A METHOD FOR THEORETICALLY OBTAINING AND PHYSICALLY REALIZING UNLIMITED RELIABILITY THROUGH REDUNDANCY

By

J. B. WHITE, JR.

Bachelor of Science

Oklahoma State University

Stillwater, Oklahoma

1956

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
DOCTOR OF PHILOSOPHY
May, 1971

A METHOD FOR THEORETICALLY OBTAINING
AND PHYSICALLY REALIZING UNLIMITED
RELIABILITY THROUGH REDUNDANCY

Thesis Approval:

Paul A. McCollum
Thesis Adviser

Henry R. Sebesta

J. F. Norton

Harold Tristoe

N. N. Durham
Dean of the Graduate College

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to the Marshall Space Flight Center of the National Aeronautics and Space Administration, not only for sponsoring my resident program but also for providing the stimulation, facilities, and atmosphere necessary to carry out the research presented herein.

In addition, I would like to thank my entire Graduate Committee for their valuable advice and assistance. Special thanks are due Professor P. A. McCollum, my Thesis Adviser, without whose lectures and instruction in digital systems much of this work would not have been possible. I will always be deeply indebted to Dr. J. R. Norton, Head of the School of General Engineering and Chairman of the Graduate Committee, for his advice and counseling. He has significantly influenced my graduate work as well as my philosophical outlook on life.

Finally, I wish to thank my entire family for their contributions; Carole, without whose patience, sacrifices, and constant goading this work would not have been accomplished; Doug, whose understanding of the absence of Dad at ball games and the fishing hole is greatly appreciated; for completeness, baby Karen should also be mentioned whose influence, if any, admittedly must have had an adverse effect.

TABLE OF CONTENTS

Chapter | Page

LIST OF TABLES

Table Page

Table		Page

LIST OF FIGURES

Figure		Page

Figure		Page

Figure | Page

CHAPTER I

INTRODUCTION

1.1 PREVIOUS WORK IN THE FIELD

The general subject of reliability may be subdivided into many categories such as reliability prediction and analysis, reliability measurement, redundancy, etc. Balaban (1) presents one method of classification and a selected bibliography on reliability in general. Although there is no unique or universally accepted classification of reliability, redundancy is commonly considered to be one of the subclasses of reliability theory and practice and is of primary concern in this investigation.

Since initially proposed in 1956 by J. von Neuman (20), the area of "synthesis of reliable organisms from unreliable components" has been given considerable attention. Redundancy, as defined by Webster, is "quality, instance, or state of being redundant," and redundant is defined as "exceeding what is natural and necessary" or as "being superfluous." This connotation is rapidly becoming outdated, since redundancy may be an absolute requirement and the only means by which an extremely high reliability can be achieved.

Short (18) presents an excellent bibliography in the redundancy field, listing 347 sources which are indicative of the rather concentrated effort in this field since 1956.

Historically, reliability improvement has been attacked through simplicity in concept, conservative design, utilization of highly reliable component parts, and extensive test programs and procedures. Within the past two decades, tremendous strides have been taken in the improvement of component part reliability. For example, in electronic circuitry, the transistor demonstrated a marked reliability improvement in comparison to the electron tube, and, in more recent years, microminiaturization and integrated circuits have contributed significantly to the improvement of electronic circuit reliability. Although large-scale integrated circuits are presently being used in a limited sense, they will be massively employed in future systems, which will result in another significant improvement. However, even with these advances in basic technology, overall system reliability, in many cases, will not improve sufficiently to meet tomorrow's critical demands because (1) systems are becoming more sophisticated and are, therefore, more complex, and (2) systems are being required to operate over extended periods of time. Therefore, other techniques must be employed, and redundancy provides a means of increasing reliability beyond the point which can be obtained through basic technology alone.

Several redundant forms, or configurations, have been discussed in the literature. Typical examples are duplexing, quadruplexing, one-out-of-n parallel redundancy, and majority logic. The investigation herein is primarily concerned with the development of the unique two-out-of-n configuration which is derived basically from the concept of majority logic. Although the term "majority logic" will not be employed extensively beyond Chapter II because it is no longer descriptive of the configuration under study, the literature on majority logic provides a firm foundation on which this investigation is based. Rozenberg and Ergott (14) have treated two-out-of-three majority logic and have shown that the mean time to failure of output voting is greater than that of input voting. Teoste (19) has shown that the mean time between failures of digital electronic equipment can be increased by several orders of magnitude by the use of von Neumann's multiplexing redundancy. However, the mean time between failures is not always a meaningful parameter to employ when comparing redundant and non-redundant configurations or in comparing various forms of redundant configurations. The best placement of voters in a triplicated logic network is treated by Gurzi (8), who shows that the utilization of a voter with each module employed is to be preferred to a single voter per redundant module. In his work, however, the logic necessary to perform the voting function is not taken into consideration. Lyons and Vanderkulk (11) discuss the use of the triple modular redundancy technique and point out the possibility that, in addition to voting or fault masking, failure detection and isolation are possible; but they do not consider the logic necessary to accomplish this function. Failure detection and isolation in a triple modular configuration may be very important in reducing maintenance problems and may be employed solely for that purpose rather than just for increasing system reliability. Triple modular redundancy is also treated by Brown, Tierney, and Wasserman (5) who also consider the logical design of the voter. The literature indicates that very little work has been done in majority logic of degrees greater than three.

In addition to the study and analysis of a two-out-of-n configuration, a major effort in this investigation will be made to optimize the redundant system in the presence of constraints. Many excellent papers concerning system optimization are available. Bellman and Dreyfus (4) treat the generalized approach of dynamic programming and show how it can be applied to optimizing redundant systems. Least-cost allocations of reliability investment are considered by Kettelle (10) who utilizes the dynamic programming approach and another method which he says yields an explicit solution to the investment allocation problem if the unreliability of each stage decreases exponentially and continuously as its cost increases. However, the validity of the assumptions in the second approach is questionable. Bellman, Dreyfus, and Kettelle assume a one-out-of-n configuration which is not physically realizable.

Gordon (7) treats optimum component redundancy for maximum system reliability in series-parallel configurations and considers optimization in the presence of constraints such as cost, weight, and power. But ideal models have been assumed, and the effect and reliability of the decision element are neglected as usual. Barlow and Hunter (2) also treat optimization in series-parallel configurations, utilizing the Lagrange multiplier technique. Herron (9) utilizes the Lagrange multiplier approach in optimizing tradeoffs of reliability versus weight. In any reliability optimization process, figures of merit are very important; i. e., system optimization must take place with respect to a particular system parameter. For example, it may be desired to obtain the maximum gain in system reliability with respect to system cost. Nathan (13) discusses a generalized figure of merit which is applicable to a wide variety of applications. He is primarily concerned with optimizing system performance, whatever it may be, with respect to system cost. In the investigation herein, criteria functions, which serve the same purpose as the figures of merit, will be developed and discussed.

Perhaps, particular mention should be made of Sasaki's (15, 16) work in the area of optimizing system reliability in the presence of constraints. Sasaki proposes a decision algorithm to optimize a system utilizing parallel redundancy where only one module must be functional. In particular, he proposes adding a module, one at a time, to the redundant stage which has the greatest failure probability. This process is continued until either the constraint condition has been reached or until the desired reliability goal is achieved. However, he does not prove that the decision algorithm will result in the most economical system. It is shown in the investigation herein that Sasaki's algorithm is a special case of the more generalized criterion function $(\Delta P)_{max}$, where ΔP is the gain in system reliability resulting from adding a module to a particular stage. Sasaki's algorithm is, therefore, not applicable to all redundancy configurations. It is important to note that optimization will depend directly on the criterion function utilized, In this investigation, the criterion function $\left(\frac{\Delta P}{\Delta C}\right)_{max}$ (i.e., ratio of gain in system reliability to increase in system complexity) is recommended and is compared to the criterion function $(\Delta P)_{max}$.

The vast amount of literature available on the subject of redundancy is generally deficient in the following areas: (1) adequate consideration has not been given to the decision element either in the reliability or the optimization model, and (2) a one-out-of-n configuration is often assumed which is not physically reliable due to the lack of a generalized decision element. The intent of the investigation herein is to eliminate, insofar as possible, these deficiencies.

1.2 STATEMENT OF THE PROBLEM

Numerous redundancy configurations have been proposed which, under certain conditions, may be used to increase system reliability. Majority logic, duplexing, quadruplexing, and, in general, requiring only one-element-out-of-n parallel elements to be functional are examples of configurations which have been considered and proposed. Duplex and quadruplex configurations can only be used in very special applications. Presently, there is no known design which is suitable for a decision element in generalized parallel redundancy where only one unit out of n is required to be functional. Therefore, this type of configuration appears only in mathematical models as a figment of imagination and is not physically realizable. To date, majority logic probably has been the most widely used approach and still offers considerable promise in digital applications. It can also be adapted to analog systems; however, the feasibility of the adaptation has not been firmly established.

The basic problem in this investigation is to develop a generalized redundancy configuration which will yield ultrareliability and which is physically realizable. A necessary and important aspect of this problem is the logic design of a decision element which provides fault masking, failure detection, isolation, and module switching. After this problem has been addressed, system design optimization utilizing the proposed technique will be studied. System design optimization entails methods and procedures for segmenting or subdividing a non-redundant system. Also included in the optimization process is the method in which these segments or modules are made redundant; i.e., the degree of redundancy applied to each module to maximize reliability within given constraints.

The concept of a generalized parallel configuration where two, as opposed to only one, of the parallel units are required to be functional for correct operation is proposed herein as a method of meeting the objectives of the basic problem and is derived from the majority logic technique. However, since the term "majority logic" is no longer descriptive of the system under study, the term "two-out-of-n" will be utilized. The configuration is general in that theoretically there are no restrictions on n except $n \geq 3$. Logic can be designed for a particular n and then be projected and derived as a function of n .

Because the approach to generalized redundancy is derived from majority logic, a thorough discussion of majority logic is given in Chapter II. A decision element which can be used with that configuration is developed and, although only voting or fault masking is required, failure detection, isolation, and module switching are covered for two reasons: (1) they are basic to the development in Chapter III and (2) when they are incorporated, the potential of majority logic is extended tremendously; i.e., when automatic failure detection and isolation is used with manual replacement. Finally, in keeping with the overall approach, system design optimization utilizing majority logic is discussed.

Chapter III treats the generalized two-out-of-n configuration where n is arbitrary but must be equal to or greater than three. It is shown that the redundant system has the greatest reliability for a given complexity when a nonredundant system is divided into modules of equal reliability and when equal degrees of redundancy are applied to each of these modules. This result is then utilized to show that a reliability as close to unity as desired can be obtained with the proposed approach. System complexity utilizing this method is also determined. Throughout this chapter, it has been assumed that a decision element is used with each module in the system. Although a single decision element per redundant stage is possible, multidecision elements eliminate the possibility of single point failures.

In any practical application, it may not be possible to divide a nonredundant system into modules of equal reliability. If this is the case, it also follows that the degree of redundancy applied to each module need not necessarily be the same. A new problem is encountered if the degree of redundancy of each module is different; namely that of interconnecting the n_i outputs from one redundant module to the n_j inputs of the next module. This problem is discussed in Chapter IV and a method of solution is proposed.

Optimization of real systems is investigated in Chapter IV where two criteria functions and decision algorithms are developed and applied to a hypothetical system. Initially, for the sake of simplicity, consideration is not given to the incorporation of the decision element. Later, however, it is shown how the initial development can be modified to include this element, and the previous example is revisited for this purpose.

The results and their usefulness, in any particular application, to a great extent depend upon the assumptions which have been made. These assumptions, in many respects, are analogous to axioms which are basic and from which mathematical theory is developed; if the axioms, or assumptions, are not applicable to a particular situation, then the theory and results which follow are of little value. Some of the assumptions on which this investigation is based are as follows:

1. Failures are independent. Redundancy techniques of any sort are of little value if this assumption is not applicable.

2. The techniques developed are primary applicable to digital circuits where outputs are in discrete form. Thus the output is a logical "1" if the output voltage is high, and a logical "0" if the output voltage is low when positive logic is utilized, and vice versa when negative logic is used. Intermittent failures are possible and are taken into consideration. Although the technique investigated is primarily for digital application, theoretically, there is no reason why it cannot be

adapted for continuous or analog systems when suitable analog-to-digital and digital-to-analog converters have been used.

3. The techniques studied are applicable to both "powered-off" and "powered-on" standby units. Powered-off standby units will probably yield higher reliability; however, it is possible that a switching sequence would be required before they are actively employed in the system. The technique proposed allows sequencing of powered-off standby units with little adverse effect.

4. Output voting, as opposed to input voting, is assumed. Thus, it is assumed that the signals entering the system are correct. This assumption places no limitations on the technique which is equally applicable to input voting.

5. Component parts, circuits, and modules are assumed to obey the exponential failure law. Certain assumptions are implicit when this law is assumed and may be found in any good textbook on probability theory.

6. The reliability of a simplex component, circuit, and module is assumed to be a function of the number of components under consideration and their average failure rate and operating time. Interconnections, such as solder on weld joints, are not included. However, the techniques proposed allow for the inclusion of the interconnections if so desired. Although discrete component parts have been assumed,

the number of gates employed on a chip, or the number of chips utilized in a system, could be readily used in the analysis in case of large-scale integrated circuit implementations.

1.3 METHOD OF SOLUTION AND RESULTS

The basic problem consists of developing a generalized approach to parallel redundancy which is physically realizable and which can be utilized to obtain ultrareliable systems; then this approach is used to determine how a system should be organized, either to yield maximum reliability within given constraint conditions or to meet a given reliability goal utilizing a minimum amount of resources.

It is shown that ultrareliability can be achieved by utilizing a two-out-of-n redundancy configuration as opposed to the one-out-of-n configuration most frequently considered. Although the one-out-of-n configuration theoretically yields greater reliability than a two-out-of-n configuration, the generalized approach to the one-out-of-n arrangement is not physically realizable. (The ratio of failure probability of a two-out-of-n to a one-out-of-n configuration is given by $\frac{n}{\overline{R}} - (n-1)$, where n is the number of parallel elements per module and $\overline{R}$ is the failure probability of a nonredundant module. This expression is always greater than 1 since $\overline{R} < 1$.) Thus, the primary reason for selecting a two-out-of-n form of redundancy is that it is possible to design a decision element which can be used with this configuration in general, and this configuration yields the highest reliability possible next to the one-out-of-n configuration. The design and development of the decision element, which detects and isolates failures, masks errors, and switches to functional operational units as failures are detected, are a major aspect of this investigation. The feasibility of the design of the decision element proposed to satisfy the functional requirements has been established through the construction and operation of a demonstrational breadboard. The breadboard, which accommodates up to 10 inputs, functions as expected and predicted. From the logical design of the decision element, it is possible to project the design complexity and thus the effect upon system reliability for an arbitrary number of inputs. It is also shown how a nonredundant system should be divided into modules to obtain maximum system reliability when redundancy is applied to the modules. To achieve maximum reliability, a nonredundant system should be divided into modules of equal reliability, and equivalent degrees of redundancy should be applied to each of these modules. When the system is organized in an optimum manner, and when a decision element is used with each module, it is shown that system reliability as close to unity as desired can be obtained. Overall system complexity can also be readily determined and predicted. The availability of resources is the only factor which limits the reliability that can be obtained.

In a practical application, it may not be possible to divide a system into portions each consisting of the same reliability. If this cannot be accomplished, then it is no longer desirable to apply equal degrees of redundancy to the modules. Utilizing different degrees of redundancy within a system creates the additional problem of interconnecting or interfacing n_i outputs from one redundant stage to the n_j inputs of the next stage. The interconnection would be no problem if a single decision element as designed herein were used between stages; however, the possibility of single-point failures would have been introduced into the system. Both the interfacing problem and the possibility of single-point failures can be eliminated by utilizing majority logic in the decision element.

Methods and techniques are investigated which can be employed in the optimization of a practical system when consideration has been given to constraints in system design parameters. Two criteria functions are developed which are used in the decision algorithm in the system optimization process. The optimization process is an iterative process and consists of adding an additional module to the ith stage according to a decision algorithm or criterion function. (Since it is assumed that if redundancy is used, it will be of degree three or greater, the initial allocation to each nonredundant module is two additional parallel elements; thereafter, only one element is added at a time.) In particular, the two criteria functions or decision algorithms which are derived and discussed in detail are as follows: (1) modules are added in such a manner to maximize the gain in overall system reliability, and (2) modules are added in a manner to maximize the ratio of gain in system reliability to the increase in system complexity. It is shown that the second method leads to a system of maximum reliability with the expenditure of a minimum amount of resources.

The results of this investigation are significant because for the first time a method is developed for theoretically obtaining and physically realizing ultrareliability in a generalized parallel redundant configuration. In this work, unlike much of the effort which has been expended in the past, the theoretical development and the practical aspects of realizability have been considered of equal importance and treated accordingly. Therefore, it is sincerely hoped that the results of this effort will be beneficial to the engineering field, in particular, and to mankind as a whole. However, it should not be considered as a means to an end, but rather as a stepping stone from which to proceed.

CHAPTER II

MAJORITY LOGIC REDUNDANCY OF DEGREE THREE

2.1 INTRODUCTION

This chapter develops the foundation from which a more generalized treatment may be pursued in Chapter III. Majority logic consisting of three parallel units of which only two must be functional for successful system operation will be of primary concern here. Logic will be developed for fault masking, failure detection, and failure isolation, and consideration will be given to system optimization with this and other designs.

The concept of majority logic is not new, having been proposed as early as 1956 by von Neumann as a means of masking failures; however, the additional features of failure detection and isolation have not been investigated in as great a detail. If failure detection and isolation can be satisfactorily accomplished, these techniques may be used to increase reliability; a great potential also exists for reducing or eliminating maintenance cost, troubleshooting time, equipment downtime, etc.

The term "majority logic" as used in this chapter will be limited to a serial-parallel configuration such as that shown in Figure 2.1.1, in

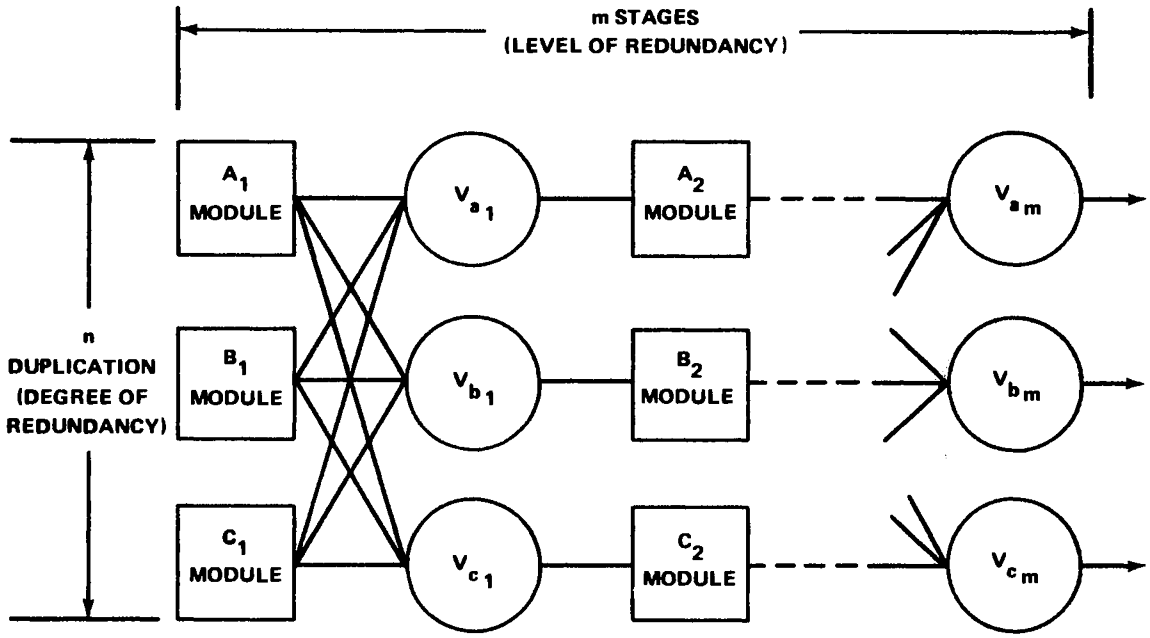


Figure 2.1.1. Two-Out-of-Three Majority Logic With Output Voting

which two-out-of-three redundant elements must be functional to obtain a correct output. This will be referred to as redundancy of degree three. The level of redundancy (i.e., the number of modules into which a nonredundant system is divided) will be optimized with respect to the decision element or voter. To accomplish this, it is necessary to develop the logical design for the decision element. It is also sometimes possible to obtain a correct output even when two modules or voters in the same stage have failed. This can occur if failures in the same stage are in opposite directions such that the third output must always agree with at least one of the failed unit's output. Initially, this will not be considered; however, the development will later be modified to take this into consideration. The true reliability will be bounded within these limits. The value that one wishes to use depends on the application, personal taste, and conservatism. It is emphasized throughout the entire investigation that the major concern is techniques leading to relatively high reliability when compared to a nonredundant system rather than an absolute estimate of reliability, although naturally one can possibly lead to the other.

In considering the two-out-of-three (i.e., degree three) approach, two methods of decision making, or voting, are possible; input voting and output voting. Output voting is shown in Figure 2.1.1. Input voting is illustrated in Figure 2.1.2. Notice that the essential difference in these two figures is the first set of voters; i.e., output voting is basically input voting if it can be assumed that the signals entering the system are correct. This may be a trivial point; however, Ergott and Rozenberg (14) have shown that output voting is always superior to input voting. As system size increases in the limit, the two methods yield equivalent results. Output voting will be assumed in the development herein, with the primary concern being relative reliability improvement. Karyl J. Gurzi (8) has treated the application of three versus a single voter between the redundant stages. To eliminate single point failures, three voters will be assumed in the work herein. However, similar design and analysis would be applicable to a single voter. A single decision element may be required at the last section. If a single signal is required in the next system, rather than carrying the three redundant signals on to the next

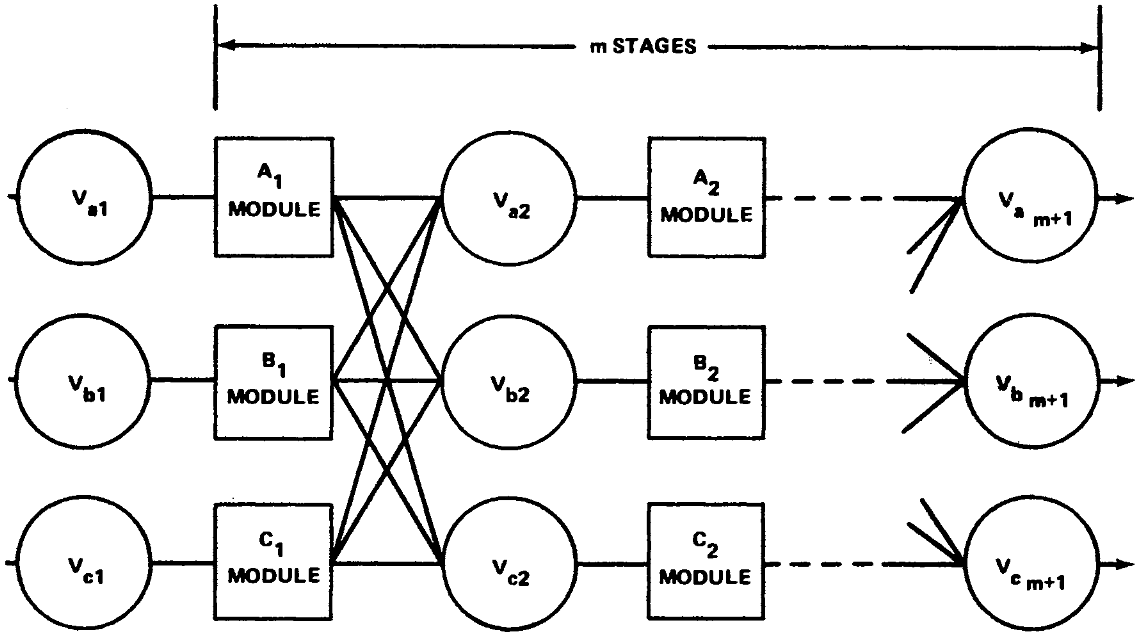


Figure 2.1.2. Two-Out-of-Three Majority Logic With Input Voting

system, this would become a requirement. As will be shown later, this may really be the limiting factor in reliability improvement.

Three axioms of probability theory which will be useful in the development of the reliability equations are:

1. If p denotes the probability that an event will occur, then $1 - p$ denotes the probability that the event will not occur.

2. If the events $\xi_1, \xi_2 \ldots \xi_n$ are independent events with probabilities $p_1, p_2, \ldots p_n$, respectively, then the probability that all of the events should occur simultaneously when all are in question is the product of the probabilities

$$P = \prod_{i=1}^{n} p_i \quad .$$

3. If the probability of mutually exclusive events $\xi_1, \xi_2 \ldots \xi_n$ is $p_1, p_2, \ldots p_n$, respectively, then the probability that any one of these events should occur when all are in question is the sum of the probabilities

$$P = \sum_{i=1}^{n} p_i \quad .$$

In the development which follows, these axioms will be assumed to be understood and will be referred to only rarely.

To apply the exponential distribution to either the system, subsystem, or component level, the following assumptions are necessary:

1. Component failures are independent and random.

2. The component failure rate, λ, is constant over the time frame being considered. This effectively assumes adequate burn-in and screening of components.

3. Components are not subject to wear or fatigue. Thus, the analysis is restricted to electrical components, and mechanical systems are not included except under unusual cases. (If mechanical components are replaced, thus circumventing a failure caused by wear, other type mechanical failures might possibly be considered as being random.)

The term "module" will be used to describe the number of subsystems or elements into which a simplex or nonredundant system has been divided and will be denoted by m . For simplicity, it will be assumed that the modules have the same number of component parts and thus the same reliability. For a system of given complexity, m may be said to represent the level of redundancy which is to be used. It will be shown in Chapter III that dividing a system in modules of equal reliability leads to the greatest reliability improvement when the modules are replicated. After a nonredundant system has been divided into m modules of equal reliability, each module is replicated n times and is then called a redundant module or simply a stage. Thus, n represents the degree of redundancy applied to each stage or to the system as a whole. Chapter III will also show that stages processing equivalent degrees of redundancy lead to maximum system reliability. For the purpose of this chapter, n will be restricted primarily to three; however, the reliability equations will be derived in general terms so that they may be used in Chapter III. The failure probability (unreliability) and success probability (reliability) of a nonredundant module will be denoted by $\overline{R}_m$ and R_m, respectively, while that of a redundant module or stage is $\overline{P}_m$ and P_m, respectively. The failure probabilities of a nonredundant and redundant system will be represented by $\overline{R}_s$ and $\overline{P}$, respectively. R may be used for different purposes; however, it will generally be used to denote the product of a module and decision element reliabilities.[1]

---

[1]Depending upon the circumstances, it is sometimes more convenient to deal with success probabilities and sometimes more desirable to use failure probabilities. Throughout this investigation, axiom (1) will be assumed to be understood and transformation between success and failure probabilities will be made as convenient.

2.2 DERIVATION AND OPTIMIZATION OF THE RELIABILITY EQUATION

The failure probability of a single redundant module containing n parallel elements in which two or more units must be functional such as shown in Figure 2.1.1 can be found from the binomial distribution and is given by the expression

$$\overline{P}_m = \sum_{i=n-1}^{n} \binom{n}{i} R^{n-i}(1-R)^i \qquad (2.2.1)$$

where $\binom{n}{i}$ denotes $\frac{n!}{i!\,(n-i)!}$ which represents the number of combinations of n things taken i at a time.[2] For the binomial distribution to be appropriate the following conditions must be fulfilled:

1. There exist n independent trails; i.e., the outcome of any trial is not dependent on those preceding it. (A trial here is assumed to be the operation of an element, usually a module, over a given period of time; the outcome is determined by the success or failure of the module.)

2. The experiment is dichotomous; i.e., there are only two possible outcomes at each trial. For the purposes herein, the possible

---

[2]There are many ways in which this can be viewed and derived. With axiom (1), this can be put in another form. Also, truth tables can be used to derive the binomial distribution and thus this expression. Moskowitz (12) uses flow graphs and networks to derive and manipulate reliability equations.

outcomes are only success and failure.

3. The probability of any particular outcome at any trial remains constant through the experiment.

Equation (2.2.1) can be expressed in the expanded form:

$$\overline{P}_m = \binom{n}{n-1} R(1-R)^{n-1} + \binom{n}{n}(1-R)^n$$
$$= (1-R)^{n-1}\,[1 + (n-1)R] \tag{2.2.2}$$

or it may be alternately represented by

$$\overline{P}_m = \overline{R}^{n-1}\,[n - (n-1)\overline{R}] \quad . \tag{2.2.3}$$

In Equation (2.2.2), R represents the product of the reliability of the module and the decision element, since as many decision elements are to be employed as there are modules (Fig. 2.1.1). Thus, in Equation (2.2.2) or (2.2.3), the module and decision element may be considered to be lumped together; i.e.,

$$R = R_m R_v \quad . \tag{2.2.4}$$

In Equation (2.2.4), R_m and R_v are the reliability of the module and decision element or voter, respectively. From axiom (1), it follows that

$$\overline{R} = 1 - \left(1-\overline{R}_m\right)\left(1-\overline{R}_v\right) = \overline{R}_m + \overline{R}_v - \overline{R}_m\overline{R}_v \quad . \tag{2.2.5}$$

Substitution of Equation (2.2.5) into (2.2.3) yields the following relationship for failure probability of a redundant module or stage:

$$\overline{P}_m = \left(\overline{R}_m + \overline{R}_v - \overline{R}_m\overline{R}_v\right)^{n-1} \left[n - (n-1)\left(\overline{R}_m + \overline{R}_v - \overline{R}_m\overline{R}_v\right)\right] \quad . \tag{2.2.6}$$

It is desired to determine the system organization which yields optimality; i.e., how should a nonredundant system be subdivided to optimize (minimize) the overall redundant system failure probability? If $\overline{R}_m\overline{R}_v$ is small compared to $\overline{R}_m$ and $\overline{R}_v$, the overall redundant system failure probability may be approximated by

$$\overline{P} \approx m\left(\overline{R}_m + \overline{R}_v\right)^{n-1} \left[n - (n-1)\left(\overline{R}_m + \overline{R}_v\right)\right] \tag{2.2.7}$$

where m represents the number of modules into which a nonredundant (simplex) system has been divided and n the degree of redundancy applied at each stage, which for the purposes of Chapters II and III has been assumed to be the same for all stages. For Equation (2.2.7) to be valid or a good approximation, the cross terms or second-order terms must be small in comparison to the first-order terms.

Much difficulty is encountered if an attempt is made to use classical techniques to optimize this equation; i.e., to take the first partial derivatives, set them equal to zero, and solve for the variables; then take the second partial derivatives to test for minimum-maximum conditions. In the first place, four variables are present such that a complex relationship is obtained when the partial derivatives are taken and set equal to zero. Much simplification is possible, however, if the ratio of the unreliability of a nonredundant module to that of a redundant module is considered. This ratio will be denoted by β and is given by the relationship

$$\beta = \frac{\overline{R}_m}{\overline{P}_m} = \frac{\overline{R}_m}{\left(\overline{R}_m + \overline{R}_v - \overline{R}_m\overline{R}_v\right)^{n-1}\left[n - (n-1)\left(\overline{R}_m + \overline{R}_v - \overline{R}_m\overline{R}_v\right)\right]} \quad . \tag{2.2.8}$$

If maximum reliability is gained at each stage, it follows that maximum gain in system reliability results. $\overline{R}_v$ will be taken as being fixed since it requires a given number of component parts to accomplish the decision element function; for the purposes of this chapter, n will be taken as fixed at $n = 3$. A general treatment will be considered in Chapter III. Under these assumptions, Equation (2.2.8) takes the form

$$\beta = \frac{\overline{R}_m}{3\left(\overline{R}_m + \overline{R}_v - \overline{R}_m\overline{R}_v\right)^2 - 2\left(\overline{R}_m + \overline{R}_v - \overline{R}_m\overline{R}_v\right)^3} \quad . \tag{2.2.9}$$

Differentiating Equation (2.2.9) with respect to $\overline{R}_m$ and setting the result equal to zero yields

$$\frac{\partial\beta}{\partial\overline{R}_m} = \frac{N}{D}$$

where

$$N = 3\left(\overline{R}_m + \overline{R}_v - \overline{R}_m\overline{R}_v\right)^2 - 2\left(\overline{R}_m + \overline{R}_v - \overline{R}_m\overline{R}_v\right)^3$$

$$- \overline{R}_m \left[6\left(\overline{R}_m + \overline{R}_v - \overline{R}_m\overline{R}_v\right)\left(1 - \overline{R}_v\right) - 6\left(\overline{R}_m + \overline{R}_v - \overline{R}_m\overline{R}_v\right)^2\left(1 - \overline{R}_v\right)\right]$$

and

$$D = \left[3\left(\overline{R}_m + \overline{R}_v - \overline{R}_m\overline{R}_v\right)^2 - 2\left(\overline{R}_m + \overline{R}_v - \overline{R}_m\overline{R}_v\right)^3\right]^2 \quad . \tag{2.2.10}$$

Multiplying by the denominator and dividing by $\overline{R}_m + \overline{R}_v - \overline{R}_m\overline{R}_v$ yields

$$3\left(\overline{R}_m + \overline{R}_v - \overline{R}_m\overline{R}_v\right) - 2\left(\overline{R}_m + \overline{R}_v - \overline{R}_m\overline{R}_v\right)^2 - 6\overline{R}_m\left(1 - \overline{R}_v\right)$$

$$+ 6\overline{R}_m\left(\overline{R}_m + \overline{R}_v - \overline{R}_m\overline{R}_v\right)\left(1 - \overline{R}_v\right) = 0 \quad . \tag{2.2.11}$$

Since second- and higher-order terms are relatively small and may be neglected without appreciable error, Equation (2.2.11) becomes

$$3\overline{R}_m + 3\overline{R}_v - 6\overline{R}_m = 0$$

$$\overline{R}_m = \overline{R}_v \quad . \tag{2.2.12}$$

The fact that this leads to a maximum reliability gain rather than a minimum or an inflection point will not be covered in more detail here, but will be covered in general in Chapter III. Detailed numerical examples will also be given there to demonstrate that this indeed represents an optimum design; thus, the minimum failure probability of a single redundant module is given by the expression

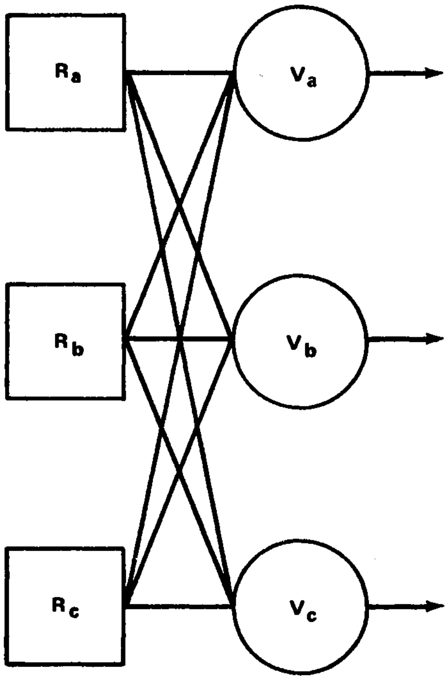
$$\overline{P}_m = 3\left(2\overline{R}_m - \overline{R}_m^{\,2}\right)^2 - 2\left(2\overline{R}_m - \overline{R}_m^{\,2}\right)^3 \tag{2.2.13}$$

when the system is organized in the optimum manner.

