c^* r_A^* - D(k-1,1)\, r_d^* r_c^* r_A^*$$
$$\cdot [r_s' (1-E(i,0)) + 1 - r_s' + r_s' E(i,0)]$$
$$- D(k-1,2)\, r_d^* r_c^* r_A^* [r_s' (1-E(i,1))$$
$$+ r_s' (E(i,1) - E(i,0)) + 1 - r_s' + r_s' E(i,0)]$$
$$- D(k-1,3)\, r_d^* r_c^* r_A^* [r_s' (1-E(i,2))$$
$$+ r_s' (E(i,2) - E(i,1)) + 1 - r_s' + r_s' E(i,0)$$
$$+ r_s' (E(i,1) - E(i,0))]$$
$$= 1 - r_d^* r_c^* r_A^* [D(k-1,0)+D(k-1,1)+D(k-1,2)$$
$$+ D(k-1,3)].$$

$$P_{s,FAIL}(t,\Delta t) = 1-P_{s,s+1}(t,\Delta t)-P_{s,s+2}(t,\Delta t)-P_{s,s+3}(t,\Delta t)$$
$$-P_{s,s+4}(t,\Delta t)-P_{s,s+5}(t,\Delta t)-P_{s,s}(t,\Delta t)$$
$$= 1 - r_d^* r_c^* r_A^* \; D(k-1,0) + D(k-1,1)$$
$$\cdot [r_s' (1-E(s,0)) + 1 - r_s' + r_s' E(s,0)]$$
$$+ D(k-1,2)[r_s' (1-E(s,1)) + r_s' (E(s,1)-E(s,0))$$
$$+ 1-r_s' + r_s' E(s,0)] + D(k-1,3)[r_s' E(s,2)-E(s,1)$$
$$+ r_s' (E(s,1)-E(s,0)) + 1-r_s' + r_s' E(s,0)]$$
$$= 1 - r_d^* r_c^* r_A^* \; D(k-1,0)+D(k-1,1)+D(k-1,2)$$
$$+ D(k-1,3)[r_s' E(s,2) + 1 - r_s']$$

But $E(s,2) = \sum_{q=0}^{2} \binom{2}{q} (1-r')^{(2-q)} (r')^q = 1$

so

$$P_{s,FAIL}(t,\Delta t) = 1 - r_d * r_c * r_A * [D(k-1,0) + D(k-1,1) + D(k-1,2) + D(k-1,3)].$$

$$P_{s+1,FAIL}(t,\Delta t) = 1 - P_{s+1,s+2}(t,\Delta t) - P_{s+1,s+3}(t,\Delta t) - P_{s+1,s+5}(t,\Delta t) - P_{s+1,s+1}(t,\Delta t)$$

$$= 1 - r_d * r_c * r_A * \; D(k-1,0) + D(k-1,1)[r_s'(1-E(s+1,0)) + 1 - r_s' + r_s' E(s+1,0)] + D(k-1,2) \cdot [1 - r_s' + r_s' E(s+1,0) + r_s'(E(s+1,1) - E(s+1,0))] + D(k-1,3)[r_s'(E(s+1,1) - E(s+1,0)) + 1 - r_s' + r_s' E(i,0)]$$

$$P_{s+1,FAIL}(t,\Delta t) = 1 - r_d * r_c * r_A * \; D(k-1,0) + D(k-1,1) + D(k-1,2) \cdot [1 - r_s' + r_s' E(s+1,1)] + D(k-1,3) [r_s' E(i,1) + 1 - r_s']$$

But $E(s+1,1) = \sum_{q=0}^{1} \binom{1}{q} (1-r')^{(1-q)} (r')^q = 1$

So

$$P_{s+1,FAIL}(t,\Delta t) = 1 - r_d * r_c * r_A * [D(k-1,0) + D(k-1,1) + D(k-1,2) + D(k-1,3)].$$

$$P_{s+2,FAIL}(t,\Delta t) = 1 - P_{s+2,s+2}(t,\Delta t) - P_{s+2,s+3}(t,\Delta t) - P_{s+2,s+4}(t,\Delta t) - P_{s+2,s+5}(t,\Delta t)$$

$$= 1 - r_d * r_c * r_A * [D(k-1,0) + D(k-1,1) + D(k-1,2) + D(k-1,3)].$$

$$P_{s+3,FAIL}(t,\Delta t) = 1-P_{s+3,s+4}(t,\Delta t)-P_{s+3,s+5}(t,\Delta t)-P_{s+3,s+3}(t,\Delta t)$$

$$= 1-r_d*r_c*r_A*[D(k-2,0)+D(k-2,1)+D(k-2,2)].$$

$$P_{s+4,FAIL}(t,\Delta t) = 1-P_{s+4,s+4}(t,\Delta t)-P_{s+4,s+5}(t,\Delta t)$$

$$= 1-r_d*r_c*r_A*[D(k-3,0)+D(k-3,1)].$$

$$P_{s+5,FAIL}(t,\Delta t) = 1-P_{s+5,s+5}(t,\Delta t) = 1-r_d*r_c*r_A* D(k-4,0).$$