2.3 CONSIDERATION OF FAILURES IN OPPOSITE DIRECTIONS CANCELLING

When the majority logic or two-out-of-three technique is used, some advantage can be taken by noting that failures in opposite directions can cancel each other, in which case, only one module, rather than two, is required to the functional. To indicate how the failure probability expression can be derived under these conditions, Figure 2.3.1 will be helpful.

F and S (Fig. 2.3.1) indicate failures and successes, respectively. The remarks under system status are applicable to the situation where two-out-of-three modules must be good; i.e., they do not consider the possibility of failures in the opposite direction cancelling. If it is assumed that $R_a = R_b = R_c$ which is valid since they are identical modules as far as possible, combinations 1, 2, 3, and 5 in Figure 2.3.1 result in a failed system given by the expression



COMBINATION NUMBER	R_a	R_b	R_c	SYSTEM STATUS
1	F	F	F	FAILED
2	F	F	S	FAILED
3	F	S	F	FAILED
4	F	S	S	OPERATIVE
5	S	F	F	FAILED
6	S	F	S	OPERATIVE
7	S	S	F	OPERATIVE
8	S	S	S	OPERATIVE

Figure 2.3.1. Block Diagram of a Two-Out-of-Three Majority Logic With Its Truth Table

$$\bar{P}_m = \bar{R}^3 + 3\bar{R}^2 R = \bar{R}^3 + 3\bar{R}^2\left(1-\bar{R}\right) = 3\bar{R}^2 - 2\bar{R}^3 . \tag{2.3.1}$$

This is equivalent to Equation (2.2.3) which was derived directly from the binomial distribution. Notice that in the second combination of the truth table, shown in Figure 2.3.1, the system would not have failed if R_a had failed to a logical "0" and R_b to a logical "1" or vice versa. This may be expressed in the Boolean form

$$\bar{R}_{a0} \cdot \bar{R}_{b1} \cdot R_c + \bar{R}_{a1} \cdot \bar{R}_{b0} \cdot R_c$$

where the second subscript indicates failure mode. Since this condition can occur in three ways (combinations 2, 3, and 5 in Figure 2.3.1), the reliability gained by taking into consideration the possibility that failures can cancel is

$$P_g = \overline{R}_{a0} \cdot \overline{R}_{b1} \cdot R_c + \overline{R}_{a1} \cdot \overline{R}_{b0} \cdot R_c + \overline{R}_{a0} \cdot R_b \cdot \overline{R}_{c1}$$

$$+ \overline{R}_{a1} \cdot R_b \cdot \overline{R}_{c0} + R_a \cdot \overline{R}_{b0} \cdot \overline{R}_{c1} + R_a \cdot \overline{R}_{b1} \cdot \overline{R}_{c0} \quad . \tag{2.3.2}$$

The total probability of a failure is the sum of the probabilities of component failures to a "0" state and to a "1" state; thus, $\overline{R} = \overline{R}_0 + \overline{R}_1$. Without further knowledge of a specific application or circuit, there is no reason to suspect a failure to any particular state to be more predominant than to the other state; consequently, $\overline{R}_0 = \frac{1}{2}\overline{R}$ and $\overline{R}_1 = \frac{1}{2}\overline{R}$. This leads to the conclusion that $\overline{R}_0 = \frac{1}{2}(1-R)$ and $\overline{R}_1 = \frac{1}{2}(1-R)$. Substituting these values into Equation (2.3.2) yields the reliability gained from consideration of failures in opposite directions which is given by

$$P_g = 6R\left[\frac{1}{2}(1-R)\right]^2 = \frac{3R}{2}\left[1 - 2R + R^2\right] \quad . \tag{2.3.3}$$

Thus, the reliability of one module of a majority logic system when the possibility of failures cancelling has been taken into consideration is given by

$$P_m = (3R^2 - 2R^3) + \left(\frac{3R}{2} - 3R^2 + \frac{3}{2} R^3\right) = \frac{1}{2} (3R - R^3) \qquad (2.3.4)$$

and the failure probability is given by[3]

$$\overline{P}_m = \frac{1}{2} (3\overline{R}^2 - \overline{R}^3) \quad . \qquad (2.3.5)$$

Since it has been shown that

$$\overline{R} = \overline{R}_m + \overline{R}_v - \overline{R}_m \overline{R}_v$$

and that when the system is organized optimally, $\overline{R}_m = \overline{R}_v$, Equation (2.3.5) may be expressed as

$$\overline{P}_m = \frac{1}{2} \left[3\left(2\overline{R}_m - \overline{R}_m{}^2\right)^2 - \left(2\overline{R}_m - \overline{R}_m{}^2\right)^3 \right] \quad . \qquad (2.3.6)$$

The actual value of the failure probability for a majority logic module lies somewhere between the values obtained from Equations (2.3.1) and (2.3.6); the choice of which is used depends upon the amount of conservatism one wishes to include. However, it is noted that Equation (2.3.6) yields almost one-half the failure probability that Equation (2.3.1) yields. The possibility that failures can cancel will not be discussed further in this work. Again, major emphasis is not upon reliability prediction, but rather

---

[3] It should be noted that in general it is not possible to obtain the expression for the failure probability by simply replacing R in the reliability expression with $\overline{R}$. This is possible for a two-out-of-three system since two operational units result in an operational module and two failed units result in a failed system. (See the truth table of Figure 2.3.1.)

upon techniques which lead to highly reliable systems and system organization to accomplish this purpose.

2.4 LOGIC DESIGN FOR FAILURE DETECTION, ISOLATION, AND FAULT MASKING

In practice, if a system is to be optimized, the procedure which should be used is as follows: Develop the decision element logic design and estimate its failure probability. Subdivide the nonredundant system into m modules, each of which has a failure probability equal to that of the decision element. Since the decision element and the problem of fault masking, failure detection, and isolation play such a vital role in system organization, it is logical to address this aspect next.

The decision element can be designed for two different purposes depending upon application. In one case, it may only be necessary to fault mask failures and not be concerned about failure detection and isolation. Such may be the case if nothing can be done about the failures once they have occurred; i.e., repair and replacement are not feasible. Only one failure per stage is permissible, the module and decision element being regarded as an integral part of the module. On the other hand, if a failed module can be replaced either manually or automatically, then automatic failure detection is very desirable and could lead to a reliability limited only by the spare parts available and possibly also could result in potential cost savings in troubleshooting, repair, periodic maintenance, equipment downtime, etc. Automatic failure detection and isolation, although not having been given adequate attention in the past, possess a tremendous potential in certain applications. For example, failure detection and isolation may not be worthwhile in realtime missile systems where repair and replacement are not possible; on the other hand, it may be very desirable and beneficial in a commercial computer system where repair and replacement are permissible.[4] In the past, primary emphasis has been given to reliability improvement alone without replaceable items; however, in the future, when maintenance, system downtime, etc., are taken into consideration, automatic failure detection and isolation as well as fault masking could become very important. The techniques proposed herein become of interest when viewed from this standpoint and are very likely to receive much more attention in the future. The technological growth in electronic elements may reduce circuit costs below maintenance cost, downtime, etc., making redundancy attractive when viewed from a cost standpoint alone.

The logical design of a decision element or "voter" whose output represents the majority of the inputs is not particularly new and may be found in Shooman (17) as well as other sources. Table 2.4.1 shows that an output is desired for the following conditions:

---

[4]Manual repair and replacement may be possible in earth orbital space stations and interplanetary manned missions. In fact, this may be the only means of obtaining satisfactory reliability over the desired time frame. The example given here is meant to apply primarily to boost and reentry phases of flight.

$$f_1 = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC = BC + AC + AB \qquad (2.4.1)$$

TABLE 2.4.1

TRUTH TABLE FOR LOGIC DECISION ELEMENT

A	B	C	Desired Output f	Error Conditions
0	0	0	0	None
0	0	1	0	f_C
0	1	0	0	f_B
0	1	1	1	f_A
1	0	0	0	f_A
1	0	1	1	f_B
1	1	0	1	f_C
1	1	1	1	None

Thus, an output of a logical "1" is desired when any two or all inputs are logical "1's." The gating necessary to accomplish this function is shown in Figure 2.4.1. This figure shows a very simple circuit consisting of only three AND gates and an OR gate. From this, it may be concluded that a very low level of redundancy can be applied to a system if design optimality is the objective; i.e., a nonredundant system can be subdivided into m modules, each equivalent to only four gates such that the condition $\overline{R}_m = \overline{R}_v$ is met.

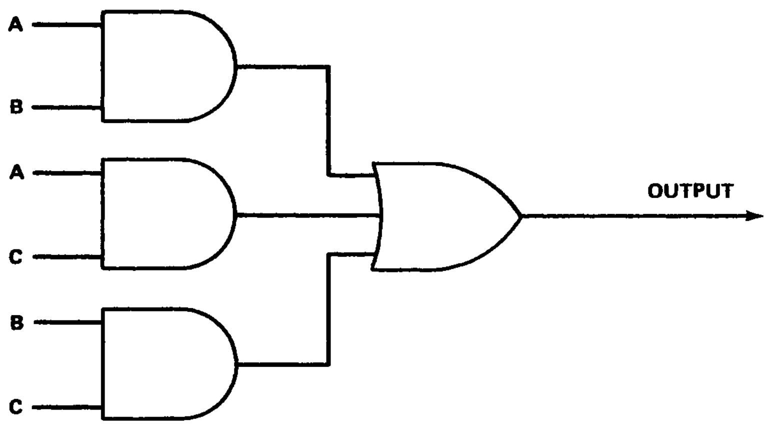


Figure 2.4.1. Fault Masking Logic

The logic element described in the previous paragraph serves only the function of failure masking. Failure detection and isolation may also be of interest as previously indicated. For failure detection, a logic element is desired which provides an output under the condition

$$f_1 = \overline{ABC + \overline{A}\,\overline{B}\,\overline{C}}$$

$$f_1 = (\overline{A} + \overline{B} + \overline{C})\ (A + B + C) \qquad . \tag{2.4.2}$$

Several different equivalent Boolean expressions may be derived to represent this function, such as

$$f_1 = A\overline{C} + A\overline{B} + \overline{A}B + \overline{A}C \qquad . \qquad (2.4.3)$$

The logic necessary to implement this function is shown in Figure 2.4.2 and consists of four AND gates and an OR gate.

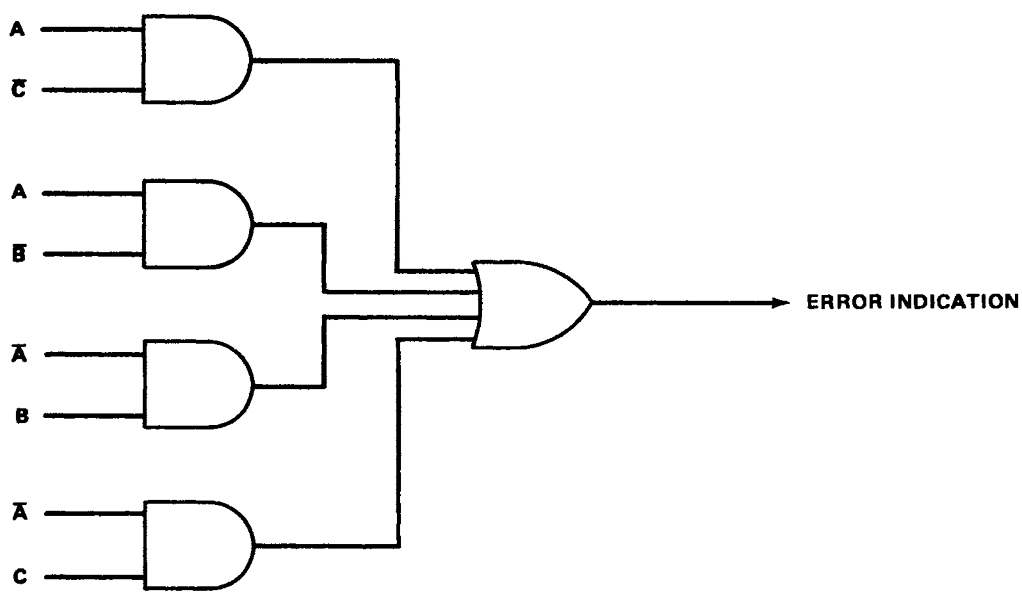


Figure 2.4.2. Failure Detection Logic

Thus far, a voter whose output represents the majority of the inputs and a failure detector which has an output when a disagreement occurs in the inputs have been developed. It is desirable not only to be able to detect a failure, but also to isolate it to the module so that it might possibly be replaced either manually or automatically. From Table 2.4.1, it is observed that modules A, B, and C have failed under the following conditions:

$$
\begin{aligned}
f_A &= \overline{A}BC + A\overline{B}\,\overline{C} \\
f_B &= \overline{A}B\overline{C} + A\overline{B}C \qquad (2.4.4) \\
f_C &= AB\overline{C} + \overline{A}\,\overline{B}C
\end{aligned}
$$

To implement this function and isolate a failure, only six AND gates and three OR gates, as shown in Figure 2.4.3, are required. It should be noted that the failure detection and isolation logic would not normally be considered as part of the voter failure probability ($\overline{R}_V$) for a two-out-of-three organization and would have no influence on system design optimization since reliability as developed in this chapter is not dependent on these functions. However, whenever automatic repair and replacement are considered, they very definitely play a vital part in system reliability. Such considerations are the subject of Chapter III; however, there it will be considered from a slightly different viewpoint. When the above functions have been incorporated in system design, the following functions can be accomplished:

1. Faults are automatically masked and no single failure will cause a system failure. The number of failures most likely to occur before the redundant system fails is a function of system complexity,

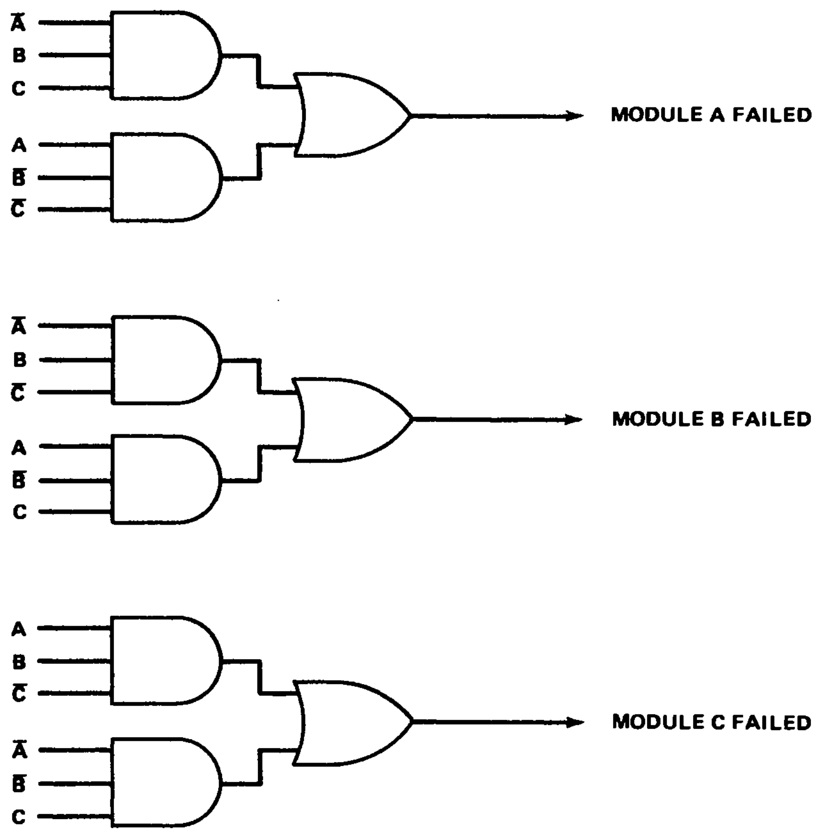


Figure 2.4.3. Failure Isolation Logic

the number of modules into which a nonredundant system is divided, operating time, etc.[5]

2. Automatic failure indication.

3. Failure isolation to the module level.

By utilizing the above approach, it is possible to:

---

[5]See Appendix A for a more detailed treatment of the number of failures most likely to occur.

1. Improve considerably the reliability of a system. For a system with manual replacement, reliability is basically limited only to the supply of spare modules.

2. Delete periodic maintenance requirements.

3. Reduce troubleshooting time, repair time, etc.

4. Eliminate equipment downtime.

5. Possibly minimize spare parts supply.

Much more research is required to determine the tradeoffs in increased hardware cost necessary to accomplish 1., 2., and 3. above and the amount of savings to be realized when these are accomplished. In general, with the type system proposed, Appendix A indicates that there is no great hurry to replace a failed module since the total system is still operational and is likely to remain in that state even after several failures have occurred. The major theme of this investigation is 1. above, so little more will be said about automatic failure detection and isolation when used with manual replacement. However, these techniques and approaches if properly used can also have a tremendous influence on 1. as will be given in Chapter III.

Before proceeding, it is instructive to note that the logic developed for the functions above is not unique and can be implemented in several alternate forms. The method one uses will depend to a very large extent upon the type of logic building blocks available. Another way of implementing the failure detection and isolation is shown in Figure 2.4.4.

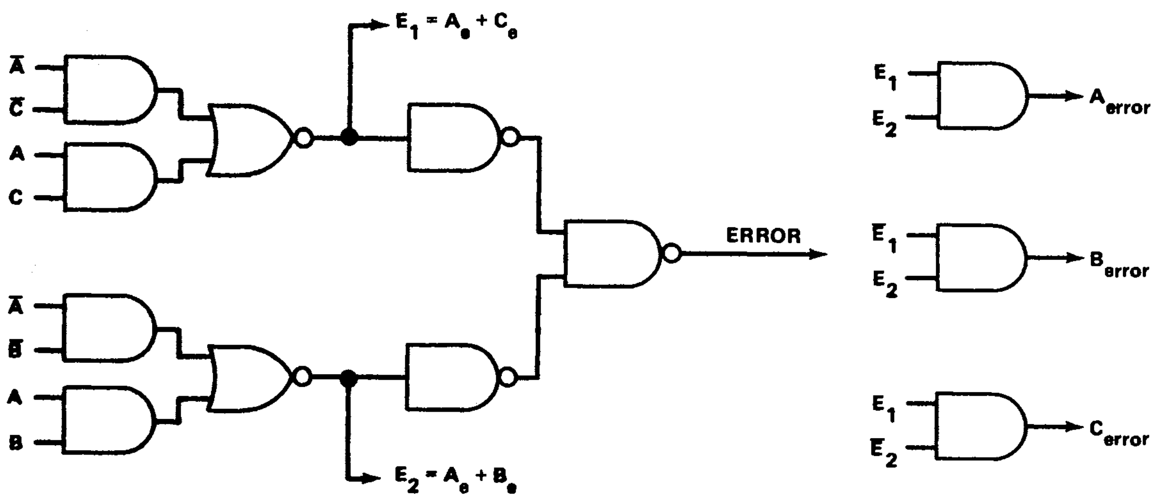


Figure 2.4.4. Alternate Method of Implementing Failure Detection and Isolation Logic.

Since E_1 represents an error either in A or C and E_2 an error in A or B, these can be logically combined to isolate the failure as shown. AND/OR/INVERT and AND/INVERT logic blocks have been used in this implementation. Only 12 gates are required here while 14 were required in the previous implementation. However, five inverters have been added, which may be part of a logic block.

2.5 RELIABILITY GAINED THROUGH REDUNDANCY OF DEGREE THREE

The logic necessary for the decision element has been developed; now, an estimate of system reliability can be obtained by using the majority logic approach with optimum design. The voter consisted of only five gates or approximately 20 discrete component parts, which would indicate that for optimum system design, a module should also contain approximately 20 component parts. For optimality, the failure probability of a module $(\overline{R}_m)$ can be expressed in terms of the failure probability of the total nonredundant system $(\overline{R}_s)$ and the number of modules (m) into which it has been subdivided as follows: From axiom (2), it follows that the reliability of a nonredundant system (R_s) expressed in terms of R_m and m is given by the relationship

$$R_s = (R_m)^m$$

$$\overline{R}_s = 1 - R_s = 1 - (R_m)^m = 1 - (1 - \overline{R}_m)^m \qquad (2.5.1)$$

$$\overline{R}_s = 1 - \left[1 - m\overline{R}_m + \frac{(m)\,(m - 1)\,\overline{R}_m^{\,2}}{2!} - \ldots\right] \quad .$$

Thus, if $\overline{R}_m$ is small, second- and higher-order terms can be neglected in which case the module failure probability can be approximated by

$$\overline{R}_m \approx \frac{\overline{R}_s}{m} \quad . \qquad (2.5.2)$$

Substituting Equation (2.5.2) into (2.3.1) and noting also that $\overline{P} \approx m\,\overline{P}_m$ in a manner similar to that shown above yields the following relationship:

$$\overline{P} \approx m\left[3\left(\frac{2\overline{R}_s}{m} - \frac{\overline{R}_s^{\,2}}{m^2}\right)^2 - 2\left(\frac{2\overline{R}_s}{m} - \frac{\overline{R}_s^{\,2}}{m^2}\right)^3\right] \quad . \qquad (2.5.3)$$

Also, let N_T be the number of component parts in a nonredundant system and n_m be the parts in a module which for optimum design was estimated to be approximately 20 with the above voter design; then

$$m = \frac{N_T}{n_m} = 0.05\ N_T \tag{2.5.4}$$

and Equation (2.5.3) takes the form

$$\overline{P} = 0.05\ N_T \left[3\left(\frac{40\overline{R}_s}{N_T} - \frac{400\overline{R}_s^{\,2}}{N_T^{\,2}} \right)^2 - 2\left(\frac{40\overline{R}_s}{N_T} - \frac{400\overline{R}_s^{\,2}}{N_T^{\,2}} \right)^3 \right] \quad . \tag{2.5.5}$$

The failure probability of a nonredundant system is given by the expression

$$\overline{R}_s = 1 - e^{-N_T \lambda t} \tag{2.5.6}$$

where N_T is the number of components in the nonredundant system, λ the average component failure rate, and t the operating time.[6] Arbitrarily, take a reasonable value of $\lambda = 10^{-8}$ and $t = 10^4$; then Equation (2.5.6) is given by

$$\overline{R}_s = 1 - e^{-10^{-4} N_T} \quad . \tag{2.5.7}$$

---

[6]In Equation (2.5.6), $t = 1/\lambda N_T$ is the mean time between failures (mtbf) of the complete nonredundant system; thus, $\overline{R}_s = 0.632$. In some cases, it would be desirable to normalize about this value; however, if N_T is varied (e.g., increased) since λ is a constant, this effectively alters (decreases) t and makes the given nonredundant system more reliable.

Table 2.5.1 shows $\overline{P}$ and $\alpha = \overline{R}_s/\overline{P}$ for three values of N_T which cover a fairly reasonable range. The last value shown in Table 2.5.1 for $N_T = 100$ k is of particular interest. The failure probability of a simplex system has been decreased from practically 1 to almost 0 through the application of the optimum redundancy organization.

TABLE 2.5.1

TYPICAL SYSTEM PARAMETERS FOR $\lambda = 10^{-8}$ FAILURES PER HOUR AND OPERATING TIME OF $t = 10^4$ HOURS

(i.e., $\lambda t = 10^{-4}$)

N_T	m	$\overline{R}_s$	$\overline{P}$	α
1k	50	0.09500	0.000007	13.57×10^3
10k	500	0.632000	0.000957	660.40
100k	5000	0.999955	0.002340	427.33

The relative complexity of a redundant system when compared with that of a simplex system is given by

$$c = n(1 + a)$$

where n is the degree of redundancy and $a = \frac{n_v}{n_m}$; i.e., the ratio of decision element and module complexity. For optimum design, it has been shown that $a = 1$, and since in this chapter $n = 3$ is assumed, it is noted that the redundant system which is optimally designed will contain approximately six times as many component parts as a nonredundant system. From this, it may be concluded that by utilizing the two-out-of-three majority logic technique, the relative complexity of the redundant system should be no greater than six nor less than three times that of a nonredundant system regardless of optimality considerations. The exact value depends upon the level of redundancy application.

Does the majority logic scheme always improve the reliability of a system? To answer this question, Equation (2.3.1) can be equated to the failure probability of a module and the equation solved for $\overline{R}$. Thus,

$$\overline{R} \geq 3\overline{R}^2 - 2\overline{R}^3$$

$$2\overline{R}^2 - 3\overline{R} + 1 \geq 0 \quad . \tag{2.5.8}$$

$$\overline{R} \geq 1 \ , \ \overline{R} \leq \frac{1}{2}$$

The first case where $\overline{R} \geq 1$ is not physically realizable since $0 \leq \overline{R} < 1$; thus, only $\overline{R} \leq \frac{1}{2}$ yields a reasonable bound. Therefore, majority logic yields a reliability improvement only if

$$\overline{R}_m + \overline{R}_v - \overline{R}_m \overline{R}_v \leq \frac{1}{2} \quad . \tag{2.5.9}$$

This equation also indicates why the last term can be ignored, because it can never be more than roughly 1/16 of the total or approximately 1/8 that of $R_m + R_v$. Intuitively, Equation (2.5.9) is minimized when $R_m = R_v$ as was previously shown. Thus, for optimum design, Equation (2.5.9) becomes

$$2\overline{R}_m - \overline{R}_m^{\,2} \leq \frac{1}{2} \tag{2.5.10}$$

and solving for $\overline{R}_m$ a value of $\overline{R}_m \leq 0.293$ is obtained. The failure probability of the total simplex system has not been restricted by this condition since it is given approximately by

$$\overline{R}_s \approx \sum_{i=1}^{m} \overline{R}_{m_i} = m\overline{R}_m \quad . \tag{2.5.11}$$

Only the failure probability of a module and decision element is restricted to be within these limits.

Notice also that although time does not appear explicitly in the above equations, it nevertheless is included through the relationship

$$\overline{R}_m = 1 - e^{-n_m \lambda t} = 1 - e^{\frac{-N_T \lambda t}{m}} = 1 - e^{\frac{-\lambda_s t}{m}} \quad . \tag{2.5.12}$$

For Equations (2.5.10) and (2.5.12) to be valid

$$t \leq -\frac{m}{\lambda_s} \ln (1 - \overline{R}_m)$$

$$t \leq -\frac{m}{\lambda_s} \ln (0.707) \qquad (2.5.13)$$

$$t \leq \frac{0.342m}{\lambda_s}$$

where λ_s is the failure rate of the nonredundant system, m is the number of modules into which the simplex system has been divided, and t is the operating time. With $t = k/\lambda_s$ (i.e., operating time is k times the mtbf of a simplex machine), then Equation (2.5.13) becomes

$$k \leq 0.342m \quad .$$

For an optimally designed machine, m will be fixed since N_T is fixed and the operating time then must be less than the above constant value for an improvement in reliability.

2.6 SUMMARY

The purpose of this chapter has been primarily to develop the necessary background from which a more generalized analysis can be treated in Chapter III. It has been shown that system reliability can be improved considerably with majority logic techniques, especially when the system is organized in an optimum manner such that $\overline{R}_m = \overline{R}_v$. Consideration has been given to the logical implementation of fault masking, failure detection, and failure isolation. It has been suggested that a tremendous possibility exists when these techniques are incorporated with manual replacement where feasible and it is recommended that further research should be undertaken in this area. The functional relationship between the number of failures which may be expected before a redundant system can be expected to fail was also developed as Appendix A to this chapter.

In Chapter III, the basic approach developed in this chapter will be continued; however, it will be desirable to view the organization from a slightly different standpoint. Although the idea of majority logic will no longer be required, only that two modules in any stage be functional, the similarity to this chapter, both in system approach and logic design development, will become readily apparent.

CHAPTER III

GENERALIZED PARALLEL REDUNDANCY REQUIRING TWO-OUT-OF-n FUNCTIONAL ELEMENTS

3.1 INTRODUCTION

In Chapter II, techniques were developed for fault masking, failure detection, and failure isolation in a major logic, two-out-of-three configuration. It was also mentioned that failure detection and isolation may be used to considerable advantage when combined with manual replacement of modules. It is quite natural to question why they could not also be used for automatic replacement of modules. This basic question is the primary subject of this chapter.

Although the subject of automatic replacement of modules is embedded in the subject of majority logic, it is much more general. The term majority logic is no longer descriptive of the system under study. In general, it is only required that two-out-of-n parallel modules in each stage be functional for correct operation; it is general in another aspect as well. For years now, probability models of parallel units have been studied, usually without regard to the decision or switching element. In the rare cases where consideration has been given to this element, the number of parallel modules used was restricted to a particular configuration. In the investigation herein, a decision and switching element will be developed which can be adapted to any number of n parallel units. The effect of this design upon system reliability will be indicated, and system optimization will be treated taking into consideration the decision element design. The question of practicability will also be covered.

The generalized system to be studied is shown in block diagram form in Figure 3.1.1. The nonredundant system will be divided into m modules of equal reliabilities and replicated n times. It will be assumed that the degree of redundancy (n) of each stage is the same. Initially, it will be assumed that a decision and switching element is provided for each module. This condition will be relaxed in Chapter IV, as well as the condition of equal n for each stage. Thus, failure of a decision element, in effect, appears as if the following module has failed and is compensated for in the following decision element. Essentially, the next decision element in the serial chain corrects for either a preceding decision element or module failure.

3.2 CONSIDERATION OF EQUIVALENT STAGES YIELDING OPTIMALITY

It can be shown that the assumption of breaking a nonredundant system into m identical modules and replicating n times for each stage leads to optimum reliability improvement; i.e., it can be shown that in

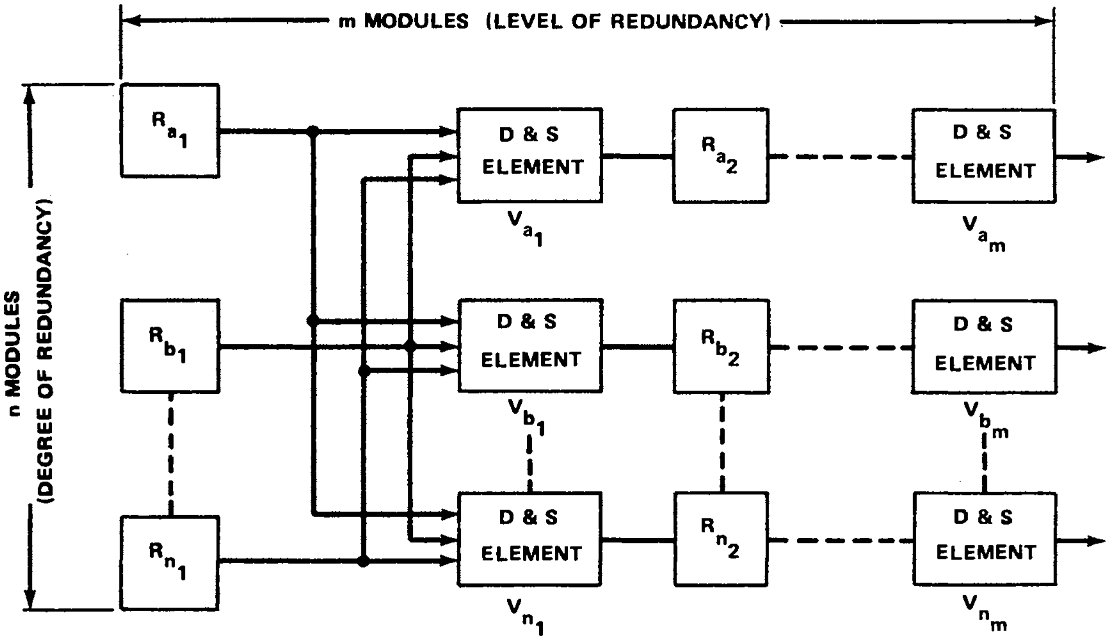


Figure 3.1.1. Generalized Parallel Redundant System

order to obtain maximum reliability, it is necessary that

$R_{a_1} = R_{a_2} = \ldots = R_{a_m}$ and that $n_1 = n_2 = n_3 = \ldots = n_m$. (The first subscript on R is not required here since $R_{a_1} = R_{b_1} = R_{c_1}$, etc., and will be dropped.)

The generalized reliability is a function of R and n as follows:

$$P\left(R_1, R_2, \ldots, R_m, n_1, n_2, \ldots, n_m\right) = f\left(R_1, n_1\right) f\left(R_2, n_2\right) \ldots f\left(R_m, n_m\right) \quad . \tag{3.2.1}$$

Equation (3.2.1) simply states that the overall redundant system reliability which is a function of the reliability of each module and the degree of redundancy applied to each module is given by the product of the reliabilities of the individual stages. It should be noted that the functional forms of the individual stages are the same; this is the reason the notation $f(R_1, n_1) f(R_2, n_2) \cdots f(R_m, n_m)$ is used in lieu of $f(R_1, n_1) g(R_2, n_2) \cdots h(R_m, n_m)$. Further, $R_1 \cdot R_2 \cdots R_m$ is simply the reliability of a nonredundant system and is given by

$$R_1 \cdot R_2 \cdots R_m = R_s \quad . \tag{3.2.2}$$

It will be assumed that the total number of modules will be constrained to K units. In effect, this is constraining the complexity of the system, by assuming a given amount of resources. Thus,

$$n_1 + n_2 + \ldots + n_m = K \quad . \tag{3.2.3}$$

Equation (3.2.1) is to be optimized, subject to the constraints given by Equations (3.2.2) and (3.2.3). If the Lagrange multiplier technique is used, the problem can be formulated as

$$\begin{aligned} P(R_1, R_2, \ldots, R_m, n_1, n_2, \ldots, n_m) = {} & f(R_1, n_1) f(R_2, n_2) \cdots f(R_m, n_m) \\ & + \lambda_1 (R_1 \cdot R_2 \cdots R_m - R_s) \\ & + \lambda_2 (n_1 + n_2 + \ldots + n_m - K) \end{aligned} \tag{3.2.4}$$

where λ_1 and λ_2 are called the Lagrange multipliers. At the optimum point, the partial derivative of each variable must vanish; i.e.,

$$\frac{\partial P}{\partial R_1} = 0 \ , \ \frac{\partial P}{\partial R_2} = 0 \ , \ \ldots \ , \ \frac{\partial P}{\partial R_m} = 0$$

$$\frac{\partial P}{\partial n_1} = 0 \ , \ \frac{\partial P}{\partial n_2} = 0 \ , \ \ldots \ , \ \frac{\partial P}{\partial n_m} = 0 \qquad .$$

Taking the partial derivatives of Equation (3.2.4) with respect to each variable yields the following sets of equations:

$$\frac{\partial P}{\partial R_1} = 0 = \frac{\partial f\left(R_1, n_1\right)}{\partial R_1} f\left(R_2, n_2\right) \ldots f\left(R_m, n_m\right) + \lambda_1\left(R_2 R_3 \ldots R_m\right)$$

$$\frac{\partial P}{\partial R_2} = 0 = f\left(R_1, n_1\right) \frac{\partial f\left(R_2, n_2\right)}{\partial R_2} \ldots . f\left(R_m, n_m\right) + \lambda_1\left(R_1 R_3 \ldots R_m\right)$$

$$\begin{array}{ccc} | & | & | \\ | & | & | \\ | & | & | \end{array}$$

$$\frac{\partial P}{\partial R_m} = 0 = f\left(R_1, n_1\right) f\left(R_2, n_2\right) \ldots . f\left(R_{m-1}, n_{m-1}\right) \frac{\partial f\left(R_m, n_m\right)}{\partial R_m}$$

$$+ \lambda_1\left(R_1 R_2 \ldots R_{m-1}\right)$$

(3.2.5)

and

$$\frac{\partial P}{\partial n_1} = 0 = \frac{\partial f\left(R_1, n_1\right)}{\partial n_1} f\left(R_2, n_2\right) \ldots . f\left(R_m, n_m\right) + \lambda_2$$

$$\frac{\partial P}{\partial n_2} = 0 = f\left(R_1, n_1\right) \frac{\partial f\left(R_2, n_2\right)}{\partial n_2} \ldots f\left(R_m, n_m\right) + \lambda_2$$

$$\begin{array}{ccc} | & | & | \\ | & | & | \\ | & | & | \end{array}$$

$$\frac{\partial P}{\partial n_m} = 0 = f\left(R_1, n_1\right) f\left(R_2, n_2\right) \ldots . f\left(R_{m-1}, n_{m-1}\right) \frac{\partial f\left(R_m, n_m\right)}{\partial n_m} + \lambda_2 \quad . \tag{3.2.6}$$

Notice in Equations (3.2.5) that

$$\begin{array}{lcl} R_2 R_3 \ldots . R_m & = & \dfrac{R_s}{R_1} \\ R_1 R_3 \ldots . R_m & = & \dfrac{R_s}{R_2} \\ | \quad\quad | & & | \\ | \quad\quad | & & | \\ | \quad\quad | & & | \\ R_1 R_2 \ldots . R_{m-1} & = & \dfrac{R_s}{R_m} \end{array} \quad .$$

With the above substitutions, the first two sets in Equations (3.2.5) can be solved for $\lambda_1 R_s$ and equated yielding

$$\frac{R_1 \partial f\left(R_1, n_1\right)}{\partial R_1} f\left(R_2, n_2\right) = R_2 f\left(R_1, n_1\right) \frac{\partial f\left(R_2, n_2\right)}{\partial R_2}$$

$$\frac{f\left(R_1, n_1\right)}{f\left(R_2, n_2\right)} = \frac{R_1 \dfrac{\partial f\left(R_1, n_1\right)}{\partial R_1}}{R_2 \dfrac{\partial f\left(R_2, n_2\right)}{\partial R_2}} \quad . \tag{3.2.7}$$

The first two sets of Equation (3.2.6) can be solved for λ_2 and equated resulting in the relationship

$$\frac{\partial f\left(R_1, n_1\right)}{\partial n_1} f\left(R_2, n_2\right) = f\left(R_1, n_1\right) \frac{\partial f\left(R_2, n_2\right)}{\partial n_2}$$

$$\frac{f\left(R_1, n_1\right)}{f\left(R_2, n_2\right)} = \frac{\dfrac{\partial f\left(R_1, n_1\right)}{\partial n_1}}{\dfrac{\partial f\left(R_2, n_2\right)}{\partial n_2}} \quad . \tag{3.2.8}$$

When Equations (3.2.7) and (3.2.8) are solved simultaneously, the result is

$$\frac{R_1 \partial f\left(R_1, n_1\right)}{\partial R_1} \frac{\partial f\left(R_2, n_2\right)}{\partial n_2} = \frac{R_2 \partial f\left(R_2, n_2\right)}{\partial R_2} \frac{\partial f\left(R_1, n_1\right)}{\partial n_1} \tag{3.2.9}$$

and it follows that this relationship is satisfied only if

$$R_1 = R_2$$

and

$$n_1 = n_2 \quad .$$

When the first and third parts of Equations (3.2.5) and (3.2.6) are solved simultaneously, it may be shown in a similar manner that

$$R_1 = R_3$$

and

$$n_1 = n_3$$

or, using the first and last sets of Equations (3.2.5) and (3.2.6), that

$$R_1 = R_m$$

and

$$n_1 = n_m \quad .$$

Therefore, it may be concluded that the conditions

$$R_1 = R_2 = R_3 \text{----} = R_m$$

and

$$n_1 = n_2 = n_3 \text{----} = n_m$$

yield the optimum results.

Strictly speaking, it has not been proven that a maximum value of reliability results from the above conditions, but only that an extremum value has been found. In other words, the vanishing of the derivatives with respect to each of the variables is a necessary but not a sufficient condition for a maximum. However, it will be clear through further considerations that a maximum reliability is given by these values. To be more specific, consider a three-stage system with reliabilities given by

$$P_1 = 1 - \left(1 - R_1\right)^{n_1 - 1} \left[1 + \left(n_1 - 1\right)R_1\right]$$

$$P_2 = 1 - \left(1 - R_2\right)^{n_2 - 1} \left[1 + \left(n_2 - 1\right)R_2\right]$$

$$P_3 = 1 - \left(1 - R_3\right)^{n_3 - 1} \left[1 + \left(n_3 - 1\right)R_3\right] \quad . \tag{3.2.10}$$

Since $P = P_1 \cdot P_2 \cdot P_3$, an expression for P in Lagrange formulation is given by the expression

$$\begin{aligned} P = & \left\{1 - \left(1-R_1\right)^{n-1} \left[1 + \left(n_1-1\right)R_1\right]\right\} \\ & \times \left\{1 - \left(1-R_2\right)^{n_2-1} \left[1 + \left(n_2-1\right)R_2\right]\right\} \\ & \times \left\{1 - \left(1-R_3\right)^{n_3-1} \left[1 + \left(n_3-1\right)R_3\right]\right\} \\ & + \lambda_1 \left(R_1 \cdot R_2 \cdot R_3 - R_s\right) \\ & + \lambda_2 \left(n_1 + n_2 + n_3 - K\right) \end{aligned} \tag{3.2.11}$$

where R_s and K are constants and R_1 , R_2 , R_3 , n_1 , n_2 , and n_3 are variables. Taking the partial derivatives with respect to each variable and setting them equal to zero yields

$$\frac{\partial P}{\partial R_1} = 0 = -\left\{\left(n_1-1\right)\left(1-R_1\right)^{n_1-2}\left[1+\left(n_1-1\right)R_1\right] + \left(n_1-1\right)\left(1-R_1\right)^{n_1-1}\right\}$$

$$\times\left\{1-\left(1-R_2\right)^{n_2-1}\left[1+\left(n_2-1\right)R_2\right]\right\}$$

$$\times\left\{1-\left(1-R_3\right)^{n_3-1}\left[1+\left(n_3-1\right)R_3\right]\right\} + \lambda_1\left(R_2R_3\right)$$

$$\frac{\partial P}{\partial R_2} = 0 = -\left\{1-\left(1-R_1\right)^{n_1-1}\left[1+\left(n_1-1\right)R_1\right]\right\}$$

$$\times\left\{\left(n_2-1\right)\left(1-R_2\right)^{n_2-2}\left[1+\left(n_2-1\right)R_2\right]\right.$$

$$\left. + \left(n_2-1\right)\left(1-R_2\right)^{n_2-1}\right\}$$

$$\times\left\{1-\left(1-R_3\right)^{n_3-1}\left[1+\left(n_3-1\right)R_3\right]\right\} + \lambda_1\left(R_1 \cdot R_3\right)$$

$$\frac{\partial P}{\partial R_3} = 0 = -\left\{1-\left(1-R_1\right)^{n_1-1}\left[1+\left(n_1-1\right)R_1\right]\right\}$$

$$\times\left\{1-\left(1-R_2\right)^{n_2-1}\left[1+\left(n_2-1\right)R_2\right]\right\}$$

$$\times\left\{\left(n_3-1\right)\left(1-R_3\right)^{n_3-2}\left[1+\left(n_3-1\right)R_3\right]\right.$$

$$\left. + \left(n_3-1\right)\left(1-R_3\right)^{n_3-1}\right\} + \lambda_1\left(R_1 \cdot R_2\right)$$

(3.2.12)

$$\frac{\partial P}{\partial n_1} = 0 = -\left\{ \left(1-R_1\right)^{n_1-1} \ell n\left(1-R_1\right) \left[1 + \left(n_1-1\right)R_1\right] - \left(1-R_1\right)^{n_1-1} R_1 \right\}$$

$$\times \left\{ 1 - \left(1-R_2\right)^{n_2-1} \left[1 + \left(n_2-1\right) R_2\right]\right\}$$

$$\times \left\{ 1 - \left(1-R_3\right)^{n_3-1} \left[1 + \left(n_3-1\right)R_3\right]\right\} + \lambda_2$$

$$\frac{\partial P}{\partial n_2} = 0 = - \left\{1 - \left(1-R_1\right)^{n_1-1} \left[1 + \left(n_1-1\right) R_1\right]\right\}$$

$$\times \left\{ \left(1-R_2\right)^{n_2-1} \ell n \left(1-R_2\right) \left[1 + \left(n_2-1\right) R_2\right] - \left(1-R_2\right)^{n_2-1} R_2 \right\}$$

$$\times \left\{1 - \left(1-R_3\right)^{n_3-1} \left[1 + \left(n_3-1\right)R_3\right]\right\} + \lambda_2$$

$$\frac{\partial P}{\partial n_3} = 0 = - \left\{1 - \left(1-R_1\right)^{n_1-1} \left[1 + \left(n_1-1\right) R_1\right]\right\}$$

$$\times \left\{1 - \left(1-R_2\right)^{n_2-1} \left[1 + \left(n_2-1\right)R_2\right]\right\}$$

$$\times \left\{ \left(1-R_3\right)^{n_3-1} \ell n \left(1-R_3\right)\left[1 + \left(n_3-1\right) R_3\right] - \left(1-R_3\right)^{n_3-1} R_3 \right\} + \lambda_2 \quad . \tag{3.2.13}$$

When the first two parts of Equation (3.2.12) are solved for $\lambda_1 R_s$ and equated, the following equation results:

$$\left\{(n_1-1)(1-R_1)^{n_1-2}\left[1+(n_1-1)R_1\right]-(n_1-1)(1-R_1)^{n_1-1}\right\}\left\{1-(1-R_2)^{n_2-1}\left[1+(n_2-1)R_2\right]\right\}$$
$$=\left\{1-(1-R_1)^{n_1-1}\left[1+(n_1-1)R_1\right]\right\}\left\{(n_2-1)(1-R_2)^{n_2-2}\left[1+(n_2-1)R_2\right]-(n_2-1)(1-R_2)^{n_2-1}\right\} \tag{3.2.14}$$

and when the first two parts of Equation (3.2.13) are solved for λ_2 and equated, the following result is obtained:

$$\left\{(1-R_1)^{n_1-1}\ln(1-R_1)\left[1+(n_1-1)R_1\right]-(1-R_1)^{n_1-1}R_1\right\}\left\{1-(1-R_2)^{n_2-1}\left[1+(n_2-1)R_2\right]\right\}$$
$$=\left\{(1-R_1)^{n_1-1}\left[1+(n_1-1)R_1\right]\right\}\left\{(1-R_2)^{n_2-1}\ln(1-R_2)\left[1+(n_2-1)R_2\right]-(1-R_2)^{n_2-1}R_2\right\} \tag{3.2.15}$$

Solving Equations (3.2.14) and (3.2.15) simultaneously yields

$$\left\{(n_2-1)(1-R_2)^{n_2-2}\left[1+(n_2-1)R_2\right]-(n_2-1)(1-R_2)^{n_2-1}\right\}\left\{(1-R_1)^{n_1-1}\ln(1-R_1)\left[1+(n_1-1)R_1\right]-(1-R_1)^{n_1-1}R_1\right\}$$
$$=\left\{(n_1-1)(1-R_1)^{n_1-2}\left[1+(n_1-1)R_1\right]-(n_1-1)(1-R_1)^{n_1-1}\right\}\left\{(1-R_2)^{n_2-1}\ln(1-R_2)\left[1+(n_2-1)R_2\right]-(1-R_2)^{n_2-1}R_2\right\} \tag{3.2.16}$$

It is quite obvious Equation (3.2.16) can only be satisfied if

$$R_1 = R_2$$

and

$$n_1 = n_2 \quad .$$

In a similar manner, it can be shown that $R_1 = R_3$ and $n_1 = n_3$, and the desired results have been obtained.

Thus, the above result justifies the assumptions of modules of equal reliability and equal degrees of redundancy in each stage. With these assumptions, the mathematical models are simplified considerably. However, practical considerations may make them unfeasible at times;

i.e., it may be impossible or inconvenient to divide a nonredundant system into m equivalent modules because "natural" divisions exist in a particular system organization and design. More will be said about this later; however, for the purposes of this chapter, only the above conditions will be treated.

3.3 SYSTEM OPTIMIZATION WITH EQUIVALENT STAGES

The next factor to be considered is system optimization; i.e., given a nonredundant system with a failure probability $\overline{R}_s$, into how many modules should it be divided? What level of redundancy should be utilized to maximize the reliability of the redundant system? To treat this question, a ratio (γ) will be used. The ratio (γ) is defined as the failure probability of a nonredundant system to that of a redundant system. It is given approximately by

$$\gamma \approx \frac{\overline{R}_s}{m\left(\overline{R}_m + \overline{R}_v\right)^{n-1}\left[n - (n-1)\left(\overline{R}_m + \overline{R}_v\right)\right]} \qquad (3.3.1)$$

where the cross terms $\overline{R}_m \overline{R}_v$ have been neglected and $\overline{P} = 1 - \left(1-\overline{P}_m\right)^m$ has been approximated by $\overline{P} \approx m\overline{P}_m$. In Equation (3.3.1), $\overline{R}_m$ is a function of $\overline{R}_s$ and m and is given approximately by the relationship

$$\overline{R}_m \approx \frac{\overline{R}_s}{m} \quad .$$

$\bar{R}_v$ will depend on the logical design of the decision and switching element, which will be covered in detail later. In general, $\bar{R}_v$ will depend on n. With these substitutions, the variables m and $\bar{R}_s$ can be removed from γ, yielding

$$\gamma \approx \frac{\bar{R}_m}{\left(\bar{R}_m + \bar{R}_v\right)^{n-1} \left[n - (n-1)\left(\bar{R}_m + \bar{R}_v\right)\right]} \tag{3.3.2}$$

which is the ratio of the failure probability of a nonredundant module to that of a redundant module or stage. Thus, there are essentially two variables in Equation (3.3.2), $\bar{R}_m$ and n, since $\bar{R}_v$ is also considered to be a function of n. Taking the partial derivatives of Equation (3.3.2) with respect to $\bar{R}_m$ yields

$$\begin{aligned} \frac{\partial \gamma}{\partial \bar{R}_m} \approx {} & \left(\bar{R}_m + \bar{R}_v\right)^{n-1} \left[n - (n-1)\left(\bar{R}_m + \bar{R}_v\right)\right] \\ & - \bar{R}_m \left\{ (n-1)\left(\bar{R}_m + \bar{R}_v\right)^{n-2} \left[n - (n-1)\left(\bar{R}_m + \bar{R}_v\right)\right] \right. \\ & \qquad \left. - \left(\bar{R}_m + \bar{R}_v\right)^{n-1} (n-1) \right\} \end{aligned} \tag{3.3.3}$$

divided by

$$\left\{ \left(\bar{R}_m + \bar{R}_v\right)^{n-1} \left[n - (n-1)\left(\bar{R}_m + \bar{R}_v\right)\right] \right\}^2 .$$

Setting the above equal to zero, multiplying through by the denominator, and dividing through by $\left(\overline{R}_m + \overline{R}_v\right)^{n-2}$ yields

$$\left(\overline{R}_m + \overline{R}_v\right)\left[n-(n-1)\left(\overline{R}_m + \overline{R}_v\right)\right] - \overline{R}_m\left\{(n-1)\left[n-(n-1)\left(\overline{R}_m + \overline{R}_v\right)\right] - (n-1)\left(\overline{R}_m + \overline{R}_v\right)\right\} = 0 \quad . \tag{3.3.4}$$

If second- and higher-order terms are neglected, the above equation becomes approximately

$$n\left(\overline{R}_m + \overline{R}_v\right) - n(n-1)\overline{R}_m \approx 0 \quad . \tag{3.3.5}$$

Solving for $\overline{R}_m$ yields the result

$$\overline{R}_m \approx \frac{\overline{R}_v}{n-2} \quad . \tag{3.3.6}$$

Notice that this general result agrees with the special case considered in Chapter II, where it was shown that with $n = 3$, $\overline{R}_m \approx \overline{R}_v$.

Taking the partial derivative of Equation (3.3.2) with respect to n yields

$$\frac{\partial\gamma}{\partial n} \approx -\overline{R}_m\left\{\left[n-(n-1)\left(\overline{R}_m + \overline{R}_v\right)\right]\left[\left(\overline{R}_m + \overline{R}_v\right)^{n-1}\ln\left(\overline{R}_m + \overline{R}_v\right) + (n-1)\left(\overline{R}_m + \overline{R}_v\right)^{n-2}\frac{\partial\overline{R}_v}{\partial n}\right] + \left(\overline{R}_m + \overline{R}_v\right)^{n-1}\left[1-\left(\overline{R}_m + \overline{R}_v\right) - (n-1)\frac{\partial\overline{R}_v}{\partial n}\right]\right\} = 0 \quad . \tag{3.3.7}$$

The previous equation is over the square of the denominator of Equation (3.3.2). Multiplying through by this and dividing through by $-\bar{R}_m\left(\bar{R}_m+\bar{R}_v\right)^{n-2}$ yields

$$\left[n-(n-1)\left(\bar{R}_m+\bar{R}_v\right)\right]\left[\left(\bar{R}_m+\bar{R}_v\right)\ell n\left(\bar{R}_m+\bar{R}_v\right)+(n-1)\frac{\partial\bar{R}_v}{\partial n}\right]$$

$$+\left(\bar{R}_m+\bar{R}_v\right)\left[1-\left(\bar{R}_m+\bar{R}_v\right)-(n-1)\frac{\partial\bar{R}_v}{\partial n}\right]=0 \quad . \qquad (3.3.8)$$

Equation (3.3.8) must also be compatible with Equation (3.3.6). Therefore, $\bar{R}_m+\bar{R}_v=\dfrac{(n-1)\bar{R}_v}{(n-2)}$ can be substituted into Equation (3.3.8) yielding

$$\left[n-\frac{(n-1)^2\bar{R}_v}{n-2}\right]\left\{\left(\frac{n-1}{n-2}\right)\bar{R}_v\,\ell n\,\frac{(n-1)\bar{R}_v}{n-2}+\frac{(n-1)\partial\bar{R}_v}{\partial n}\right\}$$

$$+\left(\frac{n-1}{n-2}\right)\bar{R}_v\left[1-\left(\frac{n-1}{n-2}\right)\bar{R}_v-\frac{(n-1)\partial\bar{R}_v}{\partial n}\right]=0 \quad .$$

(3.3.9)

Equation (3.3.9) may be expanded to obtain

$$\left[n-\frac{(n-1)^2\bar{R}_v}{n-2}\right]\left(\frac{n-1}{n-2}\right)\bar{R}_v\,\ell n\,\frac{(n-1)\bar{R}_v}{n-2}+\left(\frac{n-1}{n-2}\right)\bar{R}_v$$

$$-\left(\frac{n-1}{n-2}\right)^2\bar{R}_v^{\,2}+\left[n(n-1)-\frac{n(n-1)^2\bar{R}_v}{n-2}\right]\frac{\partial\bar{R}_v}{\partial n}=0 \quad .$$

(3.3.10)

Multiplying Equation (3.3.10) by $\frac{n-2}{n-1}$ yields

$$\left[n - \frac{(n-1)^2 \overline{R}_v}{n-2}\right] \overline{R}_v \ln \frac{(n-1)\overline{R}_v}{n-2} + \overline{R}_v - (n-1)\overline{R}_v^{\,2}$$

$$+ n\left[(n-2) - (n-1)\overline{R}_v\right] \frac{\partial \overline{R}_v}{\partial n} = 0 \quad . \tag{3.3.11}$$

By neglecting second-order terms of $\overline{R}_v$, Equation (3.3.11) is given approximately by

$$\ln \frac{(n-1)\overline{R}_v}{n-2} + \frac{1}{n} + \left[\frac{(n-2)}{\overline{R}_v} - (n-1)\right] \frac{\partial \overline{R}_v}{\partial n} = 0 \tag{3.3.12}$$

or by rearranging terms

$$\ln \frac{(n-1)\overline{R}_v}{n-2} = \left[(n-1) - \frac{(n-2)}{\overline{R}_v}\right] \frac{\partial \overline{R}_v}{\partial n} - \frac{1}{n} \quad . \tag{3.3.13}$$

Equation (3.3.13) is a transcendental function and it is impossible to solve explicitly for n in terms of $\overline{R}_v$ even if $\frac{\partial \overline{R}_v}{\partial n}$ were known. It was noted that $\overline{R}_v$ is also a function of n ; therefore, when n is known, $\overline{R}_v$ and $\frac{\partial \overline{R}_v}{\partial n}$ will also be known. The decision element logic design will now be considered to determine $\overline{R}_v$ and $\frac{\partial \overline{R}_v}{\partial n}$.

3.4 LOGIC DESIGN OF A GENERALIZED DECISION ELEMENT

The decision element to be developed herein must accomplish the following functions:

1. Fault masking such that as long as two modules out of n are operational, the output is always correct.

2. Failure detection to sense that something needs to be done.

3. Failure isolation so that the failed module can be identified.

4. Automatic module switching such that a failed module may be replaced with a good unit.

Factors 1., 2., and 3. above have been considered in Chapter II; therefore, all that remains to be considered here is 4. A block diagram of the decision element which will accomplish these functions is shown in Figure 3.4.1. The diagram consists of three basic parts: (1) module selection logic, (2) failure detection and control logic, and (3) a voter similar to that considered in Chapter II. A decision element will be employed with each module in the system.

The basic operating philosophy is as follows: Out of the n inputs to the module section logic, three are selected for use in the system. Initially, these will be inputs A, B, and C and will be assigned to channels X, Y, and Z, respectively. As failures occur in these channels, they are detected by the failure detection and the control logic which switches out the failed module and switches to the next

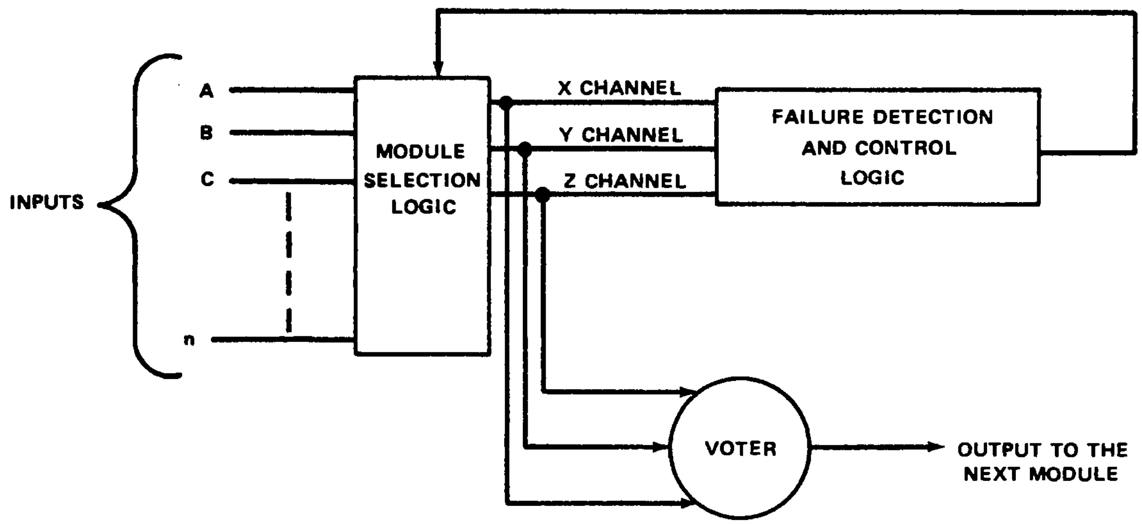


Figure 3.4.1. Block Diagram of Generalized Decision and Switching Element

good unit. Means must be provided for remembering which of the n modules is being used and in which channel it is being employed. Arbitrarily, it was decided to initially assign and use A, B, and C only in the X, Y, and Z channels, respectively. However, the remaining n-3 modules can be assigned sequentially to any of these channels as failures occur. When the nth module has been assigned to either the X, Y, or Z channel, another failure causes either the A, B, or C modules to be reassigned, depending on whether that failure was in the X, Y, or Z channel; a failure in X results in A being reassigned to X , a failure in Y results in B being reassigned to

Y , etc. Thus, A, B, and C can only be assigned to channels X, Y, and Z, respectively.

The basic elements were developed in Chapter II; however, means of selecting three out of the n modules and control and switching logic must also be developed. The detailed logic for controlling a stage consisting of six modules is shown in Figure 3.4.2.

The logic equations for the various portions of the decision element are as follows: Notice that since AND/OR INVERT logic is being used, the output will be in complement form.

Voter

$$\begin{aligned}\bar{f} &= \overline{\bar{X}\bar{Y}\bar{Z} + \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z}} \\ &= \overline{\bar{X}\bar{Y} + \bar{X}\bar{Z} + \bar{Y}\bar{Z}} \quad .\end{aligned} \tag{3.4.1}$$

Error Detection

No X and Y errors have occurred if

$$\overline{X_e} \cdot \overline{Y_e} = \bar{X}\bar{Y}\bar{Z} + \bar{X}\bar{Y}Z + XY\bar{Z} + XYZ \tag{3.4.2}$$

or there exists an error in X or Y if

$$\begin{aligned}X_e + Y_e &= \overline{\bar{X}\bar{Y}\bar{Z} + \bar{X}\bar{Y}Z + XY\bar{Z} + XYZ} \\ &= \overline{\bar{X}\bar{Y} + XY} = E_{R2} \quad .\end{aligned} \tag{3.4.3}$$

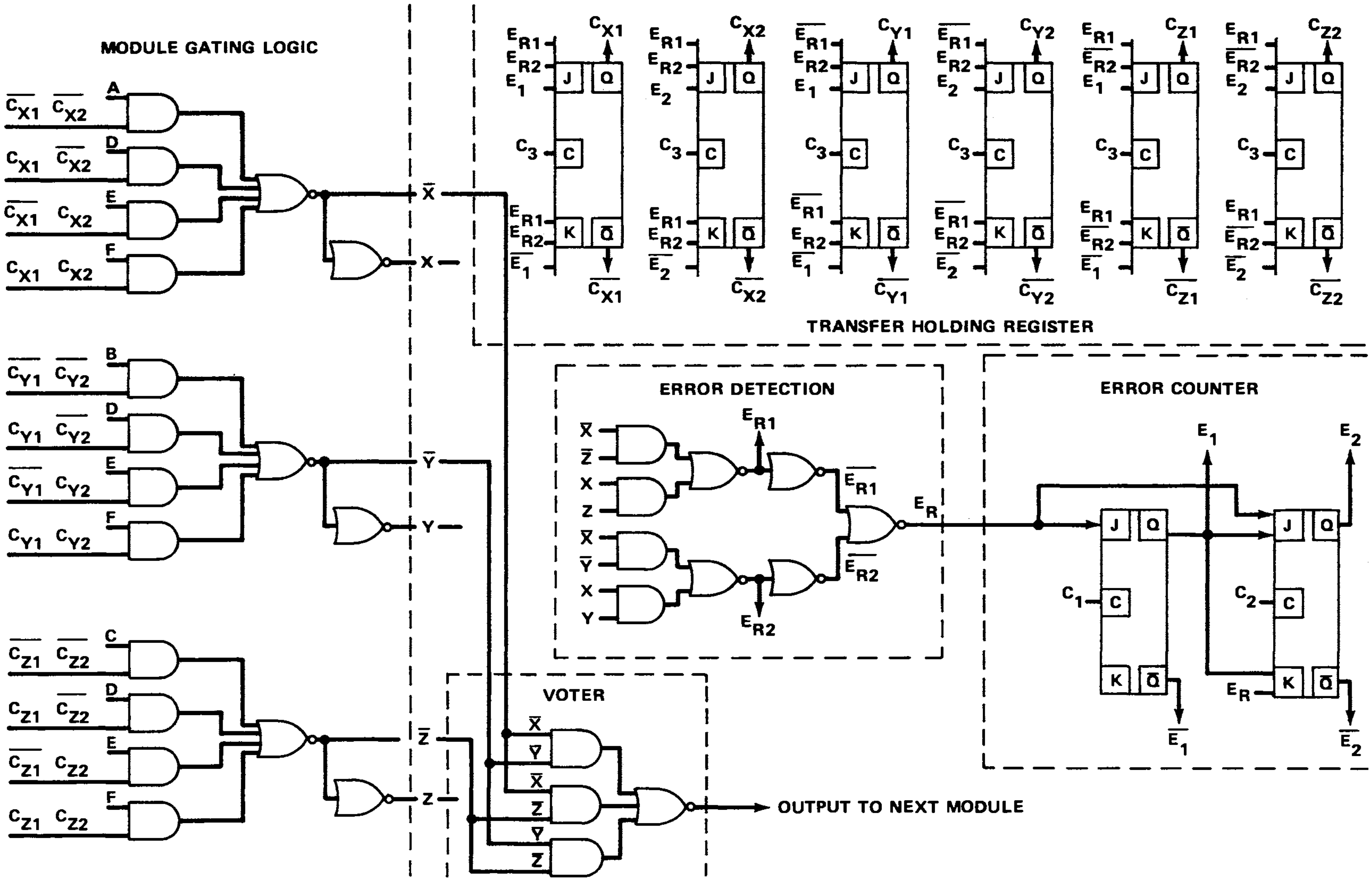


Figure 3.4.2. Logic Diagram For Generalized Decision Element

Similarly, there are no X or Z errors if

$$\overline{X_e} \cdot \overline{Z_e} = \bar{X}\bar{Y}\bar{Z} + \bar{X}Y\bar{Z} + X\bar{Y}Z + XYZ \qquad (3.4.4)$$

or an error has occurred in X or Z if

$$\begin{aligned} X_e + Y_e &= \overline{\bar{X}\bar{Y}\bar{Z} + \bar{X}Y\bar{Z} + X\bar{Y}Z + XYZ} \\ &= \overline{\bar{X}\bar{Z} + XZ} = E_{R1} \quad . \end{aligned} \qquad (3.4.5)$$

If there are no E_{R1} and E_{R2}, then no error has occurred; i.e.,

$$\bar{E}_{R1} \cdot \bar{E}_{R2} = (\bar{X}\bar{Y} + XZ)\ (\bar{X}\bar{Y} + XY) = \bar{E}_R$$

$$\overline{\overline{E_{R1}} \cdot \overline{E_{R2}}} = E_{R1} + E_{R2} = \overline{\bar{X}\bar{Z} + XZ} + \overline{\bar{X}\bar{Y} + XY} = E_R \quad .$$

$$(3.4.6)$$

Thus, $E_{R1} \cdot E_{R2}$ indicates an error in X . Similarly, $E_{R1} \cdot \overline{E_{R2}}$ represents an error in Y and $\overline{E_{R1}} \cdot E_{R2}$ an error in Z .

The error counter simply counts the errors as they occur. It is necessary to remember which module is being used in the X, Y, and Z channels. This function is served by the X, Y, and Z transfer registers which consist of J-K flip-flops. When an error occurs in one of the channels, this value is simply transferred to the appropriate holding register. In other words, these registers, consisting of two flip-flops each, simply copy the error counter when an error is sensed in the appropriate channel.