So

$$P_{1,FAIL}(t,\Delta t) = 1-D(k,0)-r_d\,r_c\,r_A\,\sum_{j=1}^{4} D(k,j).$$

$$P_{i,FAIL}(t,\Delta t) = 1-r_d*r_c*r_A*\sum_{j=0}^{3} D(k-1,j)$$

for $2 \le i \le s+2$.

$$P_{s+j,FAIL}(t,\Delta t) = 1-r_d*r_c*r_A*\sum_{q=0}^{5-j} D(k-j+1,q)$$

for $3 \le j \le 5$.

By substitution of the transition probability equations into the general state probability equation, the state probability equations for the double-error-correcting system are obtained as follows:

$$P_1(t + \Delta t) = P_{1,1}(t,\Delta t)P_1(t)$$

$$= D(k,0)P_1(t).$$

$$P_2(t + \Delta t) = P_{1,2}(t,\Delta t)P_2(t) + P_{2,2}(t,\Delta t)P_2(t)$$

$$= D(k,1)\ r_d\,r_c\,r_A\,P_1(t)$$

$$+ D(k-1,0)\ r_d*r_c*r_A*\ P_2(t).$$

$$P_3(t+\Delta t) = P_{1,3}(t,\Delta t)P_1(t) + P_{2,3}(t,\Delta t)P_2(t) + P_{3,3}(t,\Delta t)P_3(t)$$

$$= D(k,2)\, r_d \cdot r_c \cdot r_A \cdot r_s \cdot (1-E(2,0))P_1(t)$$

$$+ r_d * r_c * r_A * [D(k-1,1) r_s \cdot (1-E(2,0))P_2(t)$$

$$+ D(k-1,0)P_3(t)].$$

$$P_4(t+\Delta t) = P_{1,4}(t,\Delta t)P_1(t) + P_{2,4}(t,\Delta t)P_2(t) + P_{3,4}(t,\Delta t)P_3(t)$$

$$+ P_{4,4}(t,\Delta t)P_4(t)$$

$$= D(k,3)\, r_d \cdot r_c \cdot r_A \cdot r_s \cdot (1-E(2,1))P_1(t)$$

$$+ r_d * r_c * r_A * [D(k-1,2) r_s \cdot (1-E(2,1))P_2(t)$$

$$+ D(k-1,1)\, r_s \cdot (1-E(3,0))P_3(t) + D(k-1,0)P_4(t)].$$

$$P_5(t+\Delta t) = P_{1,5}(t,\Delta t)P_1(t) + P_{2,5}(t,\Delta t)P_2(t)$$

$$+ P_{3,5}(t,\Delta t)P_3(t) + P_{4,5}(t,\Delta t)P_4(t) + P_{5,5}(t,\Delta t)P_5(t)$$

$$= D(k,4) r_d \cdot r_c \cdot r_A \cdot r_s \cdot (1-E(2,2))P_1(t)$$

$$+ r_d * r_c * r_A * [D(k-1,3) r_s \cdot (1-E(2,2))P_2(t)$$

$$+ D(k-1,2) r_s \cdot (1-E(3,1))P_3(t)$$

$$+ D(k-1,1) r_s \cdot (1-E(4,0))P_4(t) + D(k-1,0)P_5(t)].$$

$$P_i(t+\Delta t) = P_{i-3,i}(t,\Delta t)P_{i-3}(t) + P_{i-2,i}(t,\Delta t)P_{i-2}(t)$$

$$+ P_{i-1,i}(t,\Delta t)P_{i-1}(t) + P_{i,i}(t,\Delta t)P_i(t)$$

$$= r_d * r_c * r_A * [D(k-1,3) r_s \cdot (1-E(i-3,2))P_{i-3}(t)$$

$$+ D(k-1,2) r_s \cdot (1-E(i-2,1))P_{i-2}(t)$$

$$+ D(k-1,1) r_s \cdot (1-E(i-1,0))P_{i-1}(t) + D(k-1,0)P_i(t)]$$

$$P_i(t+\Delta t) = r_d{}^* r_c{}^* r_A{}^* \left[D(k-1,0)P_i(t) + r_s' \sum_{j=1}^{3} D(k-1,j)(1-E(i-j,j-1))P_{i-j}(t)\right]$$

for $6 \le i \le s+2$.