To better understand the operation of the decision element, it is desirable to go through the sequence of operations which results as failures occur. Initially, the error counter and transfer registers will be in the reset condition. Therefore, the signals $\overline{C_{x1}} \cdot \overline{C_{x2}}$, $\overline{C_{y1}} \cdot \overline{C_{y2}}$, and $\overline{C_{z1}} \cdot \overline{C_{z2}}$ will be in the "set state" or represent a logical "1" condition, thus gating A, B, and C inputs to channels X, Y, and Z, respectively. When an error occurs in X, Y, and Z , the signals $E_{R1} \cdot E_{R2}$, $E_{R1} \cdot \overline{E_{R2}}$, and $\overline{E_{R1}} \cdot E_{R2}$ are turned on, respectively. An error E_R is therefore detected when $\overline{\overline{E_{R1}} \cdot \overline{E_{R2}}}$ is high. Assume, for example, a failure in channel Y (thus indicating a failure of the B input). The signals $E_{R1} \cdot \overline{E_{R2}}$ and E_R are generated and the counter is stepped one. The signal that the error has occurred in the Y channel and some "clock" C_3 , which occurs a short time later, allows the contents of the error counter to be transferred to the Y-channel holding register, thus generating the condition $C_{y1} \cdot \overline{C_{y2}}$. Notice that the contents of the other (X , Z) transfer registers have not changed. As the Y-transfer register changes from the condition $\overline{C_{y1}} \cdot \overline{C_{y2}}$ to $C_{y1} \cdot \overline{C_{y2}}$, input B is switched out of channel Y and input D is switched in. When another failure has been detected, the error counter is advanced by one count. Whether its contents are then transferred to the X-, Y-, or Z-holding registers depends on the channel in which the error occurred. For example, if the second error was also in the Y channel (input D failure), a count of two would be transferred to the Y-holding register and the condition $\overline{C_{y1}} \cdot C_{y2}$ would be generated, thus switching input D out and input E in. However, had the second error occurred in the X channel, the count of two would have been transferred to the X-holding register and the condition $\overline{C_{x1}} \cdot C_{x2}$ would cause E to be switched in the X channel in lieu of input A . The conditions of the other holding registers would not change; therefore, the inputs being employed in those channels cannot change.

The inputs to the next module are voted; thus, these inputs are correct as long as two out of the three inputs are correct. Therefore, there is no particular hurry to switch out the failed input and switch in a new input. This allows the possibility of setting up a sequence of events between a failure indication and the actual switching operation. For instance, if the spare inputs were in a power-off mode, it may be desirable to turn power on the next unit to be employed and allow a warmup period before it is actually employed in the system. Utilizing spares in the powered-down mode might possibly increase system reliability considerably. With only two operational inputs in a stage (i.e., all inputs are incorrect except two), there is a possibility of cycling; i.e., the system searches for an input which agrees with the two being employed. This cycling is not detrimental to the system due to the voted output. The decision element also allows intermittent failures in that when a module fails, it is switched out; however, it will be used again at a later time.

It should be noted that the module-selection gates consist of three input AND circuits. The design herein is general and applicable to any number of inputs or modules. As another module is added, the number of inputs to each input gate is increased by one, and the number of gates in the X, Y, and Z channel is increased by one each. Additionally, to employ four modules, four flip-flops are required; i.e., one in the error counter, and one each in the X-, Y-, and Z-transfer registers. By adding one more flip-flop to the counter and each of the registers, six modules can be accommodated. If it were then required to use seven modules, four more flip-flops would be required, but with the additional flip-flops, up to 10 modules could be accommodated without having additional flip-flops. The relationship between the number of bits in the error counter and n is given by the expression

$$n - 2 \leq 2^{c} \tag{3.4.7}$$

where n is the number of modules employed and c is the number of flip-flops or bits required in the error counter. The total number of flip-flops required is 4c or FF's $\geqq 4\ell n_2(n-2)$. As n is increased by one, the total number of input gates is increased by three, and the total number of inputs to each input gate is increased by one. Therefore, it is readily apparent that the complexity of the decision element is more affected by additional flip-flops rather than gates. Very distinct increases in complexity occur at $n = 5, 7, 11, 19$, etc., because the maximum number of modules which can be used with a c-bit counter is

$$n_{max} = 2^c + 2 \tag{3.4.8}$$

The number of discrete component parts necessary for the decision element is shown in Table 3.4.1.[7] As mentioned previously, definite jumps are noted at 5, 7, etc.

The number of parts in a decision element has been plotted as a function of the number of modules employed in Figure 3.4.3. An analytical expression for the number of parts in the decision element is desired to generalize the treatment. As shown in Figure 3.4.3, the function

$$n_v = (243 + 3n)\,\ell n_2\,(n-2) + 12n + 108 \tag{3.4.9}$$

---

[7]The number of parts used in the logic becomes rather obscure when large-scale integrated circuits are used. The question then arises as to what a part is. Further, the reliability is often quoted in terms of a logic block, and little concern is given to what is in the logic block. Large-scale integrated circuits make the techniques used in this investigation even more attractive. However, to treat relative complexity, discrete component counts will be utilized. Anything gained through integrated circuits then will be over and above that considered here. It is reiterated that the techniques proposed herein become even more palatable or feasible when advanced circuit technology is utilized. In fact, the feasibility of such an approach may depend directly on technological development. If the proposed approach is not feasible today, it will become so at some future date.

TABLE 3.4.1

NUMBER OF EQUIVALENT DISCRETE COMPONENTS REQUIRED
FOR A GIVEN NUMBER OF MODULES

Number of Modules Employed	Number of Component Parts in the Decision Element
3	21
4	420
5	655
6	685
7	960
8	975
9	1000
10	1020

fits the salient points on the graph; i.e., the points beyond which a large increase in decision element hardware is required to obtain an increase in reliability. If optimum n occurs below these points, it is safe to say from the previous discussion that n can be rounded up to these values with a minimum increase in complexity to achieve a sizable reliability gain.

A decision element utilizing the design shown in Figure 3.4.2 has been breadboarded for 10 inputs and is shown in Figure 3.4.4. The

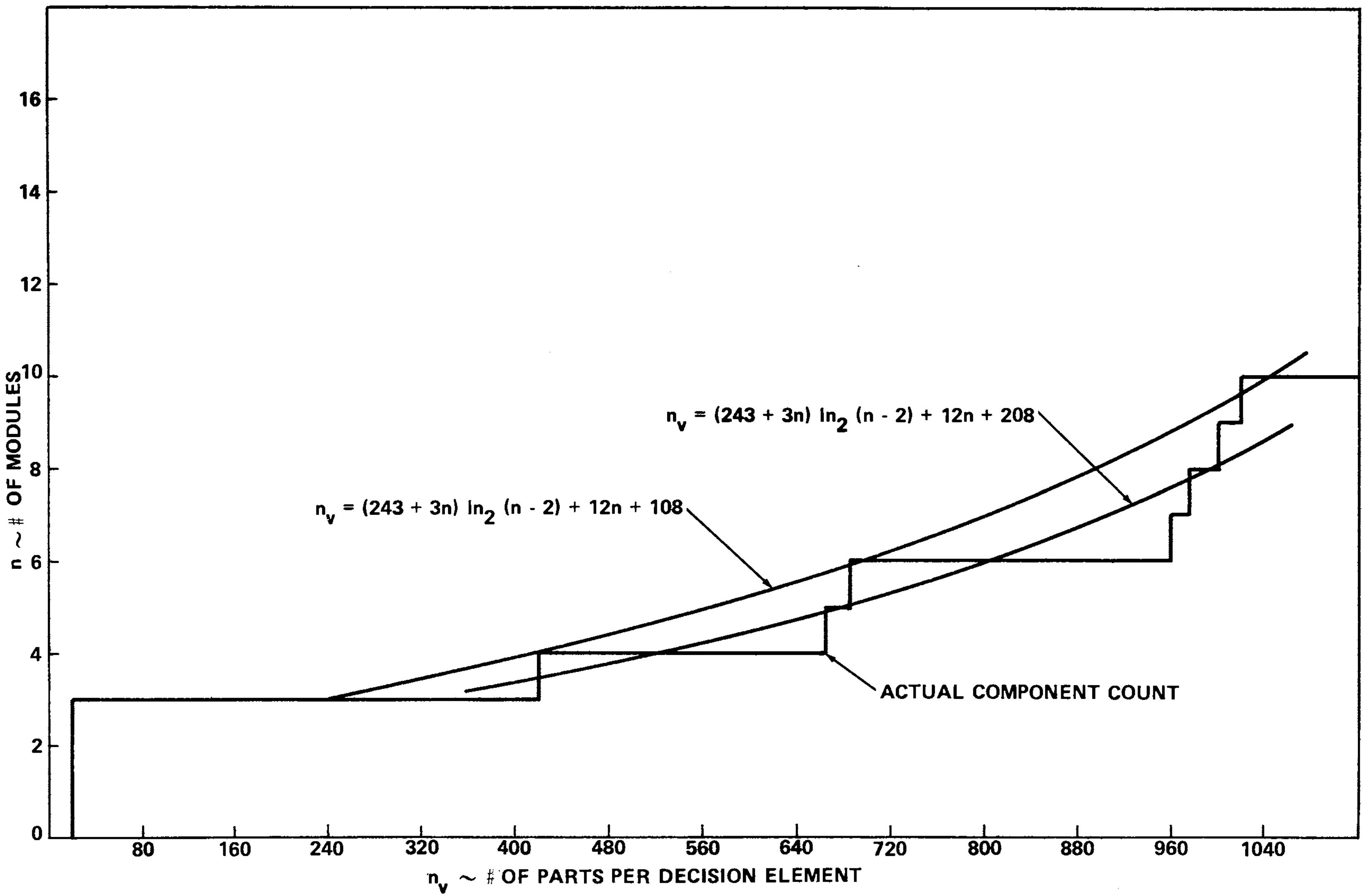


Figure 3.4.3. Curve Fit for n_v Versus n .

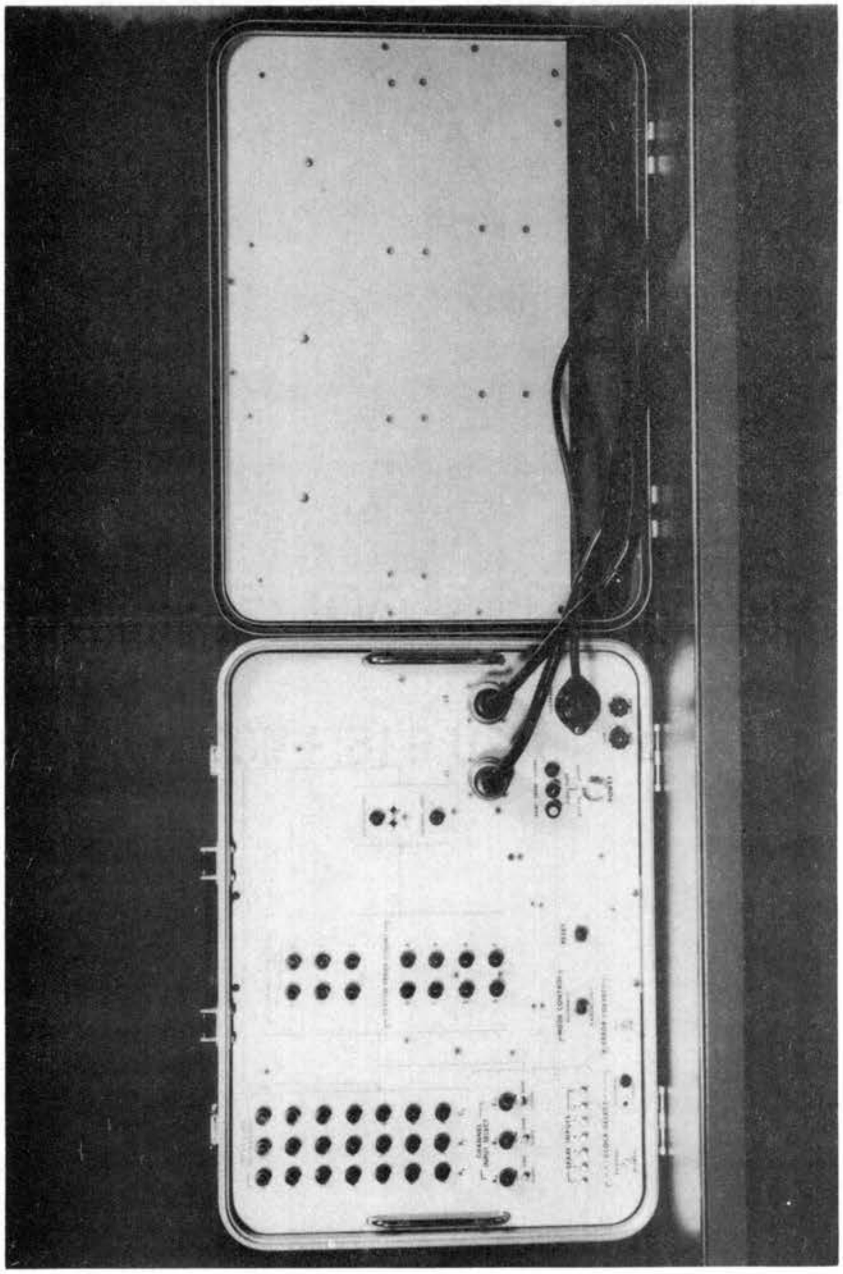


Figure 3. 4. 4. Decision Element Design Breadboard for 10 Inputs

design functions as expected and demonstrates the feasibility of the proposed approach. The breadboard was packaged in a small briefcase to make it portable and more convenient for demonstrational purposes.

3.5 SYSTEM OPTIMIZATION INCORPORATING THE GENERALIZED DECISION ELEMENT DESIGN

An expression for $\overline{R}_v$ in terms of n is desired so that Equation (3.3.13) can be evaluated. Since an exponential distribution is assumed for component parts in this investigation, the failure probability of the decision element is given by the relationship

$$\overline{R}_v = 1 - e^{-n_v \lambda t} \tag{3.5.1}$$

where n_v is the number of components in the element, λ the average component failure rate, and t the operating time. Differentiating Equation (3.5.1) with respect to n yields

$$\frac{\partial \overline{R}_v}{\partial n} = \lambda t e^{-n_v \lambda t} \frac{\partial n_v}{\partial n}$$

where n_v is given by Equation (3.4.9) and

$$\frac{\partial n_v}{\partial n} = 3 \ell n_2 (n-2) + \frac{1.44 (243 + 3n)}{n-2} + 12 \quad . \tag{3.5.2}$$

Figure 3.5.1 shows n_v plotted as a function of n [Equation (3.4.9)] and indicates that the curve is asymptotic to the line $n = 2$. Figure 3.5.2 indicates how $\frac{\partial n_v}{\partial n}$ varies with n and was obtained from Equation (3.5.2). Figures 3.5.3 and 3.5.4 show $\overline{R}_v$ and $\frac{\partial \overline{R}_v}{\partial n}$ plotted as a function of n , respectively. In Figure 3.5.4, a value of $\lambda t = 10^{-4}$ has been arbitrarily chosen; however, this is a reasonable value. From this figure, it is evident that $\frac{\partial \overline{R}_v}{\partial n}$ approaches zero as n increases, and from Figure 3.5.3 it is seen that $\overline{R}_v$ approaches unity as n increases. Thus, for a very large n and small λt , Equation (3.3.13) is given approximately by

$$\ell n \ \frac{n-1}{n-2} \approx -\frac{1}{n} \quad . \tag{3.5.3}$$

This equation is only satisifed in the limit; i.e., as n approaches infinity. Therefore, there is no theoretical limit in the reliability which can be obtained with the technique. However, there are several practical reasons why a limiting value should be placed on n .

It is not necessary to utilize the figures to demonstrate that a reliability as close to unity as desired can be obtained. However, the curves help give an intuitive feeling of the influence of n on each parameter. Substituting Equation (3.5.2) into Equation (3.3.13) gives

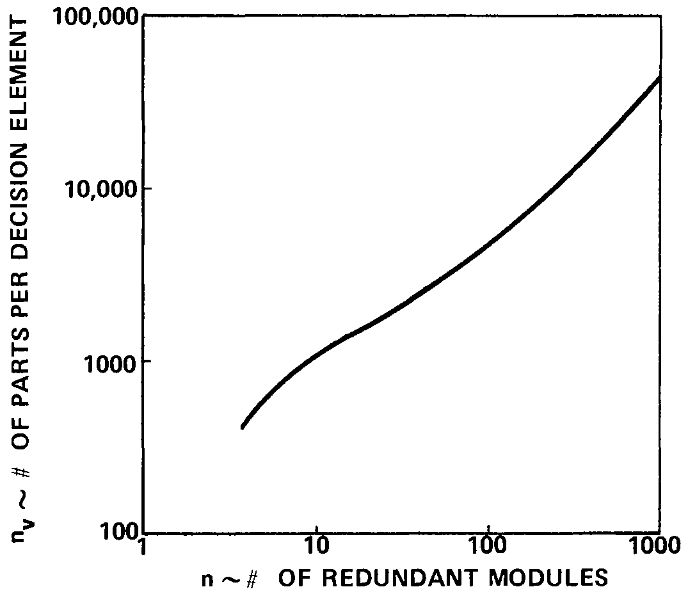


Figure 3.5.1. n_v Versus n Where the Relationship

$$n_v = (243 + 3n)\,\ell n_2 (n-2) + 12n + 108$$

Has Been Used to Determine n_v

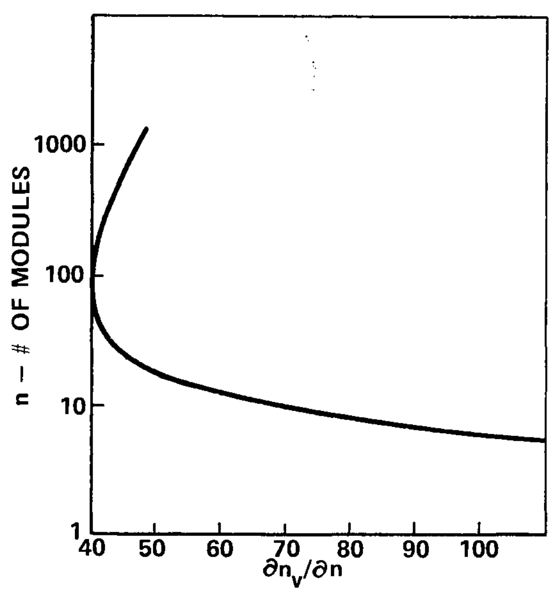


Figure 3.5.2. $\partial n_v/\partial n$ Versus n Where $\partial n_v/\partial n$ is Determined From Equation (3.5.2)

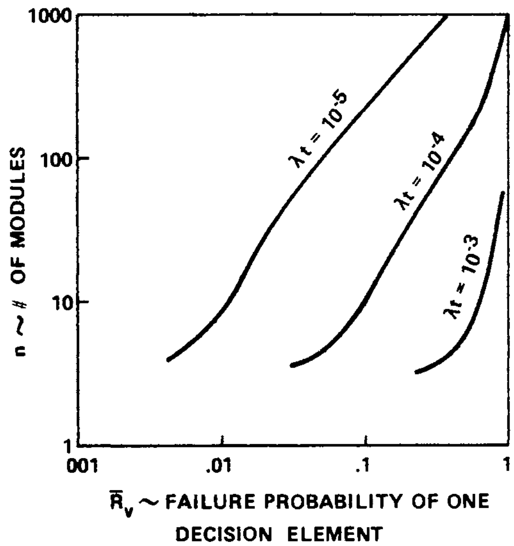


Figure 3.5.3. $\overline{R}_V$ Versus n for $\lambda t = 10^{-5}$, 10^{-4}, and 10^{-3}

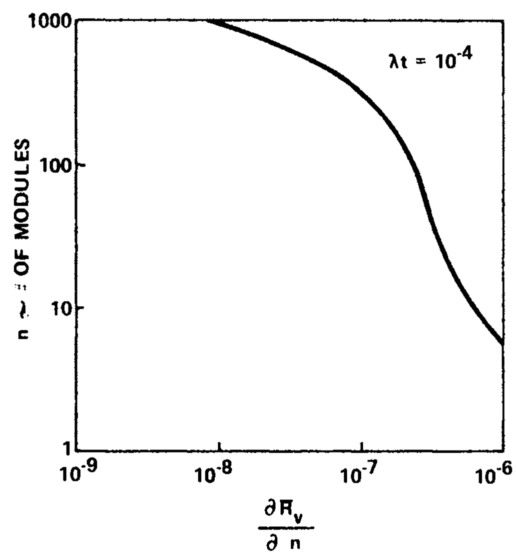


Figure 3.5.4. $\partial\overline{R}_V/\partial n$ Versus n Where $\partial\overline{R}_V/\partial n = \lambda t e^{-n_V \lambda t}\, \partial n_V/\partial n$ and $\lambda t = 10^{-4}$

$$\ell n \left[\frac{(n-1)\overline{R}_v}{n-2}\right] = \left[(n-1) - \frac{(n-2)}{\overline{R}_v}\right] (1-\overline{R}_v)\, \lambda t \left[3\,\ell n_2 (n-2) + \frac{1.44\ (243 + 3n)}{n-2} + 12\right] - \frac{1}{n} \quad . \tag{3.5.4}$$

It is readily apparent from Equation (3.5.1) that $\overline{R}_v$ approaches unity as n_v increases without bound. From Equation (3.4.9), it is seen that as n approaches infinity n_v must also increase without bound. In other words, as n becomes very large, $\overline{R}_v$ approaches unity. Thus, the right side of Equation (3.5.4) approaches zero as n approaches infinity. Since $\overline{R}_v$ and $\frac{n-1}{n-2}$ approach one in the limit, the left side also approaches zero as n increases without limit. Therefore, Equation (3.5.4) is only satisfied as n approaches infinity. It has not been shown yet that the vanishing of the derivative [i.e., satisfying Equation (3.5.4)] yields a minimum failure probability, but only that an extremum has been found. However, it will be shown through numerical evaluation that the extremum found indeed represents a minimum value.

If λt is very small (of the order of 10^{-4} or less), then

$$R_v = e^{-n_v \lambda t} \approx 1 - n_v \lambda t$$

and

$$R_m = e^{-n_m \lambda t} \approx 1 - n_m \lambda t$$

or

$$\bar{R}_v \approx n_v \lambda t$$

and

$$\bar{R}_m \approx n_m \lambda t \qquad .$$

Substituting these values into Equation (3.3.6) yields

$$n_m = \frac{n_v}{n-2} = \frac{N_T}{m} \qquad . \tag{3.5.5}$$

Thus, the number of modules into which a nonredundant system should be divided is given approximately by

$$m = \frac{N_T\,(n-2)}{n_v} = \frac{N_T\,(n-2)}{(243 + 3n)\,\ell n_2\,(n-2) + 12n + 108} \qquad . \tag{3.5.6}$$

An alternate and more accurate expression for m can be obtained by noting that since

$$\bar{R}_v \approx (n-2)\,\bar{R}_m \quad \text{and} \quad \bar{R}_m \approx \frac{\bar{R}_s}{m} \qquad ,$$

$$1 - e^{-n_v \lambda t} \approx (n-2)\,\frac{\bar{R}_s}{m} = \frac{(n-2)\left(1 - e^{-N_T \lambda t}\right)}{m} \qquad .$$

Thus,

$$m \approx \frac{(n-2)\left(1 - e^{-N_T \lambda t}\right)}{1 - e^{-[(243 + 3n)\,\ell n_2\,(n-2) + 12n + 108]\,\lambda t}} \tag{3.5.7}$$

or normalizing by letting $t = \frac{k}{\lambda N_T}$, which is k times the mean time between failures of a nonredundant system, Equation (3.5.7) becomes

$$m \approx \frac{(n-2)\left(1 - e^{-k}\right)}{1 - e^{\frac{-[(243 + 3n)\,\ell n_2\,(n-2) + 12n + 108]}{N_T}}} \; . \qquad (3.5.8)$$

The two previous equations are accurate only if

$$\left(1 - R_m\right)^m \approx m\overline{R}_m \approx \overline{R}_s$$

is accurate.

A limiting value on m can be found from Equation (3.5.7) by letting N_T approach infinity; thus

$$m_{max} = \frac{n-2}{1 - e^{-[(243 + 3n)\,\ell n_2\,(n-2) + 12n + 108]\,\lambda t}} \; . \qquad (3.5.9)$$

The failure probability of a redundant system can be expressed approximately as

$$\overline{P} = 1 - \left\{ 1 - \left[\frac{\overline{R}_s}{m}\left(1 - \overline{R}_v\right) + \overline{R}_v\right]^{n-1} \left\{ n - (n-1)\left[\frac{\overline{R}_s}{m}\left(1 - \overline{R}_v\right) + \overline{R}_v\right]\right\}\right\}^m \qquad (3.5.10)$$

where as usual

$$\overline{R}_s = 1 - e^{-N_T \lambda t} \quad .$$

This equation has been numerically evaluated and $\overline{P}$ has been plotted as a function of m for discrete values of n when $\overline{R}_s = 0.632$ and $\overline{R}_s = 0.865$ and for $\overline{R}_v = 0.1$ and $\overline{R}_v = 0.05$ in Figures 3.5.5 through 3.5.8. As indicated in these figures, Δ is the point where $m \approx \dfrac{(n-2)\,\overline{R}_s}{\overline{R}_v}$ which theoretically has been determined to be the point where the system failure probability is minimum. From these figures, there can be little doubt that the extremum found through theoretical analysis does indeed yield a minimum, as opposed to a maximum, failure probability. They also indicate that the approximations made in theoretically determining that $m \approx \dfrac{(n-2)\,\overline{R}_s}{\overline{R}_v}$ yields optimum design are accurate for all practical purposes. In addition, from these figures two additional facts may be noted: (1) after the optimum point has been reached, increasing m causes the failure probability to increase very slowly (i.e., m is not a very critical parameter), and (2) n has much greater influence on the failure probability than m .

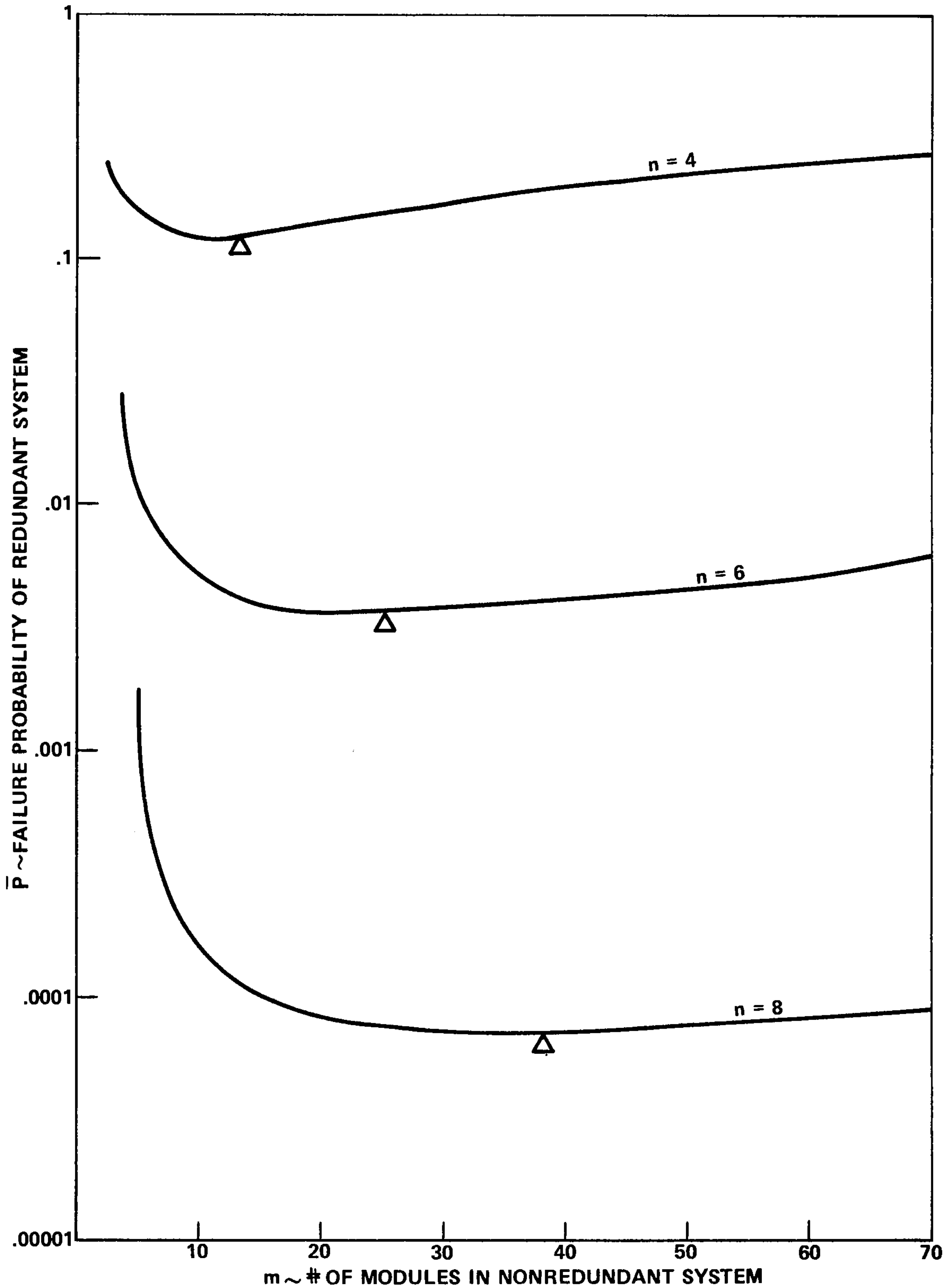


Figure 3.5.5. $\overline{P}$ Versus m for $n = 4$, 6, and 8. $\overline{R}_s = 0.632$ and $\overline{R}_v = 0.1$

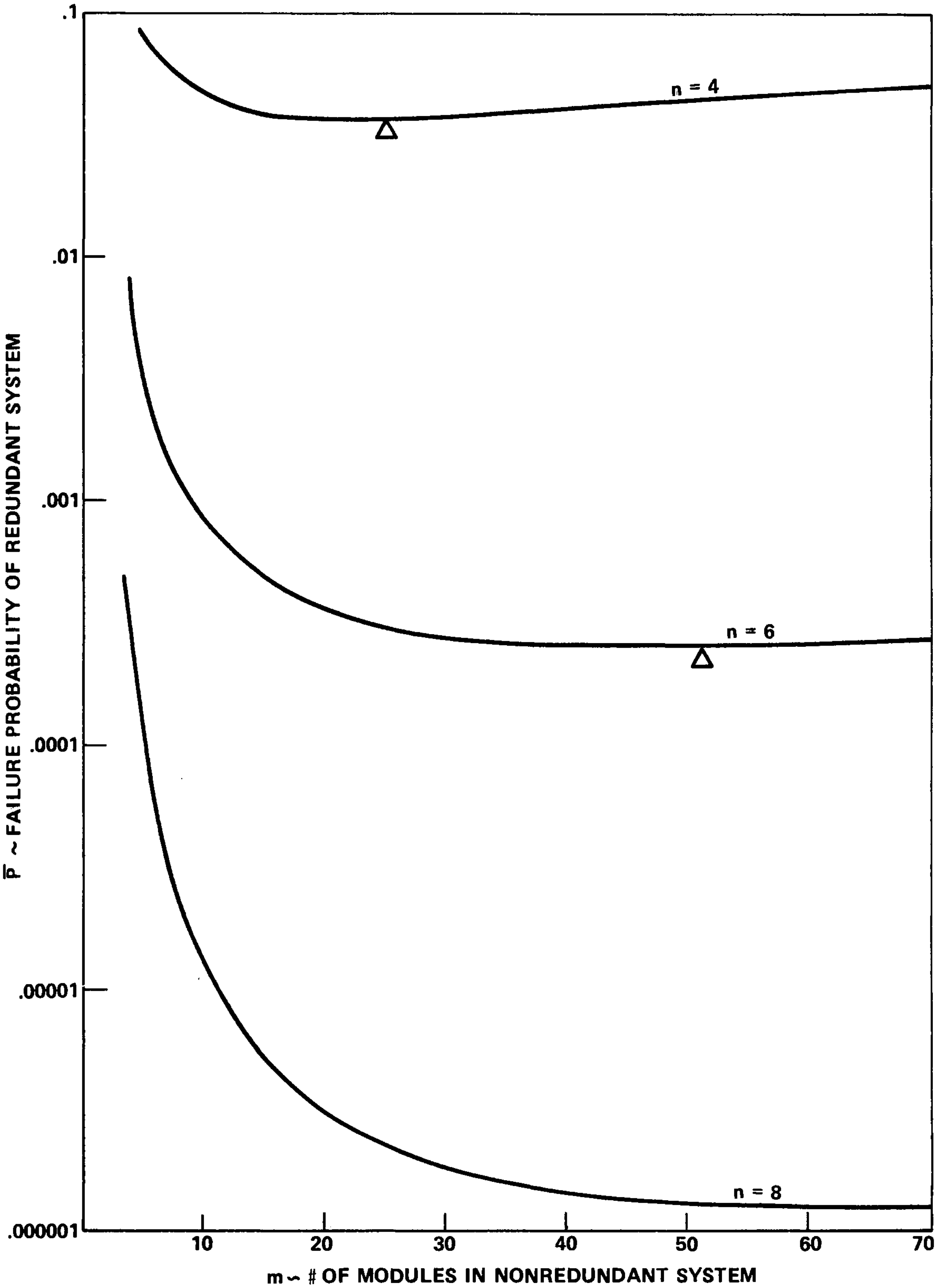


Figure 3.5.6. $\overline{P}$ Versus m for n = 4, 6, and 8. $\overline{R}_s = 0.632$ and $\overline{R}_v = 0.05$

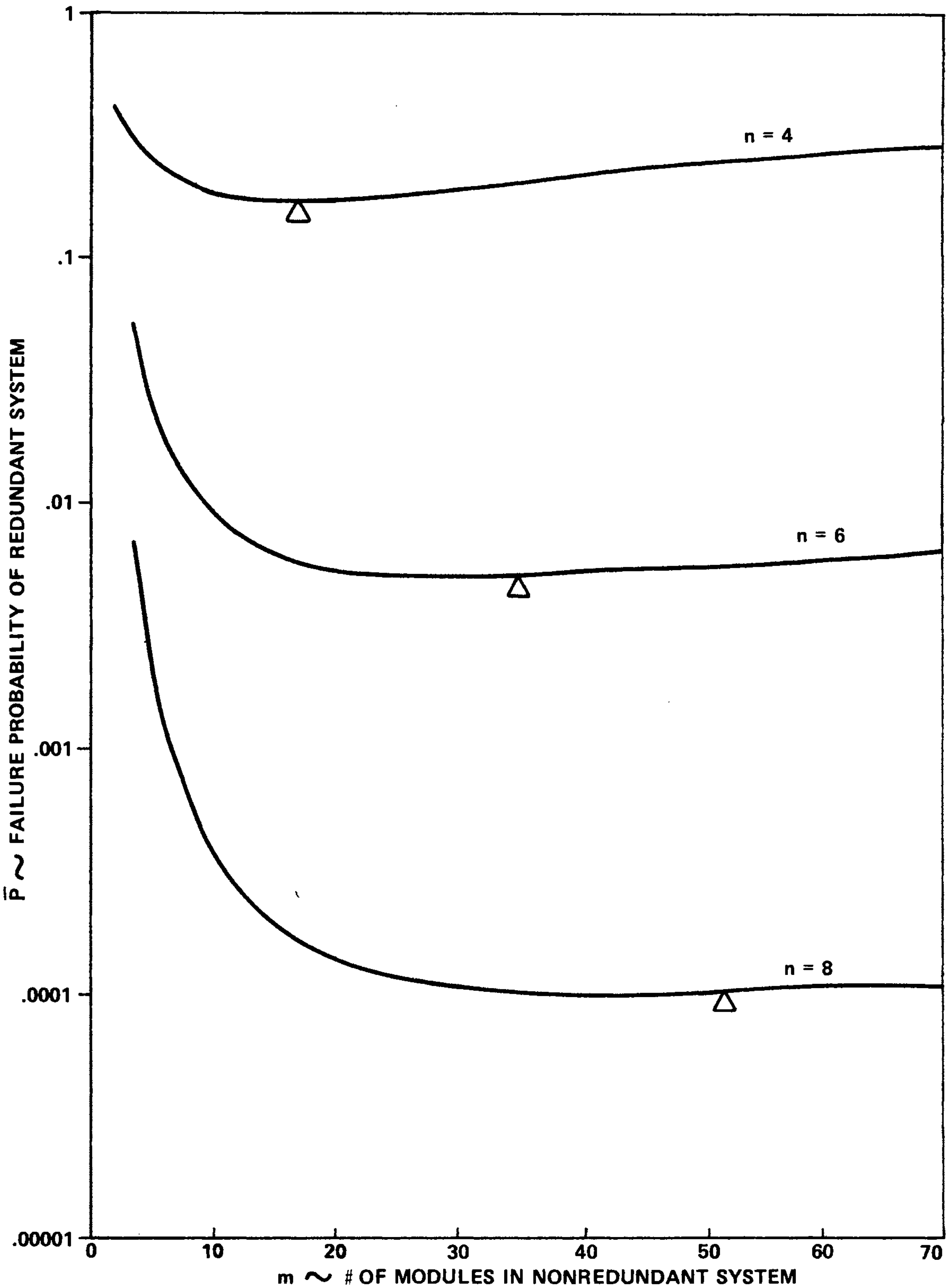


Figure 3.5.7. $\overline{P}$ Versus m for n = 4, 6, and 8. $\overline{R}_s = 0.865$ and $\overline{R}_v = 0.1$

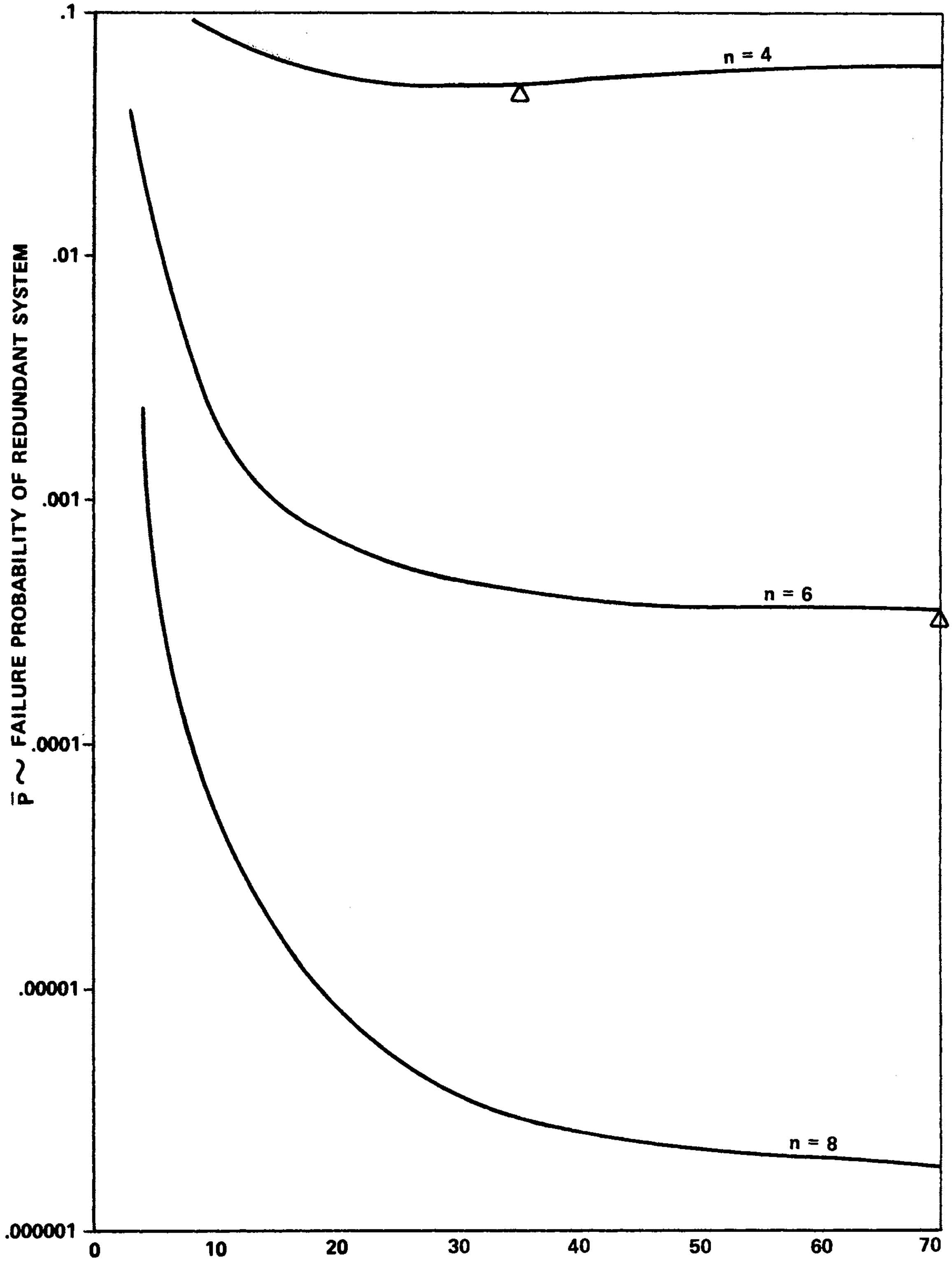


Figure 3.5.8. $\overline{P}$ Versus m for n = 4, 6, and 8
$\overline{R}_s = 0.865$ and $\overline{R}_v = 0.05$

For optimum design,

$$\overline{R}_v = (n-2)\,\overline{R}_m$$

and Equation (3.5.10), which is an approximation, can be written exactly as

$$\overline{P} = 1 - \left\{1 - \left[\overline{R}_m(n-1) - \overline{R}_m^{\,2}(n-2)\right]^{n-1}\left\{n - (n-1)\left[\overline{R}_m(n-1) - \overline{R}_m^{\,2}(n-2)\right]\right\}\right\}^m \tag{3.5.11}$$

where

$$\overline{R}_m = 1 - R_s^{\,1/m} = 1 - \left(1 - \overline{R}_s\right)^{1/m} \quad .$$

This equation yields the optimum design of a system as a function of $\overline{R}_s$, n, and m, when utilized simultaneously with Equation (3.5.8). For a given n, m can be found from Equation (3.5.8), $\overline{R}_m$ can be calculated from the above equation, and finally $\overline{P}$ can be found with Equation (3.5.11). These two equations have been numerically evaluated for $k = 1, 2, 3$; i.e., $\overline{R}_s = 0.632$, 0.865, and 0.950. Figures 3.5.9, 3.5.10, and 3.5.11 show m plotted as a function $\overline{P}$ for values of $n = 4$ through $n = 10$ and for $\overline{R}_s = 0.632$, 0.865, and 0.950, respectively. $\overline{P}$ has also been

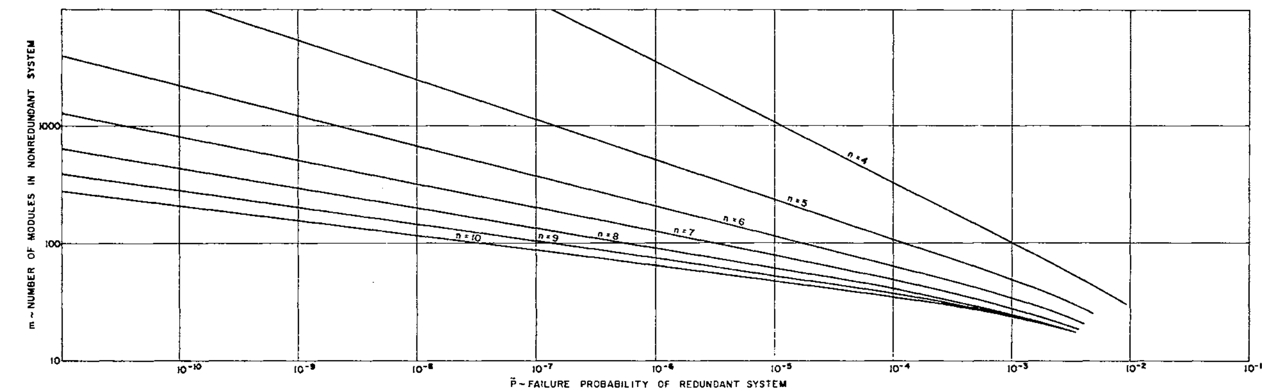


Figure 3.5.9. m Versus $\overline{P}$ for Various n When $\overline{R}_S = 0.632$; i.e., $N_T\lambda t = 1$

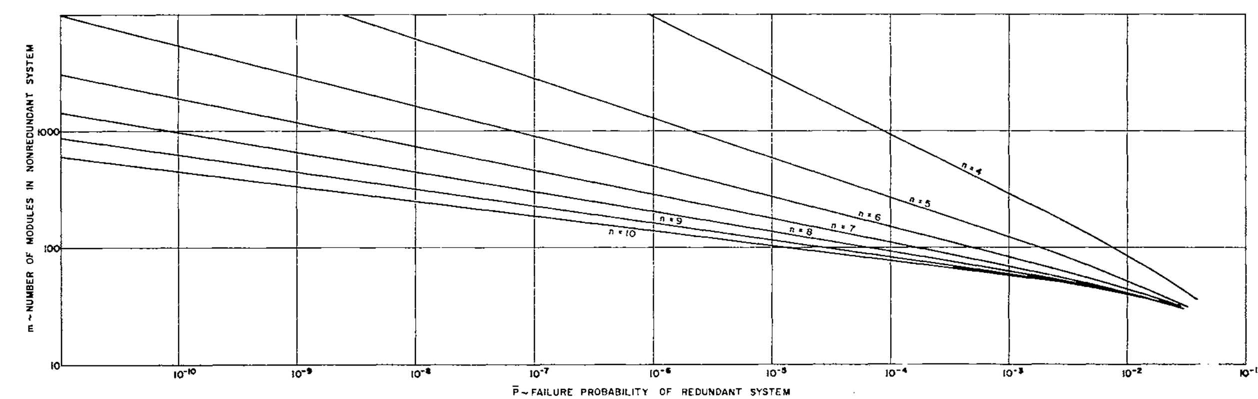


Figure 3.5.10. m Versus $\overline{P}$ for Various n When $\overline{R}_S = 0.865$; i.e., $N_T\lambda t = 2$

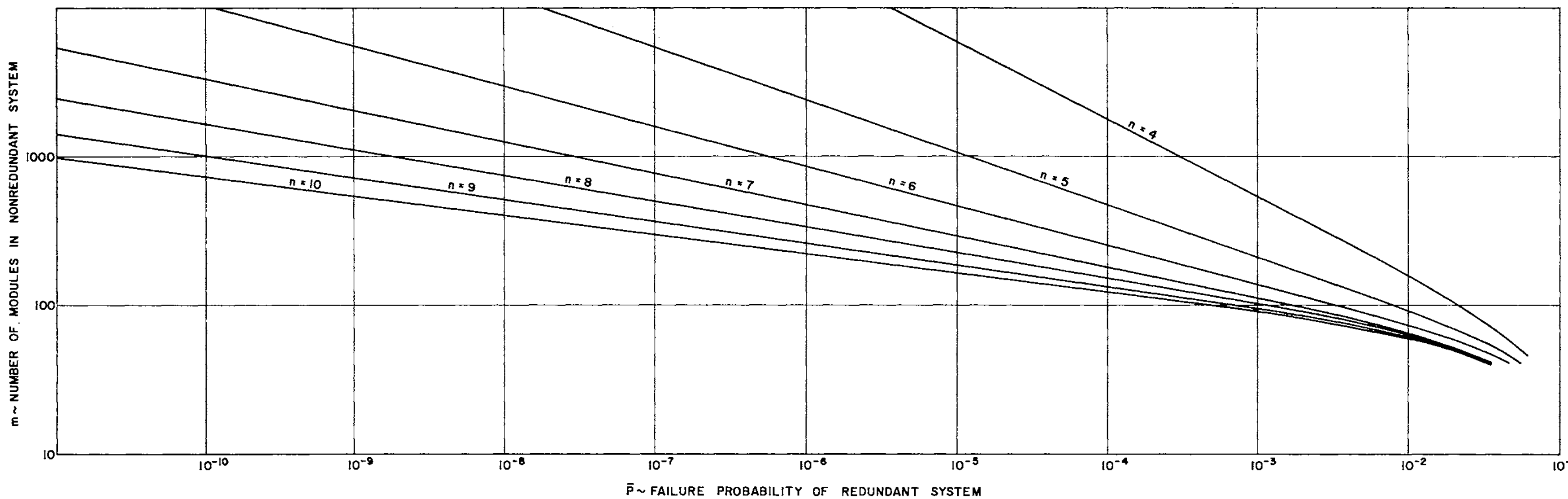


Figure 3.5.11. m Versus $\overline{P}$ for Various n When $\overline{R}_s = 0.950$; i.e., $N_T \lambda t = 3$

plotted as a function of n in Figures 5.3.12, 5.3.13, and 5.3.14 for various size nonredundant systems for $\overline{R}_s = 0.632$, 0.865, and 0.950, respectively.

For illustration, consider three nonredundant systems which contain 25K, 50K, and 75K component parts. The 25K system will be taken as a reference and it will be assumed that it has a reliability of $\overline{R}_s = 0.632$ or $N_T \lambda t = 1$; i.e., it is to be operated until it reaches its mean time to failure. Since

$$N_T \lambda t = 1 ,$$

$$\lambda t = \frac{1}{25,000} = 4 \times 10^{-5} .$$

Assume that it is desired to achieve a reliability goal of 1×10^{-6} . How should each of these systems be organized? From Figures 3.5.12, 3.5.13, and 3.5.14 it is found that $n = 8.6$, 8.1, 8.0 for the 25K, 50K, and 75K systems, respectively. Notice that since λt is assumed to be constant, all three figures must be used. From Figures 3.5.9 through 3.5.11, the values of $m = 80$, 195, and 335 are found which correspond to these values of n for the respective systems. The theoretical solution to this problem is summarized in Table 3.5.1. The significant points of Table 3.5.1 are that n does not change appreciably as the size of the system increases, because the failure probability is held constant, but m increases considerably, and the number of component parts in

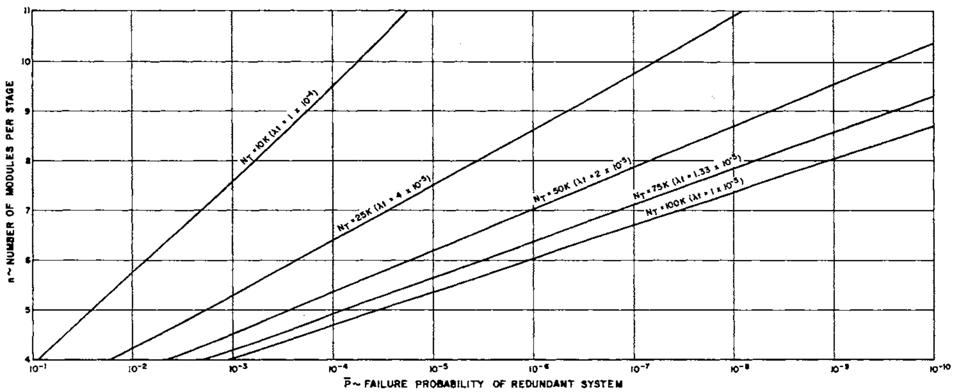


Figure 3.5.12. $\overline{P}$ Versus n for Various Size Nonredundant Systems and for $\overline{R}_S = 0.632$; i.e., $N_T \lambda t = 1$

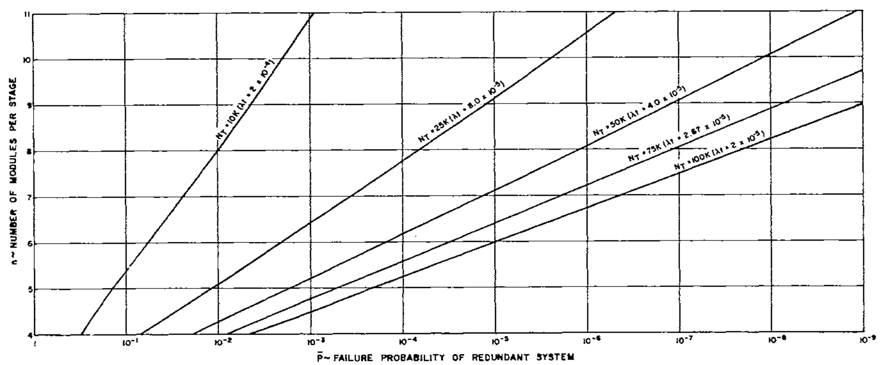


Figure 3.5.13. $\overline{P}$ Versus n for Various Size Nonredundant Systems and for $\overline{R}_S = 0.865$; i.e., $N_T \lambda t = 2$

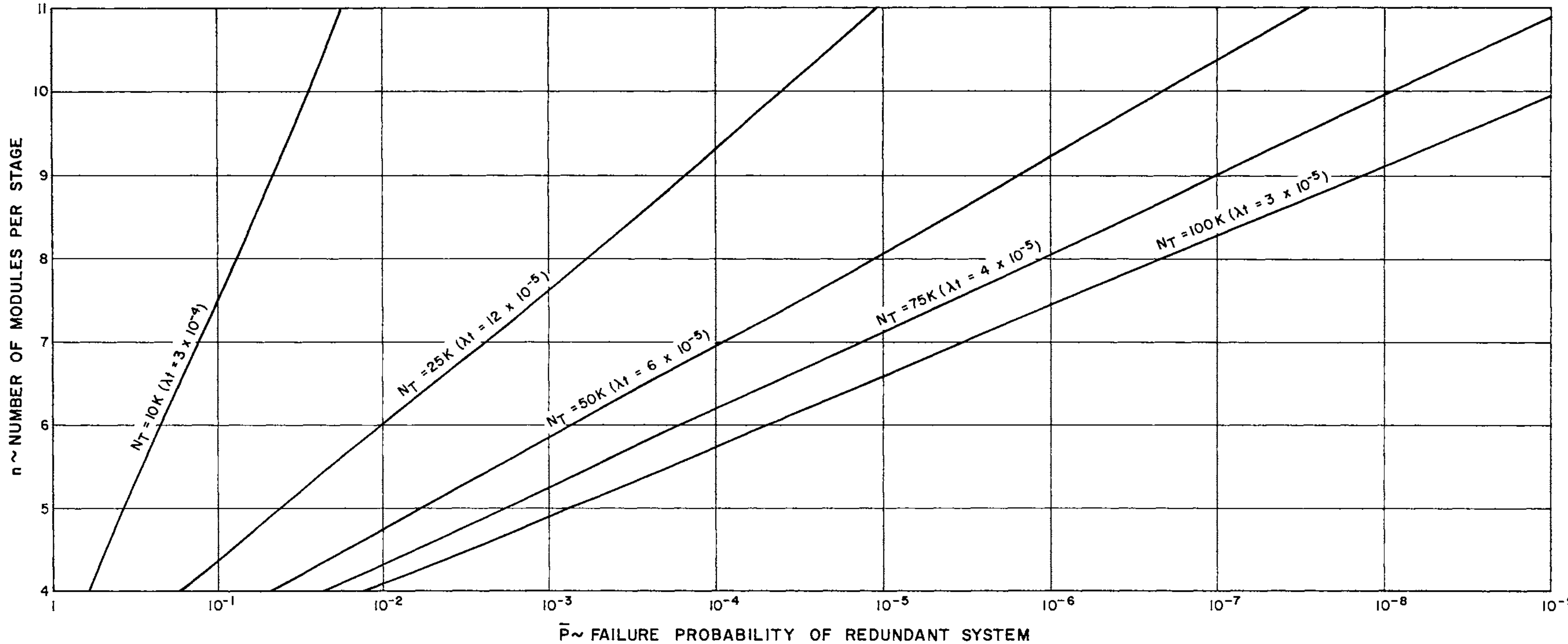


Figure 3.5.14. $\overline{P}$ Versus n for Various Size Nonredundant Systems and for $\overline{R}_S = 0.950$; i.e., $N_T \lambda t = 3$

a module decreases as N_T increases. As noted in Figures 3.5.12, 3.5.13, and 3.5.14, for a constant N_T, to obtain an appreciable decrease in failure probability, n must change considerably. As expected, failure probability is more critically related to n, while the number of parts into which a nonredundant system is divided (m) is more closely associated with system size.

TABLE 3.5.1

THEORETICAL OPTIMUM DESIGN FOR THREE DIFFERENTLY SIZED HYPOTHETICAL SYSTEMS

N_T	$\overline{R}_s$	$\overline{P}$	n	m	n_m
25K	0.632	1×10^{-6}	8.6	80	313
50K	0.865	1×10^{-6}	8.1	195	256
75K	0.950	1×10^{-6}	8.0	335	224

Since n (Table 3.5.1) is not an integer value, those organizations are not realizable. If it is desired to achieve a reliability goal of no less than 1×10^{-6}, then n must be rounded up to nine in the first two cases. But when this is done, the failure probability which can be obtained changes considerably. The results of practical systems utilizing optimum design are given in Table 3.5.2. As in the previous case, $\lambda t = 4 \times 10^{-5}$ has been assumed. The number of components required in the decision element can be found directly from Figure 3.5.1. The relative complexity (C_r) will now be treated analytically in more detail.

TABLE 3.5.2

REALIZABLE OPTIMUM DESIGN FOR THE PREVIOUS HYPOTHETICAL SYSTEMS

N_T	$\overline{R}_s$	$\overline{P}$	n	m	n_m	n_v	n_v/n_m	C_r
25K	0.632	4.6×10^{-7}	9	117	214	1000	4.67	51.1
50K	0.865	1.15×10^{-7}	9	315	159	1000	6.29	65.6
75K	0.950	1.0×10^{-6}	8	335	224	900	4.02	40.1

The complexity or the total number of components in a nonredundant system is given by the relationship

$$N_T = n_m m \tag{3.5.12}$$

where n_m is the number of component parts in a module and m is as previously defined, the total number of modules in a simplex or nonredundant system. The number of components in a redundant module is given by

$$N_m = n(n_m + n_v) \tag{3.5.13}$$

where n is the degree of redundancy applied and n_v the number of parts in the decision element. For optimum design, it has been shown that

$$\overline{R}_m \approx \frac{\overline{R}_v}{n-2}$$

or that the relationship

$$n_v \approx \frac{-\ell n\left[1 - (n-2)\left(1 - e^{-n_m \lambda t}\right)\right]}{\lambda t} \tag{3.5.14}$$

should be satisfied. For small λt, $1 - e^{-n_m \lambda t}$ and $1 - e^{n_v \lambda t}$ can be approximated by $n_m \lambda t$ and $n_v \lambda t$, respectively, and the optimum design is given approximately by

$$n_m \approx \frac{n_v}{n-2} \quad . \tag{3.5.15}$$

In this case, Equation (3.5.13) can be written as

$$N_m \approx n(n-1)\, n_m \tag{3.5.16}$$

or since the number of components in a redundant system is m times that in a redundant module, the total number of components in a redundant system is given by

$$N_r \approx n(n-1)\, n_m\, m \quad . \tag{3.5.17}$$

The relative complexity of a redundant system to that of a simplex system is, therefore, found approximately by dividing Equation (3.5.17) by Equation (3.5.12), yielding

$$C_r \approx n(n-1) \quad . \tag{3.5.18}$$

For $n = 3$, Equation (3.5.18) yields a relative complexity of six, which agrees with that found in Chapter II. The relative complexity estimated by Equation (3.5.18) is given more accurately by the relationship

$$C_r = n \left\{ 1 - \frac{\ln \left[1 - (n-2) \left(1 - e^{-n_m \lambda t} \right) \right]}{n_m \lambda t} \right\} \quad . \tag{3.5.19}$$

Since

$$n_m = \frac{N_T}{m} \quad ,$$

Equation (3.5.19) can be expressed as

$$C_r = n \left\{ 1 - \frac{\ln \left[1 - (n-2) \left(1 - e^{-\frac{N_T \lambda t}{m}} \right) \right]}{\frac{N_T \lambda t}{m}} \right\} \quad . \tag{3.5.20}$$

By letting $t = \frac{k}{\lambda N_T}$ (i.e., by normalizing by expressing t as k times the mtbf of a simplex machine), Equation (3.5.20) becomes

$$C_r = n \left\{ 1 - \frac{m}{k} \ell n \left[1 - (n-2) \left(1 - e^{-k/m} \right) \right] \right\} \quad . \tag{3.5.21}$$

The complexity obtained from Equation (3.5.21) has been plotted as a function of m for several values of n when $k = 1$ as shown in Figure 3.5.15. As m increases, C_r rapidly approaches the value approximated by Equation (3.5.18). Also, this equation is more accurate when n is small. Although the effect of k on the relative complexity cannot be determined from this figure, it can be shown that as k increases the curves approach more slowly the values estimated by Equation (3.5.18). In other words, Equation (3.5.18) becomes a better approximation as k and n become smaller; however, m has a dominating influence. The relative complexity is also given by the relationship

$$C_r = \frac{n\left(n_m + n_v\right)}{n_m} \quad . \tag{3.5.22}$$

This relationship was used in calculating the values in Table 3.5.2 because n_m and n_v were known. When Equations (3.5.18) and (3.5.22) are equated and solved for n_m, the result is that obtained previously in Equation (3.5.15).

The results of this section have shown that infinite reliability can be obtained with the proposed approach under the assumption that a nonredundant system can be broken into as many modules as desired.

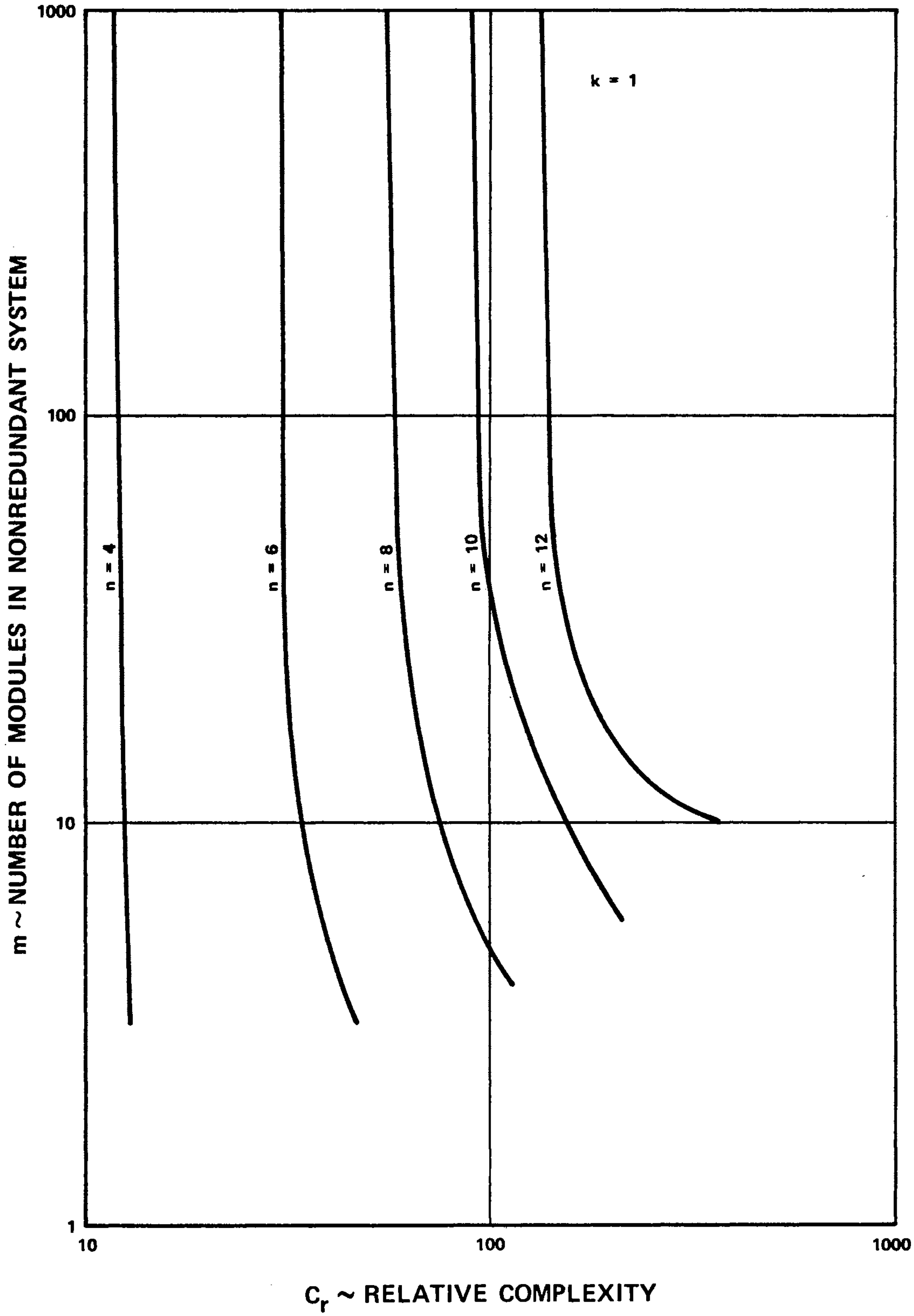


Figure 3.5.15. Relative Complexity (C_r) Versus Number of Modules in Nonredundant System (m)

However, the approach is expensive since the relative complexity increases roughly as the square of the degree of redundancy employed when n is large [Equation (3.5.18)].

3.6 CONSIDERATION OF THE REQUIREMENT FOR A SINGLE OUTPUT

The previous development treated the idealized two-out-of-n organization with little regard to practical application. It has been shown that when this technique has been applied to a system, any desired reliability can be obtained if the resulting complexity can be tolerated. The question of the feasibility of application of this technique to a practical system arises: Can unlimited reliability really be achieved if relative complexity is not a factor? The answer to this question naturally depends on the system itself, and the remainder of this chapter will be devoted to a discussion of this question.

The physical nature and requirements of the outputs of an individual system provide the key to the reliability which can be obtained with the technique proposed herein. If this technique can also be employed in the next system which follows it, or all the redundant signals can be used in the preceding system, then with the proposed technique a reliability as close to unity as desired can be achieved, provided that cost (i.e., complexity) is of no concern. However, in many applications, it is not possible to use multioutputs from a system.

Suppose, for example, that signals from a digital computer system position a servomechanism system. It is conceivable that the servomechanism is redundant; however, it may also be possible, and more likely, that the servomechanism system is being used to position a single physical device. Thus, it is possible that regardless of the degree of redundancy that is applied internally in a system only one output can be accommodated. In any system, it is most probable that there exists a requirement for a single signal at some point, in which case the redundant system must be "necked" down to provide a single output. The single element which accomplishes the converging of the redundant signals must act in series with the redundant elements; therefore, it introduces the possibility of a single-point failure and thus limits the reliability of the total system because the reliability can never be greater than the reliability of this element. In terms of the previous discussion, this element is simply the decision and switching element which accepts n_i inputs and provides a single output. This element is identical to that shown in Figure 3.4.2.

The old adage that a chain is no stronger than its weakest link also applies to the reliability of a system consisting of a chain of several elements. If a decision element at the output of a system is required which acts in series with the redundant system, the total system reliability can be no greater than the reliability of the decision element. Thus, system reliability becomes limited when viewed from this point.

However, this may not be a severe limitation because the overall system may consist of thousands or hundreds of thousands of component parts, while the single decision element may be made up of only a hundred or less component parts. The reliability of the configuration being considered is given by

$$P = \left\{1 - \left(\bar{R}_m + \bar{R}_v - \bar{R}_m\bar{R}_v\right)^{n-1} \left[n - (n-1)\left(\bar{R}_m + \bar{R}_v - \bar{R}_m\bar{R}_v\right)\right]\right\}^{m-1}$$

$$\times \left\{1 - \bar{R}_m^{\,n-1} \left[n - (n-1)\,\bar{R}_m\right]\right\} R_v \tag{3.6.1}$$

where the first portion of the equation is the reliability of the $m-1$ stages containing a decision element with each module, the second portion is the reliability of the m^{th} module containing no decision element, and R_v is the reliability of the single decision element in the serial chain. Notice that

$$\bar{R}_m + \bar{R}_v - \bar{R}_m\bar{R}_v = \bar{R}_m\left(1 - \bar{R}_v\right) + \bar{R}_v \quad ,$$

and since

$$\bar{R}_m = 1 - R_s^{\,1/m}$$

then

$$\bar{R}_m + \bar{R}_v - \bar{R}_m\bar{R}_v = 1 - R_v R_s^{\,1/m} \quad . \tag{3.6.2}$$

Since

$$R_v = e^{-n_v \lambda t}$$

and

$$R_s = e^{-N_T \lambda t}$$

or when normalized about $t = \dfrac{k}{\lambda N_T}$,

$$R_v = e^{-kn_v/N_T}$$

and

$$R_s = e^{-k} \quad .$$

Equation (3.6.2) can be expressed as

$$\overline{R}_m + \overline{R}_v - \overline{R}_m \overline{R}_v = 1 - e^{-k\left(\frac{n_v}{N_T} + \frac{1}{m}\right)} \quad . \qquad (3.6.3)$$

Substituting Equation (3.6.3) and the above relationships into Equation (3.6.1) yields

$$P = \left\{ 1 - \left[1 - e^{-k\left(\frac{n_v}{N_T} + \frac{1}{m}\right)}\right]^{n-1} \left[n - (n-1)\left(1 - e^{-k\left(\frac{n_v}{N_T} + \frac{1}{m}\right)}\right)\right]\right\}^{m-1}$$

$$\left\{ 1 - \left(1 - e^{-k/m}\right)^{n-1} \left[n - (n-1)\left(1 - e^{-k/m}\right)\right]\right\} e^{-\frac{kn_v}{N_T}} \qquad (3.6.4)$$

where

$$n_v = [(243 + 3n)\,\ell n_2\,(n-2) + 12n + 108] \quad .$$

Equation (3.6.4) has been numerically evaluated with the aid of a digital computer. The redundant system failure probability $(\overline{P})$ has been plotted as a function of the number of modules in a simplex system (m) for various size systems and for $N_T\lambda t = k = 1, 2,$ and 3 in Figures 3.6.1, 3.6.2, and 3.6.3, respectively. From these figures, it is clear that $\overline{P}$ does indeed reach a minimum value. Figures 3.6.4, 3.6.5, and 3.6.6 show m plotted as a function n for the parameters contained in the previous curves. The vertices of the curves (i.e., where n reaches a minimum value) correspond to the minimum $\overline{P}$ found in Figures 3.6.1 through 3.6.3.