$$P_{s+3}(t+\Delta t) = \sum_{j=1}^{s+2} P_{j,s+3}(t,\Delta t)P_j(t) + P_{s+3,s+3}(t,\Delta t)P_{s+3}(t)$$

$$= r_d' r_c' r_A' [D(k,2)(1-r_s' + r_s' E(2,0)) + D(k,3) r_s' \cdot (E(2,1)-E(2,0)) + D(k,4) r_s' (E(2,2)-E(2,1))]P_1(t)$$

$$+ r_d{}^* r_c{}^* r_A{}^* \sum_{j=2}^{s} [D(k-1,1)(1-r_s' + r_s' E(j,0)) + D(k-1,2) r_s' (E(j,1)-E(j,0)) + D(k-1,3) r_s' (E(j,2)-E(j,1))]P_j(t)$$

$$+ [D(k-1,1)(1-r_s' + r_s' E(s+1,0) + D(k-1,2)\, r_s' (E(s+1,1) - E(s+i,0))]P_{s+1}(t)$$

$$+ D(k-1,1)P_{s+2}(t) + D(k-2,0)P_{s+3}(t) \quad .$$

$$P_{s+4}(t+\Delta t) = \sum_{j=1}^{s+3} P_{j,s+4}(t,\Delta t)P_j(t) + P_{s+4,s+4}(t,\Delta t)P_{s+4}(t)$$

$$= r_d' r_c' r_A' [D(k,3)(1-r_s' + r_s' E(2,0)) + D(k,4)\, r_s' (E(2,1) - E(2,0))]P_1(t)$$

$$+ r_d{}^* r_c{}^* r_A{}^* \sum_{j=2}^{s+1} [D(k-1,2)(1-r_s' + r_s' E(j,0)) + D(k-1,3) r_s' (E(j,1)-E(j,0))]P_j(t)$$

$$+ D(k-1,2)P_{s+2}(t) + D(k-2,1)P_{s+3}(t) + D(k-3,0)P_{s+4}(t) \quad .$$

$$P_{s+5}(t+\Delta t) = \sum_{j=1}^{s+4} P_{j,s+5}(t,\Delta t)P_j(t) + P_{s+5,s+5}(t,\Delta t)P_{s+5}(t)$$

$$= D(k,4)\, r_d' r_c' r_A' \,(1-r_s' + r_s' E(2,0))P_1(t)$$

$$+ r_d^* r_c^* r_A^* \sum_{j=2}^{s+1} D(k-1,3)(1-r_s' + r_s' E(j,0))P_j(t)$$

$$+ D(k-1,3)P_{s+2}(t) + D(k-2,2)P_{s+3}(t)$$

$$+ D(k-3,1)P_{s+4}(t) \quad .$$

$$P_{FAIL}(t+\Delta t) = \sum_{j=1}^{s+5} P_{j,FAIL}(t,\Delta t)P_j(t) + P_{FAIL}(t)$$

$$= [1-D(k,0) - r_d' r_c' r_A' \sum_{j=1}^{4} D(k,j)]P_1(t)$$

$$+ \sum_{q=2}^{s+2} [1-r_d^* r_c^* r_A^* \sum_{m=0}^{3} D(k-1,m)]P_q(t)$$

$$+ \sum_{n=3}^{5} [1-r_d^* r_c^* r_A^* \sum_{q=0}^{5-h} D(k-n+1,q)]P_{s+n}(t) + P_{FAIL}(t)$$

$$= \sum_{r=1}^{s+5} P_r(t) + P_{FAIL}(t) + [-D(k,0) - r_d' r_c' r_A'$$

$$\cdot \sum_{j=1}^{4} D(k,j)]P_1(t) + r_d^* r_c^* r_A^*$$

$$\Big[\sum_{q=2}^{s+2} \Big(\sum_{m=0}^{3} D(k-1,m)\Big)P_q(t)$$

$$- \sum_{n=3}^{5} \Big(\sum_{q=0}^{5-n} D(k-n+1,q)P_{s+n}(t)\Big)\Big]$$

But $\sum_{r=1}^{s+5} P_r(t) + P_{FAIL}(t) = 1$

So

$$P_{FAIL}(t + \Delta t) = 1 - [D(k,0) + r_d \acute{} r_c \acute{} r_A \acute{} \sum_{j=1}^{4} D(k,j)]P_1(t)$$

$$- r_d * r_c * r_A * [\sum_{q=2}^{s+2} [\sum_{m=0}^{3} D(k-1,m)P_q(t)$$

$$+ \sum_{n=3}^{5} (\sum_{q=0}^{5-n} D(k-n+1,q))P_{s+n}(t)]$$