For illustration, consider a system consisting of 25,000 component parts and which has a reliability of 0.368 $(k = 1)$ after some period of operation. Thus,

$$\lambda t = \frac{1}{25,000} = 4 \times 10^{-5} \quad .$$

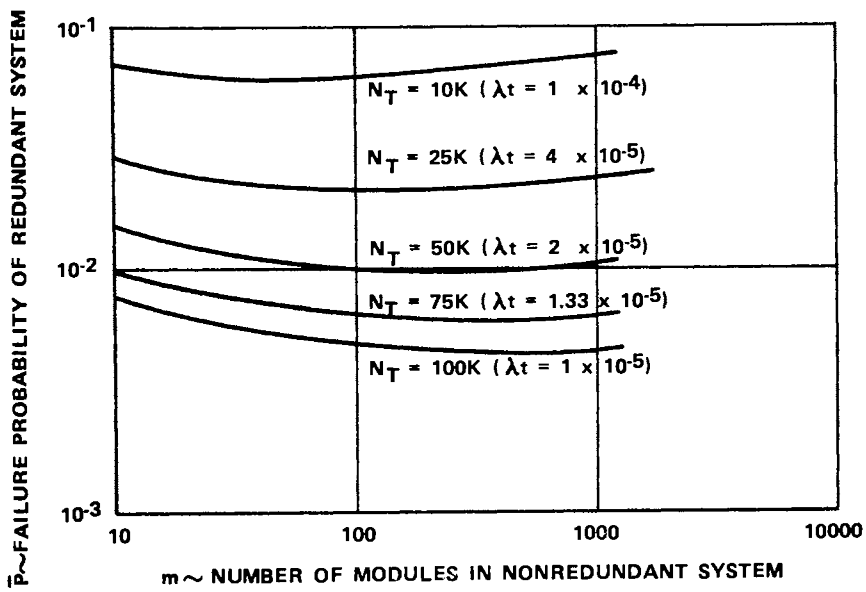


Figure 3.6.1. m Versus $\overline{\text{P}}$ for Systems of Size N_T Containing a Single Output, $N_T\lambda t = k = 1$

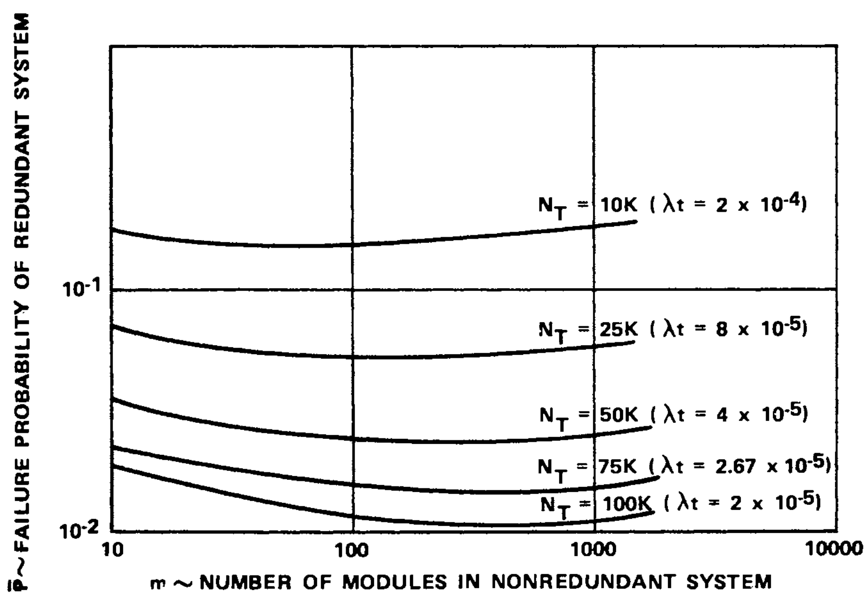


Figure 3.6.2. m Versus $\overline{\text{P}}$ for Systems of Size N_T Containing a Single Output, $N_T\lambda t = k = 2$

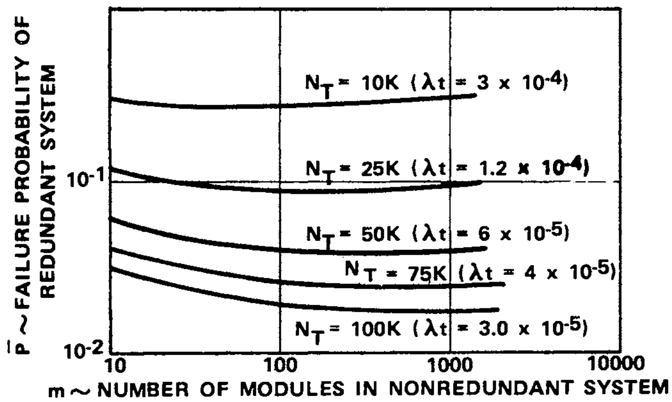


Figure 3.6.3. m Versus $\overline{P}$ for Systems of Size N_T Containing a Single Output, $N_T\lambda t = k = 3$

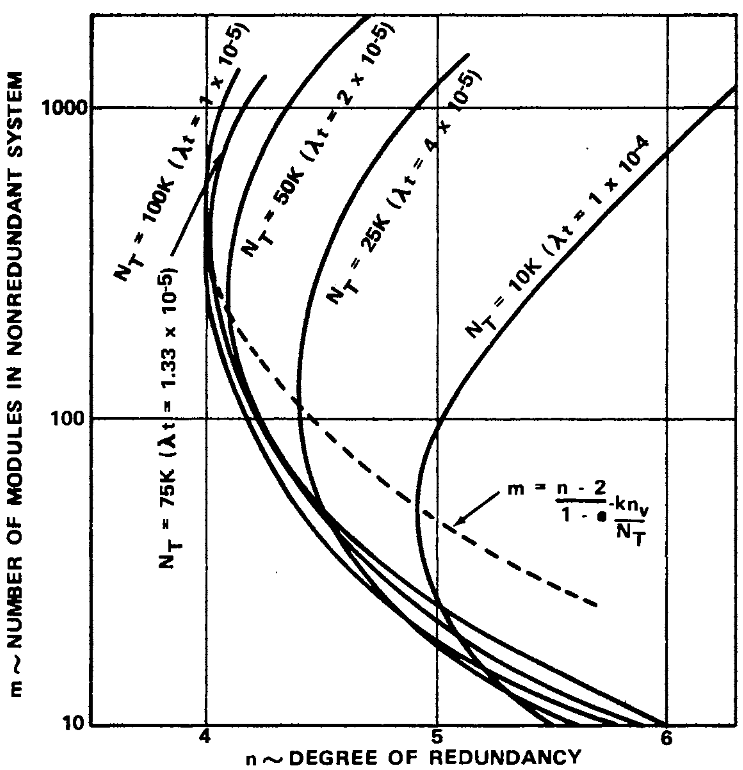


Figure 3.6.4. n Versus m for Systems of Size N_T Containing a Single Output, $N_T\lambda t = k = 1$

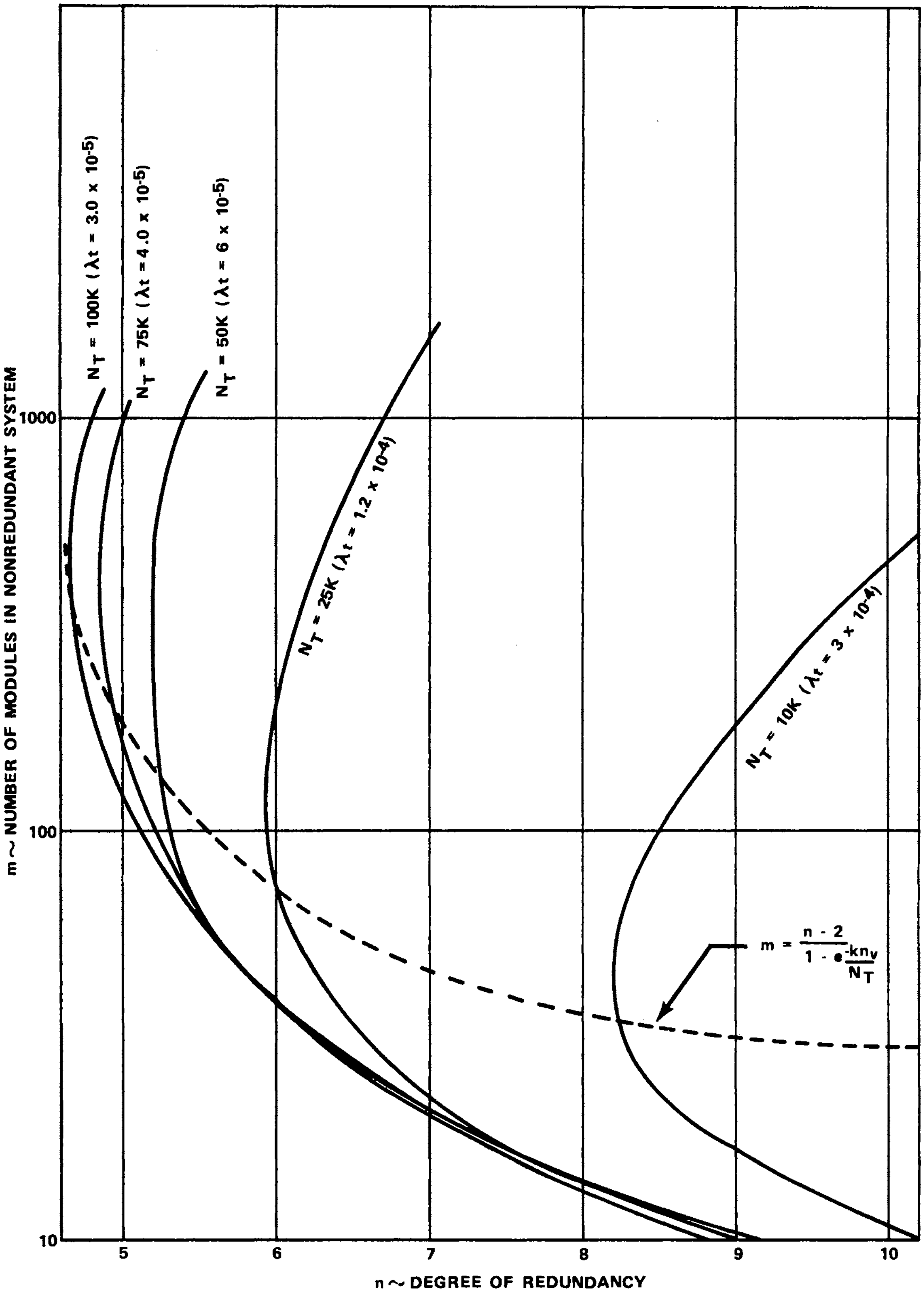


Figure 3.6.6. n Versus m for Systems of Size N_T Containing a Single Output, $N_T \lambda t = k = 3$

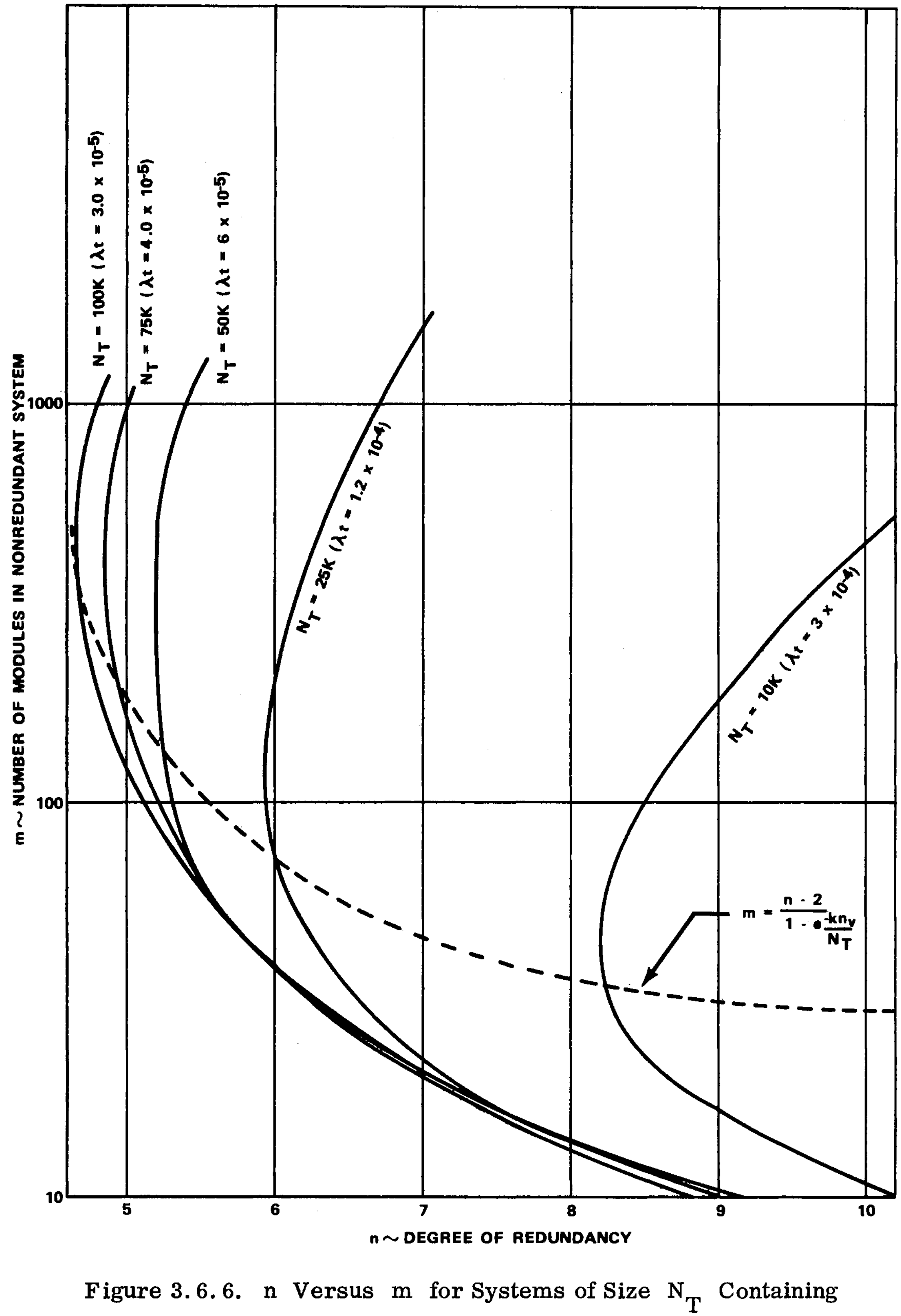


Figure 3.6.6. n Versus m for Systems of Size N_T Containing a Single Output, $N_T \lambda t = k = 3$

A single output is required. What is the minimum failure probability which can be achieved, and how should the system be organized? From Figure 3.6.1, a minimum $\overline{P}$ of approximately 2.12×10^{-2} is found at $m \approx 110$. Figure 3.6.4 gives the value of the degree of redundancy (n) which should be employed in the system at the point where $m \approx 110$ and $N_T = 25,000$ to be 4.4. The ratio of failure probability of the simplex system to that of the redundant system is approximately 30.

Suppose that the simplex system just considered doubled its size in its development process. What is the minimum failure probability which can be obtained, and how should it be organized? Assuming the system had the same component failure rate and operating time, $\lambda t = 4 \times 10^{-5}$, since N_T has doubled, then $k = 2$ must be used; i.e., at 50K, $m = 300$ must be employed. In Figure 3.6.2, a minimum $\overline{P}$ of 2.35×10^{-2} is found when $m \approx 300$. A value of $n \approx 4.7$ is then found in Figure 3.6.5. Therefore, approximately the same failure probabilities can be achieved in the two systems by varying the way the nonredundant system is divided and by employing different degrees of redundancy.

An interesting result is that the system is organized in such a manner to fulfill the m_{max} relationship found in Equation (3.5.9). This function has been plotted and appears as a dotted curve in Figures 3.6.4, 3.6.5, and 3.6.6. It is seen that this curve passes through the vertices of the other curves, which indicates that as much reliability as possible must be gained in the system through the subdivision of the nonredundant system. Although as shown previously, reliability is more readily affected by changing n rather than m for single output systems; increasing n causes the reliability of the last decision element to decrease, thereby decreasing the overall redundant system reliability.

Since the values of n in the example are not integers, they are only of theoretical value and should be rounded for any practical application. However, theoretically they are of considerable value in determining the effects of system parameters on system reliability.

Since the condition

$$m_{max} = \frac{n-2}{1 - e^{-kn_v/N_T}}$$

is always approximately fulfilled, it is possible to remove a variable, either N_T or m , from Equation (3.6.4). Since N_T is generally known, it may be more beneficial to remove m by substituting the above expression into Equation (3.6.4). The expression

$$
\begin{aligned}
P = \Bigg\{ 1 - &\left[1 - e^{-k\left(\frac{n_v}{N_T} + \frac{1}{n-2}\right)} \cdot e^{\frac{k}{n-2} e^{-\frac{k}{n-2}}} \right]^{(n-1)} \\
&\left[n - (n-1)\left(1 - e^{-k\left(\frac{n_v}{N_T} + \frac{1}{n-2}\right)} \right) \cdot e^{\frac{k}{n-2} e^{-\frac{kn_v}{N_T}}} \right] \Bigg\}^{\frac{n-2}{1-e^{-kn_v/N_T}}} \\
\Bigg\{ 1 - &\left[1 - e^{-\frac{k}{n-2}\left(1 - e^{-\frac{kn_v}{N_T}}\right)} \right]^{(n-1)} \\
&\left[n - (n-1)\left(1 - e^{-\frac{k}{n-2}\left(1 - e^{-\frac{kn_v}{N_T}}\right)} \right) \right] \Bigg\} e^{-\frac{kn_v}{N_T}}
\end{aligned}
\qquad (3.6.5)
$$

is then obtained where

$$
n_v = [(243 + 3n)\, \ell n_2\, (n-2) + 12n + 108] \quad .
$$

An evaluation of Equation (3.6.5) will not be undertaken since little additional information would be obtained.

The relative complexity of the system can be found from Equation (3.5.21) or (3.5.22). Since the value of n is limited, a limit exists in relative complexity.

In this section it has been shown that the failure probability of a simplex system can be decreased considerably, particularly if the simplex system is very unreliable, even when a single system output is required. It has been determined how a system should be organized to achieve maximum gain in reliability. However, due to the single output requirement, it is not possible to obtain infinite reliability as in the last section.

It is possible to obtain still greater reliability by using a majority logic two-out-of-three configuration in the single decision element. This type of element is illustrated and discussed in Chapter IV. The equations developed in this section can be readily modified to accommodate this situation by substituting R'_V given in Equation (4.4.1) for R_V in Equation (3.6.1). This will not be done here since techniques rather than numerical results are of prime interest. However, a limiting value in reliability will still exist because the output signal must still pass through, in this case, a single voter similar to that shown in Figure 2.4.1. Such a device would consist of only approximately 20 component parts. The decision element considered in this section contained 400 to 650 parts, depending upon the degree of redundancy employed in the system; thus, the limiting value on failure probability would be expected to decrease considerably.

CHAPTER IV

PRACTICAL SYSTEM OPTIMIZATION WITH CONSTRAINTS

4.1 INTRODUCTION

In the previous chapters, optimization of an ideal system was considered. It was shown that for optimality, a system should be organized such that $\bar{R}_m \approx \frac{\bar{R}_v}{n-2}$ and that overall system reliability could be increased to a value as close to unity as desired, provided that system complexity could be tolerated and that the n redundant outputs could be utilized as inputs in the following system. If only one input could be accommodated by the next system, the failure probability of the system can be no less than that of the single decision element in the series chain. In addition, it was further proven that all the modules into which a nonredundant system was divided should have the same reliability and that the degree of redundancy applied to each module should be the same. These conditions lead to optimum system reliability. The question now arises as to the practicability of this approach and the underlying assumptions. If such an approach is not practical, how should a system be designed to achieve a reasonable gain in system reliability with reasonable or limited resources? This problem can be framed in two basic ways as follows:

1. Given a practical system (organized in a manner such that the modules do not have equal reliabilities), what is the maximum reliability which can be achieved within given complexity constraints such as cost, weight, power, etc?

2. The dual problem to 1. is that given a reliability requirement, how can this requirement be achieved to minimize the resources, cost, weight, power, etc?

Since it is quite likely that the reliability of the modules into which a nonredundant system can be naturally segmented will not be equal, it therefore follows that the degree of redundancy used at each stage may not necessarily be equivalent. This inherently introduces a new problem: that of interfacing or interconnecting n_i redundant elements in one stage to n_j redundant elements in the next stage. Notice that generally $n_i \neq n_j$. One method of accomplishing this function will be covered in detail.

There are several theoretical approaches which can be used to optimize a system under the above conditions. Classically, Lagrange multipliers might be used provided certain conditions, such as continuity and differentiability, are satisfied. It was shown in Table 3.4.1 and Figure 3.4.2 that in actual practice the reliability and number of component parts in the decision element are a discrete function of n_i, the degree of redundancy applied to the i^{th} stage, although for the development herein, a continuous approximation was used. Further, as the number of stages increase (they could conceivably approach a hundred), the Lagrange formulation becomes unwieldy.

Another technique which has been developed by Bellman (3) in recent years is the so-called dynamic, iterative, or recursive programming. This approach, which is numerical in nature and had to await the development of digital computer systems, circumvents the aforementioned problem of continuity and differentiability. In addition, a large number of stages can be adequately handled, provided sufficient computer capacity (time and memory) is available. From a practical point of view, however, the number of constraints which can be handled with this approach must be limited, again depending upon computer capacity and time available. The Lagrange formulation may also be coupled with the dynamic programming method to facilitate convergence.

A technique has been proposed by Sasaki (16) which he says leads to optimum reliability gain with a minimum expenditure of resources. However, he does not prove this will always be the case and essentially just states a decision algorithm. The first method to be discussed herein yields results similar to those of Sasaki; however, derivation will be given to show under what assumptions this approach should be used. In addition, Sasaki's algorithm in general does not lead to the most economical system. The second method developed herein leads to the most economical design.

Many figures of merit or criteria functions may be considered, and the final results possibly depend upon the one employed. Three basic criteria functions will be considered herein, and examples of each will be considered and the results compared. Many figures of merit have been proposed by various authors for numerous applications. A plausible criterion function to be first considered is simply $(\Delta P)_{max}$; i.e., redundancy should be added to a system to maximize the gain in overall system reliability. The decision algorithm proposed by Sasaki is to increase the reliability of the least reliable stage which is a special case of the general case $(\Delta P)_{max}$. It is not clear that Sasaki's approach or that the function $(\Delta P)_{max}$ leads to a system of minimum cost. Another criterion function that will also be discussed is $\left(\frac{\Delta P}{\Delta C}\right)_{max}$, which represents the ratio of the gain in system reliability to the increase in the overall system complexity. In essence, it represents the system reliability and cost gradient and should, therefore, be as large as possible. This function does indeed yield optimum reliability at minimum cost. Still, another possible criterion function that can be used is $\frac{\Delta P}{P} \Big/ \frac{\Delta C}{C}$, which is defined as the ratio of the percentage gained in reliability to the percentage increase in overall system complexity.

The basic approach in this chapter will be to develop theorems pertinent to the criteria functions and decision algorithms without initially considering the decision element. An example will then be worked using each of these criteria functions, and the results will be compared. A decision element necessary for utilizing n_i outputs with n_j inputs will be proposed and it will be shown how the decision element may be included in the previous development. Finally, the example will then be revisited incorporating the decision element design.

4.2 DEVELOPMENT OF CRITERIA FUNCTIONS AND DECISION ALGORITHMS

In this section, the criteria functions mentioned in the previous paragraph will be discussed and developed in more detail. To facilitate this development, several theorems will be stated and proven.

Theorem 1: In a system consisting of m serial elements with reliabilities $p_1, p_2, p_3 \ldots\ldots, p_m$, respectively, and a system reliability given by

$$P = p_1 \cdot p_2 \cdot p_3 \cdot\cdot\cdot\cdot\cdot\ p_m$$

when parallel elements are to be added to the system, maximum gain in system reliability is obtained by adding it such that $\dfrac{\Delta p_i}{p_i}$ is maximized.

Notice p_i is the reliability of the i^{th} stage which may either be nonredundant or redundant with at least degree three. If the redundant two-out-of-n approach is being assumed, it must contain at least three parallel elements. Thus, initially, it may be required to add two parallel modules.

Proof: Assume that the p_i's have been ordered such that $p_1 < p_2 < --- < p_m$. The system reliability is given by

$$P = p_1 \cdot p_2 \cdot p_3 \cdot \cdot \cdot \cdot \; p_m \quad . \tag{4.2.1}$$

Adding Δp to the i^{th} stage yields

$$P + \Delta P = p_1 \cdot p_2 \ldots (p_i + \Delta p_i) \ldots p_m \quad . \tag{4.2.2}$$

Notice that $\frac{P}{p_1} = p_2 \, p_3 \ldots\ldots p_m$ or that in general Equation (4.2.2) can be expressed as

$$\begin{aligned} P + \Delta P &= (p_i + \Delta p_i) \frac{P}{p_i} \\ \Delta P &= \frac{\Delta p_i}{p_i} P \quad . \end{aligned} \tag{4.2.3}$$

Since P is a constant at any step, the maximum $\frac{\Delta p_i}{p_i}$ therefore will yield maximum gain in system reliability, ΔP, and the theorem is proven.

Specifically, the two-out-of-n configuration is of basic interest herein. However, notice that Equation (4.2.3) is completely general and applicable to any redundant configuration. In Chapter III, it was shown that for a two-out-of-n configuration the reliability of a redundant stage is given by

$$p_i = (1 - R_i)^{n_i - 1} \left[1 + (n_i - 1) R_i\right] \quad . \qquad (4.2.4)$$

Adding another module to the i^{th} stage yields a new reliability given by

$$p_i' = 1 - (1 - R_i)^{n_i} \left[1 + n_i R_i\right] \quad . \qquad (4.2.5)$$

The reliability gained in the i^{th} stage by adding another module therefore is

$$\Delta p_i = p_i' - p_i = 1 - (1 - R_i)^{n_i} \left[1 + n_i R_i\right]$$

$$- \left\{1 - (1 - R_i)^{n_i - 1} \left[1 + (n_i - 1) R_i\right]\right\}$$

$$\Delta p_i = (1 - R_i)^{n_i - 1} \left\{1 + (n_i - 1) R_i - (1 - R_i) \left[1 + n_i R_i\right]\right\} \quad . \qquad (4.2.6)$$

Equation (4.2.6) may be simplified and written as

$$\Delta p_i = n_i (1 - R_i)^{n_i - 1} R_i^2$$

$$\Delta p_i = n_i \overline{R}_i^{\,n_i - 1} (1 - \overline{R}_i)^2 \quad . \qquad (4.2.7)$$

When higher-order terms are neglected, Δp_i is given approximately by

$$\Delta p_i \approx n_i \overline{R}_i^{\,n_i - 1} \quad . \tag{4.2.8}$$

Equation (4.2.4) shows that

$$p_i = 1 - \overline{R}_i^{\,n_i - 1} \left[n_i - (n_i - 1)\overline{R}_i \right] \quad .$$

Again, neglecting higher-order terms yields

$$p_i \approx 1 - n_i \overline{R}_i^{\,n_i - 1} \approx 1 - \bar{p}_i \quad . \tag{4.2.9}$$

Therefore, solving Equation (4.2.9) for $\bar{p}_i$ yields

$$\bar{p}_i \approx n_i \overline{R}_i^{\,n_i - 1} \tag{4.2.10}$$

and the ratio of the $\dfrac{\Delta p_i}{p_i}$ is given by the expression

$$\frac{\Delta p_i}{p_i} = \frac{\overline{p_i}}{1 - \overline{p_i}} = \frac{1}{\dfrac{1}{\overline{p_i}} - 1} \quad . \tag{4.2.11}$$

In this case, a maximum $\dfrac{\Delta p_i}{p_i}$ is obtained when the largest $\overline{p_i}$ is used; i.e., when the most unreliable stage is improved. This was the result obtained, or rather suggested, but not proven by Sasaki. However, it is not general and depends on the type of redundancy being utilized. For example, it does not apply when going from a nonredundant module to a two-out-of-three redundant module; i.e., when adding two more modules in parallel. Since, in this case, it can be shown that

$$\frac{\Delta p_i}{p_i} = \overline{R}_i - 2\overline{R}_i^{\,2} , \tag{4.2.12}$$

then $\left(\dfrac{\Delta p_i}{p_i}\right)_{max}$ occurs when $\overline{R}_i = \frac{1}{4}$. The general case of $\left(\dfrac{\Delta p_i}{p_i}\right)_{max}$ which is always applicable will be utilized as one case in the remaining work in this chapter.

It is interesting to notice also that when an equivalent Δp has been applied individually to each stage, the greatest gain in system reliability is still realized when it is applied to the least reliable stage. Since it was assumed that $p_1 < p_2 < p_3 \text{ ---- } < p_m$, and since

$$\Delta P_i = \frac{\Delta p_i}{p_i} P ,$$

it follows that

$$\frac{\Delta p}{p_1} P > \frac{\Delta p}{p_2} P > \frac{\Delta p}{p_3} P \ldots . > \frac{\Delta p}{p_m} P$$

and that $\Delta P_1 > \Delta P_2 > \text{ ---- } \Delta P_m$; i.e., the gain in system reliability is greatest when the least reliable stage is made more redundant. In practice, however, it would be difficult to apply an equal Δp to each stage since the nonredundant system possibly cannot be subdivided into modules of equal reliability.

Theorem 2: The ratio $\frac{\Delta P}{\Delta C}$ is maximized when $\frac{\Delta p_i}{\Delta n_i c_i p_i}$ is maximized, where c_i is the relative complexity of the i^{th} nonredundant module. In addition, redundancy is added in the most economical manner when this criterion is satisfied.

Proof: If complexity were being determined in terms of the number of component parts, it could be represented by

$$c_i = \frac{n_{mi}}{N_T}$$

where n_{mi} is the number of component parts in the i^{th} module and N_T the total number of parts in the redundant system. If weight were of concern, c_i would represent the weight of the i^{th} nonredundant module to the weight of the nonredundant system. In general, cost, weight, and power can have weighted values such that c_i can be expressed in the form

$$c_i = a \frac{u_i}{\sum_{i=1}^{m} u_i} + b \frac{v_i}{\sum_{i=1}^{m} v_i} + c \frac{w_i}{\sum_{i=1}^{m} w_i} \tag{4.2.13}$$

where $\sum_{i=1}^{m} u_i$, $\sum_{i=1}^{m} v_i$, and $\sum_{i=1}^{m} w_i$ represent the total nonredundant system cost, weight, and power, respectively, and u_i, v_i, and w_i the cost, weight, and power of the i^{th} module in the nonredundant system; a, b, and c are weighting factors representing relative importance of these factors. Thus, the complexity of the i^{th} redundant module is given by the expression

$$C_i = n_i c_i = n_i \left[a \frac{u_i}{\sum_{i=1}^{m} u_i} + b \frac{v_i}{\sum_{i=1}^{m} v_i} + c \frac{w_i}{\sum_{i=1}^{m} w_i} \right] . \qquad (4.2.14)$$

The change in system complexity by adding modules to the i^{th} stage is equivalent to the change in module complexity and is given by

$$\Delta C = \Delta n_i c_i = \Delta n_i \left[a \frac{u_i}{\sum_{i=1}^{m} u_i} + b \frac{v_i}{\sum_{i=1}^{m} v_i} + c \frac{w_i}{\sum_{i=1}^{m} w_i} \right] . \qquad (4.2.15)$$

Theorem 1 shows that $\Delta P = \frac{\Delta p_i}{p_i} P$; thus, dividing this by Equation (4.2.15) yields

$$\frac{\Delta P}{\Delta C} = \frac{\Delta p_i P}{\Delta n_i c_i p_i}$$

and since P is a constant for each step, the desired result is obtained. The fact that the criterion leads to the most economy follows directly from the observation that the reliability/cost gradient is optimized at each step; thus, the resulting system must necessarily yield the maximum reliability which can be obtained within given cost constraints or conversely minimum costs which are necessary to achieve a given reliability requirement.

Theorem 3: The ratio $\frac{\Delta P}{P} \Big/ \frac{\Delta C}{C}$ is maximized when $\frac{\Delta P}{\Delta C}$ is maximized.

Proof:

$$\frac{\frac{\Delta P}{P}}{\frac{\Delta C}{C}} \doteq \frac{\Delta PC}{P\Delta C}$$

and since $\frac{C}{P}$ is a constant at each step, the desired results follow immediately.

Thus, from theorems 2 and 3, it is observed that both $\frac{\Delta P}{\Delta C}$ and $\frac{\Delta P}{P} \Big/ \frac{\Delta C}{C}$ are maximized when $\frac{\Delta p_i}{\Delta n_i c_i p_i}$ is maximized.

Two theorems due to Sasaki (16), although not directly pertinent to the developments herein, possibly are of passing interest and therefore will be included.

Theorem 4: Assume that $p_i < p_j$. If p_i is increased by Δp_i, the overall system reliability may alternately be increased by an equivalent amount when p_j is increased by

$$\Delta p_j = \frac{\Delta p_i \, p_j}{p_i} \quad .$$

Proof: Assume that Δp_i and Δp_j are added to the i^{th} and j^{th} stages, respectively, such that the overall system reliability gain is the same in each case. Further assume that the system has been ordered such that

$$p_1 < p_2 < p_3 ---- < p_m \quad .$$

Then

$$p_1 \cdot p_2 \cdot \cdot \cdot \cdot \; p_{i-1} \, (p_i + \Delta p_i) \cdot \cdot \cdot \cdot \; p_j \cdot \cdot \cdot \cdot \; p_m$$

$$= p_1 \cdot p_2 \cdot p_3 \cdot \cdot \cdot \; p_{i-1} \, p_i \cdot \cdot \cdot \cdot \; p_{j-1} \, (p_j + \Delta p_j) \cdot \cdot \cdot \cdot \; p_m \quad ; \tag{4.2.16}$$

therefore,

$$(p_i + \Delta p_i) p_j = p_i (p_j + \Delta p_j)$$

$$\Delta p_i \, p_j = p_i \, \Delta p_j$$

and the desired results that

$$\Delta p_j = \frac{\Delta p_i \, p_j}{p_i} \tag{4.2.17}$$

follow immediately.

Theorem 5: Assume that $p_i < p_j$. If p_i is increased by Δp_i and the result is such that

$$p_j + \frac{\Delta p_i \, p_j}{p_i} > 1 \qquad ,$$

then the gain in system reliability is greater when p_i is increased by Δp_i than when p_j is made unity.

Proof: The reliability of the j^{th} stage must be between 0 and 1, i.e.,

$$p_j + \Delta p_j < 1 \qquad .$$

The above inequalities can only be satisfied if

$$\frac{\Delta p_j}{p_j} < \frac{\Delta p_i}{p_i} \qquad ,$$

and the desired result is obtained.

4.3 ILLUSTRATION OF UTILIZATION OF CRITERIA FUNCTIONS AND DECISION ALGORITHMS

Thus far, nothing has been said of a decision element which can accomplish the required function of interconnecting the n_i stage outputs to the n_j inputs of the next stage and, as yet, no consideration has been given to the incorporation of a decision element in the above theorems; this will be discussed later. However, it is instructive and beneficial at this point to illustrate how these theorems can be used in design optimization. From the theorems, it is obvious that redundant elements are to be added to one stage at a time so as to maximize one of the parameters.

$$(\Delta P)_{max} = \left(\frac{\Delta p_i}{p_i}\right)_{max}$$

$$\left(\frac{\Delta P}{\Delta C}\right)_{max} = \left(\frac{\Delta P/P}{\Delta C/C}\right)_{max} = \left(\frac{\Delta p_i}{\Delta n_i\, c_i\, p_i}\right)_{max} .$$

The first function establishes a procedure of adding redundant elements to give maximum gain in system reliability. In the second function, primary emphasis is placed on adding modules such that the reliability/complexity gradient is maximized. Which of these methods one wishes to use depends on what one is after. Probably a more interesting question is: Do these criteria functions lead to the same results?

As an example of the utilization of the criteria functions which result in decision algorithms, consider a nonredundant system with parameters given in Table 4.3.1. The problem is to optimize the reliability of the redundant system within the constraints of a total cost not to exceed 99, a total weight less than 57, and a total power less than 83. It should be pointed out that the units on these can be dimensionless. Thus, the final system cost can be no greater than 99/21 of the initial cost, final system weight no more than 57/12, the initial weight, etc. Also, the problem of minimizing costs, weight, and power necessary to achieve a system reliability goal of 0.9995 will be treated for illustrative purposes.

TABLE 4.3.1

PARAMETERS OF A HYPOTHETICAL NONREDUNDANT SYSTEM

Stage	Nonredundant Module Reliability	Module Cost (u_i)	Module Weight (v_i)	Module Power (w_i)
1	0.99943	1	1	1
2	0.94064	2	1	3
3	0.88185	3	2	2
4	0.82306	4	2	2
5	0.76427	5	3	4
6	0.70548	6	3	5
Total System	0.36790	21	12	17

This problem will be solved with each criterion function developed previously. The first criterion function considered is

$$(\Delta P)_{max} = \left(\frac{\Delta p_i}{p_i}\right)_{max} .$$

The iterative process or decision algorithm to determine the maximum reliability within the constraint condition is as follows:

1. For each possible stage change, calculate $\frac{\Delta p_i}{p_i}$. Since a two-out-of-n configuration is being assumed, the initial increment will be to add two modules to the stages as determined by

$$\left(\frac{\Delta p_i}{p_i}\right)_{max} \quad .$$

2. The new complexity value is calculated from $\sum_{i=1}^{m} n_i u_i$, $\sum_{i=1}^{m} n_i v_i$, and $\sum_{i=1}^{m} n_i w_i$ where n_i is the number of modules employed by the i^{th} stage, and u_i , v_i , and w_i are the cost, weight, and power values associated with the i^{th} module.

3. The process continues as long as

$$K_u - \sum_{i=1}^{m} n_i u_i \geq \Delta n_i \left(u_i\right)_{min}$$

$$K_v - \sum_{i=1}^{m} n_i v_i \geq \Delta n_i \left(v_i\right)_{min}$$

$$K_w - \sum_{i=1}^{m} n_i w_i \geq \Delta n_i \left(w_i\right)_{min} \quad ,$$

where K_u is the u^{th} constraint, etc. In this example, $K_u = 99$, $K_v = 57$, and $K_w = 83$.

d. After two modules have been initially assigned to a stage when its $\left(\frac{\Delta p_i}{p_i}\right)_{max}$ occurs, only one module at a time is assigned to a redundant element. In other words, if a nonredundant module is chosen by the criterion function to be made redundant, two modules are initially assigned to it. Thereafter, only one additional parallel module can be assigned to that redundant stage at any particular step.

Thus, the first step is to calculate $\frac{\Delta p_i}{p_i}$ for each possible stage change. Since no redundancy has been added to the system, this value is given in general by

$$\frac{\Delta p_i}{p_i} = \frac{3R_i^2 - 2R_i^3 - R_i}{R_i} = 3R_i - 2R_i^2 - 1 \qquad (4.3.1)$$

or expressing Equation (4.3.1) in terms of $\overline{R}_i$ and simplifying yields

$$\frac{\Delta p_i}{p_i} = \overline{R}_i \left(1 - 2\overline{R}_i\right) \quad . \qquad (4.3.2)$$

The $\frac{\Delta p_i}{p_i}$ for the various stages has been calculated and is given as follows:

$$\frac{\Delta p_1}{p_1} = 0.00057 \qquad \frac{\Delta p_4}{p_4} = 0.11432$$

$$\frac{\Delta p_2}{p_2} = 0.05231 \qquad \frac{\Delta p_5}{p_5} = 0.12459$$

$$\frac{\Delta p_3}{p_3} = 0.09023 \qquad \frac{\Delta p_6}{p_6} = 0.12104$$

(Notice that for this initial step, Equation (4.2.12) indicates that maximum $\frac{\Delta p_i}{p_i}$ occurs for $\overline{R}_i$ closest to 0.2500; thus, the initial maximum $\frac{\Delta p_i}{p_i}$ could have been determined by inspection.)

Since $\frac{\Delta p_5}{p_5}$ is the largest value, and since this is the initial assignment to that unit, two modules will be added. It will then have a failure probability given by

$$\bar{p}_5 = 3\overline{R}_5^{\,2} - 2\overline{R}_5^{\,3} = 0.14051 \quad .$$

The remaining resources to be allocated are

$$99 - \sum_{i=1}^{6} n_i u_i = 99 - 31 = 68 \geq 1$$

$$57 - \sum_{i=1}^{6} n_i v_i = 57 - 18 = 39 \geq 1$$

$$83 - \sum_{i=1}^{6} n_i w_i = 83 - 25 = 58 \geq 1 \quad .$$

Notice that since this is greater than $\Delta n_i(c_i)_{min} = 1$, other allocations are possible.

A new $\frac{\Delta p_5}{p_5}$ is calculated by adding one module to the 5^{th} stage. In general, for one additional module, Δp_i is given by Equation (4.2.7) which is

$$\Delta p_i = n_i \left(\overline{R}_i\right)^{n_i - 1} \left(1 - \overline{R}_i\right)^2 .$$

Thus, for the 5^{th} stage Δp_5 is 0.09738 and $\frac{\Delta p_5}{p_5}$ is 0.11329. This value is compared to those previously calculated for the other stages, the largest value is chosen, and that stage's degree of redundancy is increased by one or two, depending on whether this is the first allocation to that stage.

The $\frac{\Delta p_6}{p_6} = 0.12104$ is now observed to be maximum; therefore, two modules are added to it giving a failure probability of $\bar{p}_6 = 0.20912$. The resources available for allocation after two modules have been added to the 6^{th} stage are

$$99 - \sum_{i=1}^{6} n_i u_i = 99 - 43 = 56$$

$$57 - \sum_{i=1}^{6} n_i v_i = 57 - 24 = 33$$

$$83 - \sum_{i=1}^{6} n_i w_i = 83 - 35 = 48 \quad .$$

The sequence in which modules are added to the various stages is shown in Figure 4.3.1. The circled numbers indicate the step in the process. For example, the first step is to increase the number of modules in stage 5 from one to three, the second step is to increase the number of modules in stage 6 to three, etc. The value of the criterion function, $\frac{\Delta p_i}{p_i}$, at each step is also shown in this figure. The dashed line (step 13) indicates that this stage was selected to be made more redundant, but that a constraint would have been exceeded if this were done. Therefore, the largest value of the criterion function which does not exceed the constraints is chosen. The final system configuration is given in Table 4.3.2.

The total overall system reliability is 0.951709 and the total cost, weight, and power are 98, 54, and 79 units, respectively. The relative cost, weight, and power when compared to that of the nonredundant system are 4.67, 4.50, and 4.65, respectively. Appendix B.1 gives more detail about the computer program which was used in the optimization process and presents the detailed parameters of the system after each step.

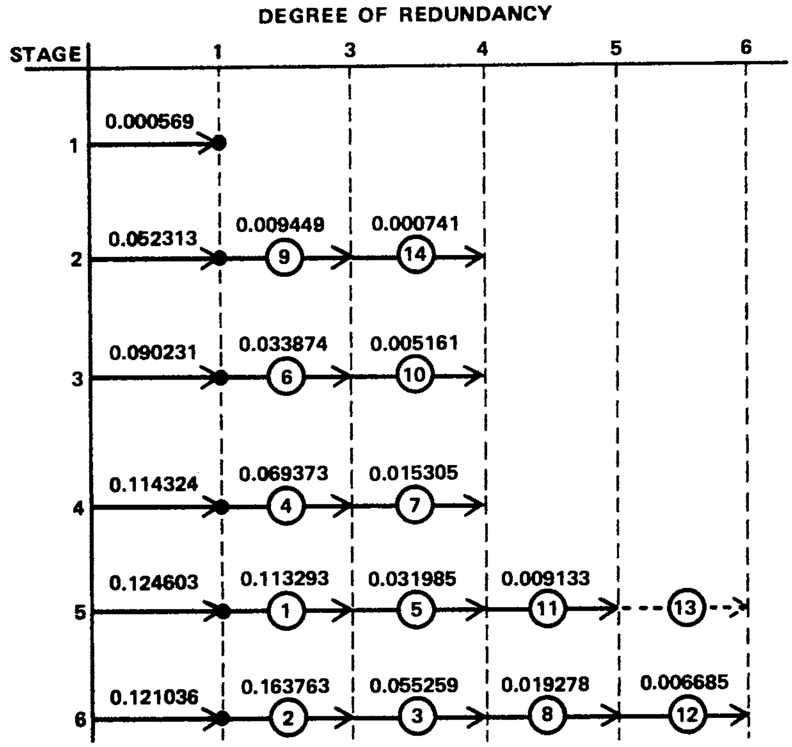


Figure 4.3.1. Optimization Sequence for Criterion Function

$$(\Delta P)_{max} = \left(\frac{\Delta p_i}{p_i}\right)_{max}$$

This approach and computer program can also be used in achieving a specific reliability goal. In this case, the constraint conditions are removed and the process is continued until the goal has been reached. The parameters of the system are then read out of the program at this point. As an example, it is desirable to determine the cost, weight, and power in achieving a reliability goal of 0.9995. Detailed data are also given in Appendix B.1. The process proceeds initially as in the previous example, but is carried on for several more steps. The final system configuration and parameters are summarized in Table 4.3.3.

TABLE 4.3.2

SUMMARY OF RESULTS FOR CRITERION FUNCTION

$$(\Delta P)_{max} = \left(\frac{\Delta p_i}{p_i}\right)_{max}$$

Stage	Number of Modules	Stage Reliability
1	1	0.999430
2	4	0.999201
3	4	0.993987
4	4	0.980782
5	5	0.987472
6	6	0.989967
P (Overall System Reliability) = 0.951709		
$\sum n_i u_i$ (cost) = 98		
$\sum n_i v_i$ (weight) = 54		
$\sum n_i w_i$ (power) = 79		

TABLE 4.3.3

SUMMARY OF RESULTS FOR ACHIEVING A RELIABILITY GOAL OF 0.9995 FOR BOTH CRITERIA FUNCTIONS

Stage	Number of Modules	Stage Reliability
1	3	0.999999
2	5	0.999941
3	6	0.999875
4	8	0.999963
5	9	0.999932
6	10	0.999877

P = 0.999588

Cr (cost) = 8.0 (168 units)

Cr (weight) = 7.75 (93 units)

Cr (power) = 7.76 (132 units)

Figure 4.3.2 indicates how $\bar{P}$ varies with cost for each step in the process. The other parameters, weight and power, have very similar shaped curves, and it is not worthwhile presenting them. Table 4.3.3 indicates also that the relative complexities in cost, weight, and power are nearly equal at the conclusion of the process.

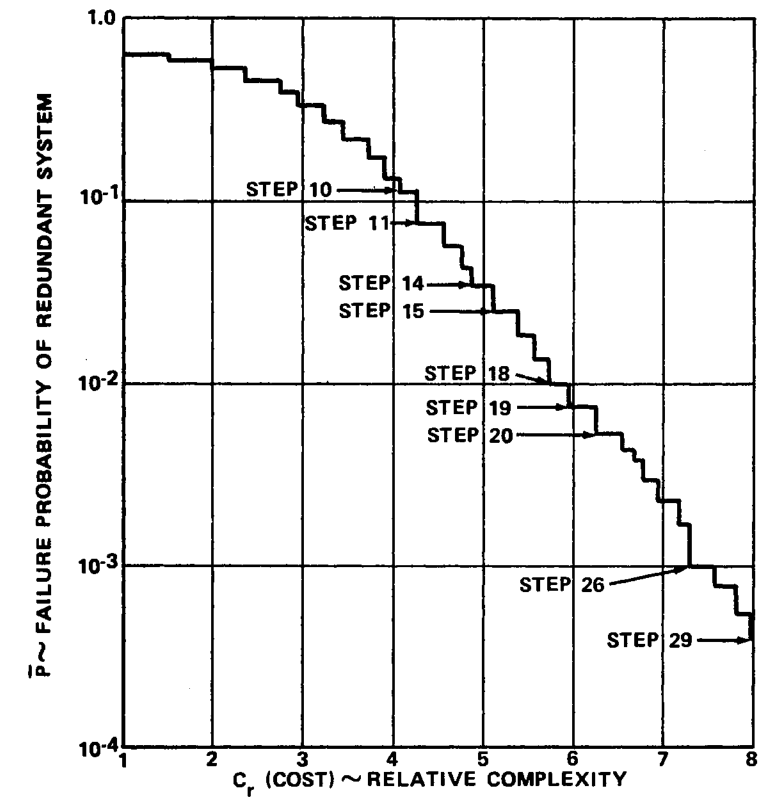


Figure 4.3.2. $\overline{P}$ Versus C_r (cost) to Obtain a Reliability Goal of 0.9995 with the Criterion Function

$$(\Delta P)_{max} = \left(\frac{\Delta p_i}{p_i}\right)_{max}$$

The example will now be solved, using the criterion function $\left(\dfrac{\Delta p_i}{\Delta n_i\, c_i\, p_i}\right)_{max}$ where values of $a = b = c = 0.3333$ are used in c_i which was calculated from Equation (4.2.13). The steps in the process along with the value of the criterion function at each step are shown in Figure 4.3.3.

Detailed data concerning variations of system parameters in the process are presented in Appendix B.2, and a summary of the final results is given in Table 4.3.4.

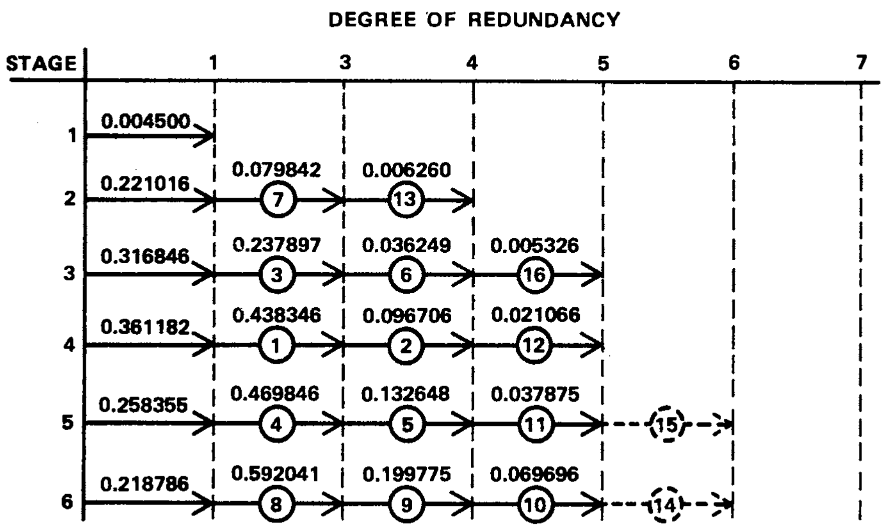


Figure 4.3.3. Optimization Sequence for Criterion Function

$$\left(\frac{\Delta P}{\Delta C}\right)_{max} = \left(\frac{\Delta p_i}{\Delta n_i\, c_i\, p_i}\right)_{max}$$

The results obtained in utilizing this criterion function to achieve a reliability goal of 0.9995 are identical to those given in Table 4.3.3. However, a comparison of Figures 4.3.1 and 4.3.3 indicates that the

TABLE 4.3.4

SUMMARY OF RESULTS FOR CRITERIA FUNCTIONS

$$\left(\frac{\Delta P}{\Delta C}\right)_{max} = \left(\frac{\Delta p_i}{\Delta n_i\, c_i\, p_i}\right)_{max}$$

Stage	Number of Modules	Stage Reliability
1	1	0.999430
2	4	0.999201
3	5	0.999118
4	5	0.995793
5	5	0.987472
6	5	0.971243
$P = 0.952892$		
$\sum n_i u_i$ (cost) = 99		
$\sum n_i v_i$ (weight) = 55		
$\sum n_i w_i$ (power) = 78		

steps in arriving at this common solution are quite different. Detailed data concerning the process may again be found in Appendix B.2, and Figure 4.3.4 indicates how $\overline{P}$ varies with relative cost.

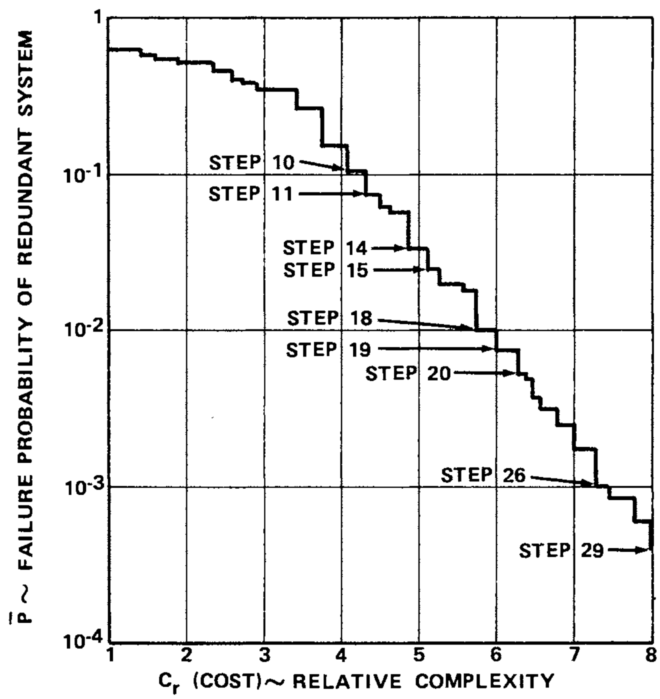


Figure 4.3.4. $\overline{P}$ Versus C_r (Cost) to Obtain a Reliability Goal of 0.9995 with the Criterion Function

$$\left(\frac{\Delta P}{\Delta C}\right)_{max} = \left(\frac{\Delta p_i}{\Delta n_i \, c_i \, p_i}\right)_{max}$$

It is desirable to compare the two criteria functions which have been developed and the results obtained from the dual problem; i.e., of maximizing reliability within given constraints and minimizing complexity in achieving a reliability goal. For comparing the results of maximizing reliability in the presence of constraints, Tables 4.3.2 and 4.3.4 may be used along with the sequencing information contained in Figures 4.3.1 and 4.3.3. From Tables 4.3.2 and 4.3.4 it is noted that the final configurations differ; 1, 4, 4, 4, 5, and 6 modules are used in the consecutive stages when $\left(\frac{\Delta p_i}{p_i}\right)_{max}$ is utilized while 1, 4, 5, 5, 5, and 5 modules are used in the consecutive stages when $\left(\frac{\Delta p_i}{\Delta n_i\, c_i\, p_i}\right)_{max}$ is used. Therefore, the resulting reliabilities differ, 0.951709 found from $\left(\frac{\Delta p_i}{p_i}\right)_{max}$ as compared to 0.952892 from $\left(\frac{\Delta p_i}{\Delta n_i\, c_i\, p_i}\right)_{max}$.

The ratio of $\left(\frac{\Delta p_i}{p_i}\right)_{max}$: $\left(\frac{\Delta p_i}{\Delta n_i\, c_i\, p_i}\right)_{max}$ for cost, weight, and power is 98:99, 54:55, and 79:78, respectively. It is significant to notice that with $\left(\frac{\Delta p_i}{\Delta n_i\, c_i\, p_i}\right)_{max}$ the constraint in cost was reached but not in the other cases. In general, this criterion utilized more resources, although this is untrue in the case of power, so possibly this is the reason why a higher reliability is obtained. Figure 4.3.1 shows the sequence in which modules are added to the stages is 5, 6, 6, 4, 5, 3, 4, 6, 2, 3, 5, 6, and 2 for $\left(\frac{\Delta p_i}{p_i}\right)_{max}$ and Figure 4.3.3 shows 4, 4, 3, 5, 5, 3, 2, 6, 6, 6, 5, 4, 2, and 3 for $\left(\frac{\Delta p_i}{\Delta n_i\, c_i\, p_i}\right)_{max}$. Thus, $\left(\frac{\Delta p_i}{p_i}\right)_{max}$ initially concentrates more on the most unreliable modules while $\left(\frac{\Delta p_i}{\Delta n_i\, c_i\, p_i}\right)_{max}$

is initially more concerned with modules of intermediate reliability and does not select the most unreliable modules until later in the process.

Figures 4.3.2 and 4.3.4 illustrate the results obtained with the two methods in obtaining a reliability goal of 0.9995. The final result for both methods was identical and is summarized in Table 4.3.3. However, it may be concluded from Figures 4.3.2 and 4.3.4 where the steps at which the results are identical are marked (i.e., steps 10, 11, 14, 15, 18, 19, 20, 26, and 29) that it was only coincidental that the final results agree. The number of steps required to reach the design goal was the same; therefore, it may be concluded that one method does not converge any faster than the other. Also, the shapes of the curves are similar and fall very close to each other. Generally, at each step the reliability of $\left(\frac{\Delta p_i}{p_i}\right)_{max}$ is higher but the relative complexity is also higher, which is simply the nature of the two functions.

The subject of optimality in many cases must be treated more qualitatively than quantitatively. For instance, there may be little debate about the outcome of an optimization process once a particular criterion has been selected; however, at the present time, there is no universally accepted criterion function which serves all optimization processes. Any criterion function must be tailored to one's particular aims, needs, and goals. For this reason, this section as well as the next may not appear to be mathematically rigorous. The criteria functions themselves have been developed with a fair degree of mathematical rigor; however, the question of what constitutes a good criterion function still remains. Much more work is required and this area is recommended for further research.

From the examples, can any general conclusion be drawn, and is one approach preferable to the other? Some conclusions which may be drawn are:

1. The criterion $(\Delta P)_{max}$ yields the steepest ascent for increasing system reliability, and at each step has at least as high a system reliability as the other method. However, this is no surprise since it was designed specifically for that purpose.

2. For the function $\left(\frac{\Delta P}{\Delta C}\right)_{max}$, the steepest ascent approach has been tempered somewhat to take into consideration system complexity. The result is that generally the relative complexity is always less than or equal to that obtained with the other method.

3. By maximizing reliability in the presence of constraints, maximizing $\frac{\Delta P}{\Delta C}$ results in a higher system reliability because the resources were more efficiently utilized.

4. In obtaining a reliability goal such that resources are minimized, the same results were obtained. However, this is only coincidental as can be seen by comparing the results at each step. The rapidity of convergence of the two methods is quite close.

5. The criterion $\left(\frac{\Delta P}{\Delta C}\right)_{max}$ allows the constraints to be weighted, thus all constraints are considered simultaneously. Weighting of constraints may be important since they may not always have the same criticality. The value of the constraints imposed on the system can conceivably be independent of the criticality of the constraint.

From the above, it is concluded that the $\left(\frac{\Delta P}{\Delta C}\right)_{max}$ criterion is superior to $(\Delta P)_{max}$ since consideration is specifically given to system economy. The reliability which can be gained may not always exceed that obtained with $(\Delta P)_{max}$, but the ratio of gain in system reliability to the increase in system complexity is always assured to be greater.

4.4 A DECISION ELEMENT FOR STAGES WITH DIFFERENT DEGREES OF REDUNDANCY

The theorems which have been developed include no provision for decision element reliability, unless possibly it can be lumped with the p_i's. If this cannot be done, then the above theorems must be modified so that the reliability of this element is taken into consideration. At this point in the development, it is advantageous to consider the logical characteristics of the decision element.

In the previous chapter, the logic was developed for utilizing n outputs of one stage with n inputs of the next stage. In other words, the stages employed the same degree of redundancy. In this chapter, this restriction is removed and the formidable problem of adapting n_i outputs of one stage as n_j inputs of the next stage is encountered. This basic problem is aggravated considerably because the problems of fault masking, failure detection and isolation, module switching, etc., now become embedded in the overall problem.

Several approaches to the problem are possible. The approach which is most appropriate must be tailored to the specific application. One approach which could be used is to employ a single decision element similar to that developed in Chapter II. The system organization suggested in Chapter III, however, employed as many decision elements as modules for the obvious reason of deleting single point failures, although considerable expense may be encountered in doing this. As module-to-decision element complexity increases, a single decision element has a decreasing effect on the overall system reliability. Thus, the question of whether to use a single element or n_i elements can only be answered when the overall system design and application have been considered and when tradeoffs have been made in system reliability and complexity. A single decision element is not proposed here because of the possibility of single point failures and because such an approach evades the basic problem. On the other hand, n_i decision elements (i.e., one with each module) will not be proposed because of system complexity and because it does not solve the basic interconnection problem.

The decision and switching element to be considered here is a compromise between single point failures and complexity and offers a feasible solution to the interconnection problem. It utilizes the two-out-of-n approach; however, it is suggested that n be limited to three. Unless n is limited to three, the age-old question of "who checks the checker" arises, and the interconnection problem still exists. A block diagram of the redundant majority logic decision element is shown in Figure 4.4.1. A decision element as previously designed has been triplicated; the output gating which is fed back to the module selection logic has thus been previously voted. The outputs to the next stage have also been voted with inputs from different channels. Since there are three $\overline{X}$'s, three $\overline{Y}$'s, and three $\overline{Z}$'s, 3^3 or 27 different combinations are possible; i.e., 27 outputs are possible which have been derived from differently voted signals. A one-to-one correspondence may be noted in Figures 4.4.1 and 3.4.2. Twelve voters per channel or a total of 36 are required in the decision element, plus one additional voter per output.

If a voted feedback control signal is in error, then another module is switched into that particular channel; however, it should

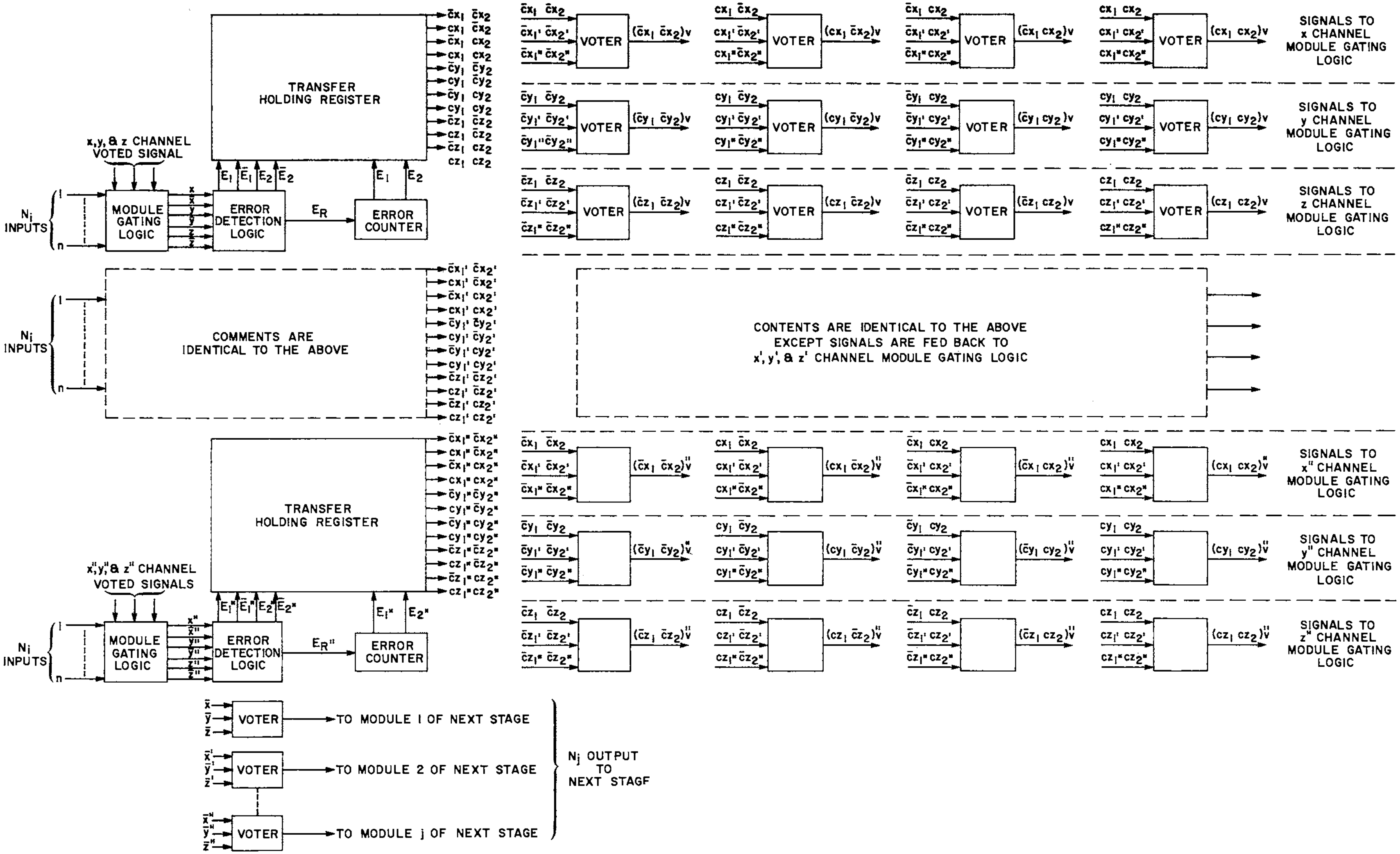


Figure 4.4.1. Triple Modular Redundant Decision Element

not affect system operation as long as the other two channels are functioning properly. A failure in a voter which feeds the next stage would effectively result in the decision element in the next stage detecting the error as a failure in one of its modules and then switching out that module. Since the decision element is primarily two-out-of-three logic, each element would have a reliability closely approximated by

$$R_v' = 3R_v^2 - 2R_v^3 \tag{4.4.1}$$

where R_v is the reliability of a single decision element and must be modified slightly to include the additional voters. It is, therefore, noted that R_v and consequently R_v' will be a function of the number of component parts utilized in the i^{th} redundant stage.

With this method of implementation, it is necessary that the previously developed criteria functions be modified to include the decision element before design optimization is possible. With this approach, the basic theorems are applicable when p_i and c_i have been modified to include decision element parameters.

Since $R_v = e^{-n_v \lambda t}$, when $t = \frac{k}{\lambda N_T}$ the reliability of one channel of the decision element is given by

$$R_v = e^{-\frac{kn_v}{N_T}} \tag{4.4.2}$$

where $n_v = (243 + 3n)\, \ell n_2\, (n-2) + 12n + 350$, as shown in Chapter III, except the constant 108 has been increased to 350 to account for the 12 voters in each channel. Each voter is assumed to contain approximately 20 components. Thus, the reliability of the redundant decision element is found from Equations (4.4.1) and (4.4.2) to be

$$R'_{v_i} = 3e^{-\frac{2kn_{v_i}}{N_T}} - 2e^{-\frac{3kn_{v_i}}{N_T}} \quad . \tag{4.4.3}$$

The normalized complexity of one decision element channel when $k = 1$ is given by

$$c_{v_i} = \frac{(243 + 3n_i)\, \ell n_2\, (n_i - 2) + 12n_i + 350}{N_T} \quad . \tag{4.4.4}$$

For the total redundant decision element, the relative complexity is approximately three times that of a single channel and is given by

$$c_{v_i} = \frac{3\left[(243 + 3n_i)\, \ell n_2\, (n_i - 2) + 12n_i + 350\right]}{N_T} \quad . \tag{4.4.5}$$

Thus, when consideration is given to the decision element, the relative complexity of a redundant stage is given approximately by

$$C_i = a \left\{ \frac{u_i}{\sum_{i=1}^{m} u_i} + \frac{3\left[(243 + 3n_i)\,\ell n_2\,(n_i - 2) + 12n_i + 350\right]}{N_T} \right\}$$

$$+ b \left\{ \frac{v_i}{\sum_{i=1}^{m} v_i} + \frac{3\left[(243 + 3n_i)\,\ell n_2\,(n_i - 2) + 12n_i + 350\right]}{N_T} \right\} \tag{4.4.6}$$

$$+ c \left\{ \frac{w_i}{\sum_{i=1}^{m} w_i} + \frac{3\left[(243 + 3n_i)\,\ell n_2\,(n_i - 2) + 12n_i + 350\right]}{N_T} \right\} .$$

Equation (4.4.6) can be simplified to yield

$$C_i = 3(a + b + c) \left[\frac{(243 + 3n_i)\,\ell n_2\,(n_i - 2) + 12n_i + 350}{N_T} \right]$$

$$+ a \frac{u_i}{\sum_{i=1}^{m} u_i} + b \frac{v_i}{\sum_{i=1}^{m} v_i} + c \frac{w_i}{\sum_{i=1}^{m} w_i} \tag{4.4.7}$$

If a, b, and c are normalized such that $a + b + c = 1$, then Equation (4.4.7) can be written as

$$C_i = 3\left[\frac{(243 + 3n_i)\,\ell n_2\,(n_i - 2) + 12n_i + 350}{N_T}\right] + a\frac{u_i}{\sum_{i=1}^{m} u_i} + b\frac{v_i}{\sum_{i=1}^{m} v_i} + c\frac{w_i}{\sum_{i=1}^{m} w_i} \tag{4.4.8}$$

In Equations (4.4.6) through (4.4.8), it has been assumed that the cost, weight, and power of the decision element are linearly proportional to the number of component parts contained in this element. In any case, if the cost, weight, and power of the decision element are known as functions of the number of component parts employed in the element, then these can be incorporated in the above expressions.

It is desirable for computational purposes to use $\frac{\Delta P}{\Delta C}$ in a slightly different form than has heretofore been used. The complexity of the i^{th} stage is given by Equation (4.4.8). The change in system complexity, resulting from adding additional modules to the i^{th} stage, is identical to the change in the i^{th} stage complexity and is given by

$$\Delta C = \Delta C_i = \Delta n_i \left[a\frac{u_i}{\sum_{i=1}^{m} u_i} + b\frac{v_i}{\sum_{i=1}^{m} v_i} + c\frac{w_i}{\sum_{i=1}^{m} w_i}\right]$$

$$+ \frac{3}{N_T} \left\{ \left[(243 + 3n_i) \ell n_2 \, (n_i - 2) + 12n_i + 350 \right] - \left[(243 + 3n_{i-1}) \, \ell n_2 \, (n_{i-1} - 2) + 12n_{i-1} + 350 \right] \right\} \tag{4.4.9}$$

where the last term is the ratio of change in decision element complexity to the complexity of the entire nonredundant system.

In theorem 1, it was shown that

$$\Delta P = \frac{\Delta p_i}{p_i} P \qquad ;$$

thus,

$$\frac{\Delta P}{\Delta C} = \frac{\Delta p_i}{\left(\Delta n_i \, c_i + \frac{3 \Delta n_v}{N_T} \right) p_i} P \tag{4.4.10}$$

where

$$c_i = a \frac{u_i}{\sum_{i=1}^{m} u_i} + b \frac{v_i}{\sum_{i=1}^{m} v_i} + c \frac{w_i}{\sum_{i=1}^{m} w_i}$$

as before, and

$$\Delta n_v \doteq (243 + 3n_i)\, \ell n_2\, (n_i - 2) + 12n_i + 350$$

$$- \left[(243 + 3n_{i-1})\, \ell n_2\, (n_{i-1} - 2) + 12n_{i-1} + 350\right] \quad . \tag{4.4.11}$$

From Chapter III, $\frac{\partial n_v}{\partial_n}$ was determined by Equation (3.5.2) and it follows that

$$\Delta n_v \approx \left[3\, \ell n_2\, (n_i - 2) + \frac{1.44\,(243 + 3n)}{n_i - 2} + 12\right] \Delta n_i \quad .$$

Since Δn_v is calculated by a digital computer as an iterative process, this approximation is not necessary; therefore, the former expression will be used. Since P is the system reliability before additional modules are added to the i^{th} stage and is therefore constant at any step in the optimization process, the criterion function can be expressed as

$$\left(\frac{\Delta P}{\Delta C}\right)_{max} = \left[\frac{\Delta p_i}{\left(\Delta n_i\, c_i + \frac{3\Delta n_v}{N_T}\right) p_i}\right]_{max} \tag{4.4.12}$$

where c_i and Δn_i are as previously given.

It is also desirable to express the overall relative complexity constraint in terms of individual weighted constraints. This is very important for two reasons: (1) all constraints are taken into consideration simultaneously, and (2) the criticality of each constraint can be weighted to take into consideration its relative importance. This can be accomplished with the relationship

$$C_r(\text{constraint}) = \frac{ac_u + bc_v + cc_w}{a \sum_{i=1}^{m} u_i + b \sum_{i=1}^{m} v_i + c \sum_{i=1}^{m} w_i} \tag{4.4.13}$$

where c_u , c_v , and c_w are the constraints of the redundant system, including the decision element in cost, weight, and power, respectively; a, b, and c are weighting factors indicating relative importance of constraints c_u , c_v , and c_w , respectively; and $\sum_{i=1}^{m} u_i$, $\sum_{i=1}^{m} v_i$, and $\sum_{i=1}^{m} w_i$ are the u, v, and w nonredundant system parameters assumed herein to be cost, weight, and power, respectively. In the previous example, which is to be reexamined, $c_u = 99$, $c_v = 57$, $c_w = 83$, $\sum_{i=1}^{6} u_i = 21$, $\sum_{i=1}^{6} v_i = 12$, and $\sum_{i=1}^{6} w_i = 17$. With

$a = b = c = 0.333$ as before, the overall relative complexity constraint is determined to be

$$C_r(\text{constraint}) = \frac{0.333\ (99 + 57 + 83)}{0.333\ (21 + 12 + 17)} = \frac{239}{50} = 4.78 \quad .$$

The reliability required in the criterion function is the product of the reliability of the stage [derived from Equation (2.2.2)] and the reliability of the decision element [Equation (4.4.3) with $k = 1$], and is given by

$$p_i = \left\{1 - (1-R_i)^{n_i-1}\left[1 + (n_i-1)R_i\right]\right\}\left\{3e^{-\frac{2n_{v_i}}{N_T}} - 2e^{-\frac{3n_{v_i}}{N_T}}\right\} \tag{4.4.14}$$

where n_{v_i} is a function of n_i as previously shown. Notice, however, that as given in Equation (4.4.14), the reliability of the decision element has been normalized about $t = \frac{1}{\lambda N_T}$; i.e., the nonredundant system has a reliability of 0.368 when $t = \frac{1}{\lambda N_T}$ or $N_T \lambda t = 1$. If N_T is increased in this equation and R is assumed to be constant, this would have the effect of decreasing λt. If λ is considered to be constant, then an increase in N_T results in a decrease in t . If t is assumed constant, then an increase in N_T results in a decrease in λ .

Therefore, caution must be used in the application of Equation (4.4.14) in which R_i is the i^{th} module reliability at the mtbf of the redundant system. The example considered in the previous section was designed with this in mind.

Equations (4.4.9) and (4.4.14) therefore allow the criterion function Equation (4.4.10) to be evaluated giving consideration to the incorporation of the decision element. Notice, however, that an additional system parameter, N_T, has been introduced and will be assumed to be known. For illustration, it will be assumed that $N_T = 50,000$ and the previous example will be revisited. Assuming that $N_T = 50,000$ implies that $\lambda t = 2 \times 10^{-5}$.

To solve the problems of designing a redundant system to yield maximum reliability within an overall relative complexity of $C_r(\text{constraint}) = 4.78$, and in achieving a reliability goal of 0.9995 with minimum expenditure of resources, a computer program very similar to that used previously has been developed and utilized. In the program, it has been necessary only to modify the reliability and complexity equations. The sequence of steps taken in the process of maximizing reliability in the presence of constraints is shown in Figure 4.4.2 and may be used in a manner similar to previous discussion. The final results and system parameters are summarized in Table 4.4.1. Detailed data concerning system parameters at each step in the optimization process are presented in Appendix B.3 along with the computer flow diagram.

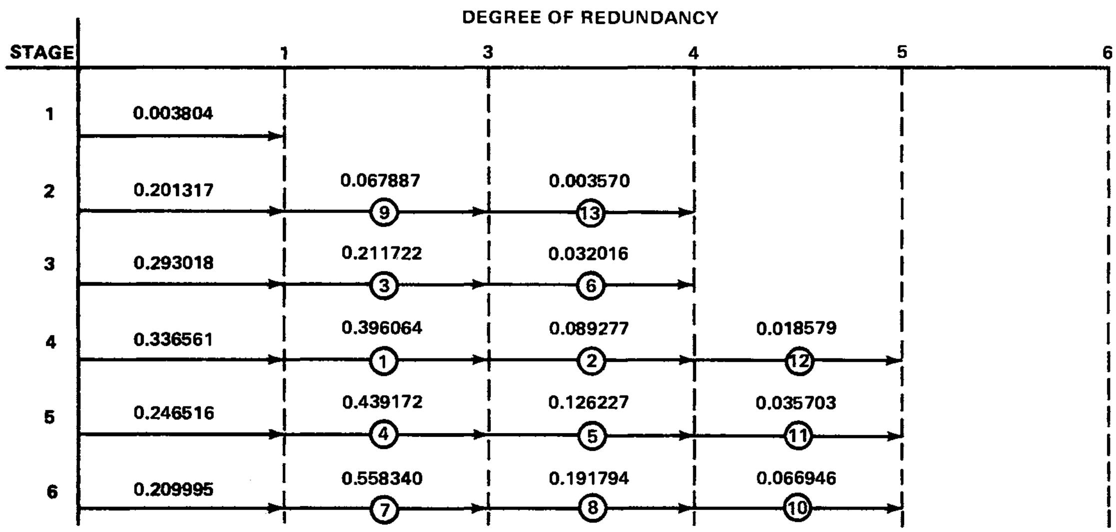


Figure 4.4.2. Optimization Sequence of System With Decision Elements for Criterion Function

$$\left(\frac{\Delta P}{\Delta C}\right)_{max} = \left[\frac{\Delta p_i}{\left(\Delta n_i\, c_i + \dfrac{3n_v}{N_T}\right) p_i}\right]_{max}$$

TABLE 4.4.1

SUMMARY OF RESULTS OF OPTIMIZING A SYSTEM WITH DECISION ELEMENTS UTILIZING THE CRITERION FUNCTION

$$\left(\frac{\Delta P}{\Delta C}\right)_{max} = \left[\frac{\Delta p_i}{\left(\Delta n_i c_i + \frac{3n_v}{N_T}\right) p_i}\right]_{max}$$

Stage	Number of Modules	Stage Reliability (Including Decision Element)
1	1	0.999430
2	4	0.998700
3	4	0.993497
4	5	0.995013
5	5	0.986699
6	5	0.970483
P (overall system reliability) = 0.944827		
C_r (total) = 4.712		

It is interesting to compare the results of Table 4.4.1 with those of Table 4.3.4 to determine the effect of the decision elements. It is noted that the decision elements reduced system reliability from 0.952892 to 0.944827. The overall complexity constraint of the system was not used in Table 4.3.4 since individual constraints were employed. However, since $a = b = c = 0.333$, it is calculated to be

$$C_r = \frac{0.333\ (99 + 55 + 78)}{0.333\ (21 + 12 + 17)} = \frac{232}{50} = 4.640 \quad .$$

The relative complexity of a system with decision elements is shown in Table 4.4.1 to be 4.712. This comparison is illegitimate in several respects; one being that if an overall constraint of 4.78 had been imposed on the system of Table 4.3.4, then more redundant modules would have been added and the reliability of that system would have been greater. It is clear, however, that incorporating the decision element into the model has adversely influenced system reliability.

Table 4.4.2 is the results obtained in an attempt to achieve a reliability goal of 0.9995. Since a maximum reliability of only 0.990275 was obtained, the goal was not achieved and the result was a dismal failure. Detailed results at each step are shown in Table B.3.2 of Appendix B. The last entry in this table indicates that according to the decision rule if another module had been added to any stage after this step, the system reliability would have been reduced, since all criteria values would become negative; i.e., Δp_i becomes negative. This is because the reliability of the decision elements decreases more than that gained in the stage by adding additional modules. Thus, a limiting value in system reliability has been found.

TABLE 4.4.2

SUMMARY OF RESULTS IN OBTAINING A RELIABILITY GOAL
IN A SYSTEM WITH DECISION ELEMENTS
FOR CRITERION FUNCTION

$$\left(\frac{\Delta P}{\Delta C}\right)_{max} = \left[\frac{\Delta p_i}{\left(\Delta n_i\, c_i + \frac{3n_{v_i}}{N_T}\right) p_i}\right]_{max}$$

Stage	Number of Modules	Stage Reliability (Including Decision Element)
1	1	0.999430
2	4	0.998700
3	5	0.998335
4	6	0.998077
5	7	0.997772
6	9	0.997921
P (overall system) = 0.990275		
C_r (total) = 6.656		

Although the results of this section vastly differ from those of Chapter III, where it was shown that infinite reliability is theoretically and physically possible under ideal conditions, it has been shown that from a practical point of view, system reliability can be substantially increased from 0.368 to 0.944827 within the constraints imposed. A maximum reliability of 0.990275 was achieved without regard to system complexity. In any case, however, it must be concluded that considerable differences exist in ideal and practical models.

CHAPTER V

SUMMARY AND CONCLUSIONS

5.1 SUMMARY

This investigation develops a generalized approach which can be used with parallel redundancy of three degrees or greater. Idealized models of parallel redundancy have been studied previously by several investigators with the assumption that only one-out-of-n parallel units must be operational for the redundant module or stage to be functional. But the problem of providing a decision element to detect and isolate failures and then without interruption to switch to a parallel module has not heretofore been considered. When this has been done, the idealized model changes drastically. Thus, a primary concern in this investigation has been to develop a generalized decision and switching element for a two-out-of-n parallel redundancy configuration, which is physically realizable and which can be used for an arbitrary number of inputs. System organization optimization from a reliability viewpoint utilizing this generalized element is then considered. Chapter II provides a technical introduction and the foundation upon which the other chapters are based. Although generalized equations for a two-out-of-n system are developed in Chapter II, beyond this, the chapter is treated as a special case of the two-out-of-n configuration with $n = 3$. The logic necessary for fault masking, failure detection and isolation, and module switching is developed. The problem of breaking a nonredundant system into modules and then making them redundant to maximize the overall redundant system reliability is considered. Also, the number of nonredundant elements which can be expected to fail before the redundant system fails is developed in Appendix A.

The generalized problem is then treated in detail in Chapter III. The logic necessary for the generalized decision element is developed. The complexity of this element is projected as a function of the number of inputs to it. It is also shown that optimum reliability results when redundant modules have the same reliability; i. e., when the level and degree of redundancy are the same in all the stages. With the decision element, which has been developed, and a system organized as recommended, a reliability as close to unity as desired can be obtained. However, system complexity is approximately the square of the number of parallel modules employed in each stage.

System reliability as close to unity as desired can only be obtained if the next system can accommodate or utilize the generated n-redundant system outputs. If the next system can utilize only one input, then the signals must be "necked" down through a single decision element. However, this single decision element is identical to those employed throughout the redundant system. In this case, the redundant system reliability is limited and can be no greater than the reliability of the decision element. System organization is then viewed from this standpoint. The necessity of a single decision element may not severely limit the gain in reliability which can be obtained with the proposed method because a nonredundant system may consist of hundreds of thousands of component parts, while only a hundred or so are required in the decision element.

A more practical system approach is considered in Chapter IV where the assumptions that a nonredundant system has been divided into modules of equal reliabilities and that identical degrees of redundancy are applied at each stage have been removed. This inherently introduces a new problem: interfacing n_i outputs from one stage with n_j inputs to the next stage. A method is proposed to solve this problem. A primary objective of Chapter IV is to find a solution to the problem; given system constraints, such as cost, power, weight, etc., how can the reliability of the system be maximized; or conversely, given a reliability goal, how can this goal be achieved to minimize these resources? Criteria functions are proposed and developed that lead to decision algorithms which can be used to solve these problems. Initially, the decision element is ignored in the illustrative example. However, after the decision algorithms have been thoroughly discussed and understood through examples, they are extended to include consideration of the decision element. The examples considered previously are then revisited.

5.2 CONCLUSIONS

The technique proposed herein for theoretically obtaining and physically realizing ultrareliability through redundancy depends to a very large extent upon the development of a decision element for fault masking, failure detection and isolation, and module switching. Although idealized models have been studied for some time where only one-out-of-n parallel modules was required to be functional, little practical value resulted from these models. In the idealized mathematical models, the decision element may have been included only as a mathematical symbol. However, the basic problem is that a generalized decision element satisfying the requirements of such a model has never been designed or physically realized. A two-out-of-n system is proposed for the simple reason that a generalized decision element can be realized. The basic approach to a two-out-of-n configuration is derived from the concept of majority logic, but the term "majority logic" is no longer descriptive for the generalized case.

A generalized decision element which can perform the functions of fault masking, failure detection and isolation, and module switching has been developed. Figure 3.4.2 shows the logic design of this element for six inputs, and the constructed breadboard (Fig. 3.4.4) accommodates 10 inputs. A particular advantage which has been realized through the logic development of the decision element is that it is possible to project its complexity for an arbitrary number of inputs, thus yielding reliability estimates as a function of the number of inputs.

With the two-out-of-n configuration and the assumption that a decision element is employed with each module, maximum reliability of the redundant system occurs when $\overline{R}_m \approx \frac{\overline{R}_v}{n-2}$, where $\overline{R}_m$ is the failure probability of a nonredundant module and $\overline{R}_v$ is the failure probability of the decision element; $\overline{R}_v$ is also a function of n. It has also been shown that for maximum reliability all modules should have the same reliability and that the same degree of redundancy should be applied to each stage. If the n outputs from the system can be utilized as inputs to the next system (i.e., the redundancy approach can be carried through to the next system), then a reliability as close to unity as desired can be obtained. However, such an accomplishment is not without penality. The relative complexity of the redundant system organized in an optimum manner compared to a nonredundant system is given by $n(n-1)$ or, for large n, by approximately the square of the degree of redundancy utilized.

If only one input can be accommodated in the next system, a single decision element must be employed at the last stage, and the reliability of the redundant system is limited and can never be greater than the single element. In practical applications, this is not a severe limitation because the decision element may consist only of a hundred or less component parts while the nonredundant system may contain thousands or even hundreds of thousands of component parts. The assumption that it is possible to subdivide a nonredundant system into m modules of equal reliabilities is not very practical, although from a theoretical standpoint, it is very valuable in establishing the possibility of the existence of an upper limit on reliability. However, it has been shown that there is no upper limit to the reliability which can be achieved with the proposed technique.

The division of a nonredundant system into segments of equal reliabilities is, quite likely, impractical. In this case, it necessarily follows that the degree of redundancy applied to these modules may be different. This introduces a new problem: designing a decision element which can accept n_i inputs and produce n_j outputs. A single decision element designed herein could be used for this function; however, the entire redundant system would fail when a decision element fails. To circumvent this problem, it is proposed that two-out-of-three majority logic be employed in the decision element. This does not create severe system limitations since in a practical application the complexity of the system will likely be much greater than that of the decision element. Furthermore, the interconnection problem can be readily solved with this approach.

With the above practical considerations, the question arises as to how to maximize system reliability within given constraints, such as cost, weight, power, etc., or conversely how to meet a reliability goal while expending a minimum amount of resources? To provide a solution to this problem, figures of merit or criteria functions are investigated. These lead to decision algorithms which are used in a recursive or iterative manner to arrive at a solution. Two criteria functions investigated in detail are $(\Delta P)_{max}$ and $\left(\frac{\Delta P}{\Delta C}\right)_{max}$. In the first case, redundant elements are added to the various stages in a manner to yield greatest gain in system reliability. In the second case, modules are added to maximize the reliability and complexity (cost, weight, or power, etc.) gradient. It follows that if $\frac{\Delta P}{\Delta C}$ is maximized at each step in the process, then the final system will also possess a maximum $\frac{\Delta P}{\Delta C}$. Detailed examples have been considered both with and without the decision element. An illustrative example has been considered in detail to show generally how the optimization process is accomplished with the type of configuration proposed herein.

5.3 RECOMMENDATIONS FOR FURTHER STUDY

One can draw an analogy between an investigation of this nature with its many related facets and a long hall with many branching corridors. One could easily take any one of these; however, if this happens, then one does not accomplish his goal. There are, therefore, several related areas requiring research which for several reasons could not be undertaken here. The investigation which has been performed required specialization in each of three areas: (1) digital system logic design, (2) probability theory, and (3) system optimization techniques which is often considered to be in the field of operations research. Therefore, any of the tasks mentioned will fall in one or more of these areas. The sequence in which the items for further study are listed in no way indicates the order of importance or preference.

1. A closed-form solution should be developed for

$$\int_0^\infty \left[1 - \sum_{i=n-1}^{n} \binom{n}{i} R^{n-i} (1-R)^i \right]^m dt$$

where

$$R = e^{-\frac{kt}{m}} \quad .$$

The above integral represents the mean time to failure of a redundant system with m identical modules, each with a redundancy of degree n . This integral was solved numerically in Appendix A. Although a closed-form solution for the integral is possible as shown in the appendix, a great deal of difficulty was encountered in evaluating the integral for large m . It appears that the solution of this integral might be expressed in terms of the Bessel function of the first kind or possibly in terms of Legendre polynomials. A neat, closed-form solution which can be readily evaluated is desired.

2. Logical design and development of a decision element which will accept n_i inputs and yield n_j outputs. The case of $i = j$ has been covered in this investigation and a majority logic approach has been proposed which circumvents the problem. However, the basic requirement suggested is to design a single (nonredundant) decision element for $i \neq j$ where both $i < j$ and $i > j$ are possible.

3. The optimization technique proposed and the results obtained herein should be studied and compared with those obtained from dynamic programming. Are there clear advantages to either method?

4. The redundancy technique proposed herein is primarily for digital systems. How can this technique be adapted for use in analog systems, and is it practical?

5. It was suggested that the majority logic technique along with the proposed decision element could be used in a system where manual repair and replacement were possible, not necessarily just to increase system reliability but primarily to reduce system downtime, trouble-shooting time, repair time, etc. What are the tradeoffs in redundant system costs versus the saving obtained by a reduction of these items?

6. In some applications, a single decision element may be desirable between the redundant stages rather than utilizing a decision element with each module. It is certainly less expensive. If this were done, since the failure probability of the decision element increases

exponentially with the degree of redundancy employed, one would expect a limiting system reliability. This aspect should be investigated in much the same manner as that undertaken herein.

7. With the advent of large-scale integrated circuits, redundancy techniques possibly could be used to overcome some of the production yield problems. Thus, many circuits could be made redundant on one chip which could therefore tolerate several failures before having to discard the chip. With a nonredundant chip, a single failure results in a loss of the entire chip. Is there reason to consider this approach from an economic standpoint?

8. Adequate consideration has not been given to majority logic of higher degree than three. What is required, for example, in a decision element for a five-out-of-nine configuration and how does this affect the overall system organization? Is such a configuration feasible when the logic for the decision has been considered?

9. Multiprocessing is of current interest in the computer field. One of the primary problems in multiprocessing systems is to inhibit a malfunction in a particular processor from destroying the operation of a complete system consisting of several individual processors. Can the techniques of failure detection and isolation developed herein be advantageously employed in a multiprocessing system?

10. In the field of operations research, there is a dire need for determining methods and procedures for establishing overall system reliability goals. What usually happens is that, due to lack of direction, when a reliability goal is to be established, top management simply pulls a number out of the air. However, a reliability goal should be considered as insurance, for it is a way of expressing the chances of something being successful on a particular trial. Are expected losses being minimized, or are human lives being protected? This, of course, depends on the system under consideration, but in any event, a method needs to be developed to express areas of primary concern (i.e., objectives) in terms of a reliability goal rather than to choose a number which is palatable and pushes the state-of-the-art, etc. An actual example of a so-called reliability specification is given by the following example: One of this country's future space systems is to be designed such that "two failures could be tolerated without loss of mission and the third failure should not result in loss of the vehicle." The exact meaning of this ground rule is left to the imagination of the reader; but in all fairness, it should be pointed out that many thousands of engineering manhours have been expended on its interpretation. Perhaps, this example exemplifies the point and stresses the necessity for mathematical analysis of management problems, particularly in the area of reliability.

The preceding items which are recommended vary greatly from a well-defined problem (item 1) to the investigation of a completely new discipline (item 10). However, this is as it should be, and it emphasizes the great need and the latitude that one has in this relatively new and fertile area of research.

A SELECTED BIBLIOGRAPHY

1. Balaban, Harold S. "A Selected Bibliography on Reliability," IRE Transactions on Reliability and Quality Control, Vol. RQC-11, July 1962, pp. 86-103.

2. Barlow, R. E., and L. C. Hunter. "Criteria for Determining Optimum Redundancy," IRE Transactions on Reliability and Control, Vol. 9, April 1960, pp. 73-77.

3. Bellman, R. "Dynamic Programming," Princeton University Press, Princeton, New Jersey, 1957.

4. Bellman, R., and Dreyfus. "Dynamic Programming and the Reliability of Multicomponent Devices," Operations Research, Vol. 6, 1958, pp. 200-206.

5. Brown, W. G., J. Tierney, and R. Wasserman. "Improvement of Electronic Computer Reliability Through the Use of Redundancy," IRE Transactions on Electronic Computer, Vol. EC-10, No. 3, September 1961, pp. 407-216.

6. Everett, Hugh III. "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources," Operations Research, Vol. 11, No. 3, May 1963, pp. 399-417.

7. Gordon, Robert. "Optimum Component Redundancy for Maximum Systems Reliability," Operations Research, Vol. 5, April 1957, pp. 229-244.

8. Gurzi, Karyl J. "Estimates for Best Placement of Voters in Triplicated Logic Network," IEEE Transactions on Electronic Computer, Vol. EC-14, No. 15.

9. Herron, D. P. "Optimizing Tradeoffs of Reliability Versus Weight," IEEE Transactions on Reliability, Vol. R-12, December 1963, pp. 50-54.

A SELECTED BIBLIOGRAPHY (Continued)

10. Kettelle, John D., Jr. "Least-Cost Allocations of Reliability Investment," Operations Research, Vol. 10, No. 2, March 1962.

11. Lyons, R. E., and Vanderkulk. "The Use of Triple-Modular Redundancy to Improve Computer Reliability," IBM Journal of Research and Development, Vol. 6, No. 2, April 1962, pp. 200-209.

12. Moscowitz, F. "The Analysis of Redundant Networks," AIEE Transactions on Communication and Electricity, No. 39, November 1958, pp. 627, 632.

13. Nathan, Irwin. "Cost Effectiveness — What It Is and Is Not," Sixth Annual National Conference on Electronic Reliability, New York, New York, May 21, 1965.

14. Rozenberg, D. P., and H. L. Ergott. "Modular Redundancy for Spaceborne Computer," Proceedings of the Spaceborne Computer Engineering Conference, Anaheim, Calif., October 30-31, 1962, pp. 137-149.

15. Sasaki, Masafumi. "A Simplified Method of Obtaining Highest System Reliability," Eighth National Symposium on Reliability and Quality Control, January 1962, pp. 489-502.

16. Sasaki, Masafumi. "An Easy Allotment Method Achieving Maximum System Reliability," Ninth National Symposium on Reliability and Quality Control, January 1963, pp. 109-124.

17. Shooman, Martin T. "Probabilistic Reliability: An Engineering Approach," McGraw-Hill, Inc., 1968.

18. Short, Robert A. "The Attainment of Reliable Digital Systems Through the Use of Redundancy — A Survey," Computer Group News, March 1968.

19. Teoste, Rein. "Design of a Repairable Redundant Computer," IRE Transactions on Electronic Computer, Vol. EC-11, No. 5, October 1962, pp. 643-649.

A SELECTED BIBLIOGRAPHY (Concluded)

20. von Neumann, J. "Probabilistic Logics and the Synthesis of Reliable Organs from Unreliable Components," Annals of Mathematical Studies, Vol. 34, Princeton University Press, 1956, pp. 43-98.

21. Webster, L. R. "Choosing Optimum Configurations," Proceedings Tenth Annual Symposium on Reliability and Quality Control, 1964, pp. 345-359.

APPENDIX A

NUMBER OF FAILURES MOST LIKELY TO OCCUR IN A TWO-OUT-OF-n REDUNDANT SYSTEM BEFORE SYSTEM FAILURE

An interesting problem not only from an academic standpoint but also one which arises in the application of replaceable modules, is the question of how many module and voter failures can one expect to have occurred before the redundant system fails. The answer to this question will give some insight into how often maintenance is required in a redundant system. To treat this problem, a slight digression is necessary to define and develop expressions for mean time between failures (mtbf).

The mtbf is an interesting and sometimes useful parameter in reliability theory.[8] Two specific requirements for any probability or reliability distribution function are:

[8]Although the mtbf is a very useful parameter, its value in determining the reliability of a system should not be overemphasized. Redundant systems whose reliability may be very good over some predetermined time period may fall off very rapidly and would not have an extremely large mtbf. The mtbf is of considerable value; however, when estimating equipment downtime, the number of failures most likely to occur in any time frame, etc.

1. The distribution function $\overline{R}(t)$ must satisfy

$$\overline{R}(t) \rightarrow 0 \quad \text{as} \quad t \rightarrow 0$$

$$\overline{R}(t) \rightarrow i \quad \text{as} \quad t \rightarrow \infty \quad ,$$

Here, $\overline{R}(t)$ denotes the probability of failure which is a function of time.

2. When the frequency function is integrated with respect to time between the limits of $-\infty$ to $+\infty$, a unit value must be obtained. The frequency function is obtained by differentiating $\overline{R}(t)$ with respect to time. Thus,

$$\int_{-\infty}^{\infty} \frac{d\,\overline{R}(t)}{dt}\, dt = 1$$

or, since time cannot be negative,

$$\int_{0}^{\infty} \frac{d\,\overline{R}(t)}{dt}\, dt = 1 \quad .$$

The mean or expected value is by definition the first central moment and is given by

$$\tau = \int_{-\infty}^{\infty} t\, f(t)\, dt$$

where $f(t)$ is the frequency function. Thus, the above equation can be written as

$$\tau = \int_{0}^{\infty} \frac{t\, d\,\overline{R}(t)}{dt}\, dt \quad . \qquad (A.1)$$

The mtbf can be written as a function of R, which often results in a more compact and useful form. Let

$$u = t$$

$$dv = \frac{d\overline{R}}{dt}\, dt$$

then

$$du = dt$$

$$v = \overline{R}$$

and Equation (A.1) takes the form

$$\tau = \int_0^\infty u\,dv = uv\Big]_{v=0}^{v=\infty} - \int_0^\infty v\,du \qquad \text{(A.2)}$$

or

$$\begin{aligned} \tau &= t\overline{R}\Big]_{t=0}^{t=\infty} - \int_0^\infty \overline{R}\,dt \\ &= t(1-R)\Big]_{t=0}^{t=\infty} - \int_0^\infty (1-R)\,dt \\ &= (t - tR)\Big]_{t=0}^{t=\infty} - \int_0^\infty dt + \int_0^\infty R\,dt \\ &= tR\Big]_{t=0}^{t=\infty} + \int_0^\infty R\,dt \quad . \end{aligned} \qquad \text{(A.3)}$$

Because

$$R \rightarrow 0 \quad \text{as} \quad t \rightarrow \infty$$

and

$$R \rightarrow 1 \quad \text{as} \quad t \rightarrow 0$$

the first term vanishes leaving

$$\tau = \int_0^\infty R\,dt \qquad . \tag{A.4}$$

For a nonredundant system, $R = e^{-N_T \lambda t}$, where N_T is the number of components (diodes, resistors, capacitors, etc.) in the system, λ is the average component failure rate, and t is the operating time. Equation (A.4) may be readily evaluated to yield

$$\tau = \int_0^\infty e^{-N_T \lambda t}\,dt = \frac{1}{\lambda N_T} \tag{A.5}$$

where λN_T is defined as the nonredundant system failure rate.

The reliability of one triplicated redundant stage (m=1) is given by

$$P = 3R^2 - 2R^3 \tag{A.6}$$

where $R = R_m R_v = e^{-\lambda t(n_v + n_m)}$. For optimization' $n_v \approx n_m$, and because replication occurs at the systems level $n_m + n_m \approx 2N_T$, Equation (A.6) can be written as

$$P = 3e^{-4N_T\lambda t} - 2e^{-6N_T\lambda t} \quad . \tag{A.7}$$

Integrating Equation (A.7) between the limits of 0 and ∞ yields an mtbf of

$$\tau_{m=1} = \frac{5}{12\,\lambda N_T} \quad . \tag{A.8}$$

Thus, it is seen that the mtbf of a redundant system triplicated at the system level is approximately only one-half that of a nonredundant system. However, the reliability of the redundant system is higher for $\overline{R} \leq 1/2$ or for $t \leq 0.693\,\tau_s$; i.e., when the operating time is less than 0.693 times the mtbf of a simplex system. The system just considered is of no practical interest since the nonredundant system consisted of only approximately 20 component parts because optimization was assumed; i.e., $n_v = n_m$ and the decision element can be designed with 20 parts. However, the point that mtbf and reliability have different meanings in redundant systems is well demonstrated.

The remainder of this appendix consists of determining the mtbf of a redundant system with n degrees of redundancy applied at each stage, and for a nonredundant system which has been divided into m modules of equal reliability. When this parameter has been determined, the number of failures which may be expected to have occurred within the redundant system at the mtbf can then be estimated. The relative complexity of decision element to module will be treated as a system variable, but the system will be numerically evaluated for optimum design.

The reliability of a redundant system containing redundancy of n degree and consisting of m identical modules is

$$P = \{1 - (1-R)^{n-1} [n - (n-1)\bar{R}]\}^{m} \qquad (A.9)$$

where R is the reliability of each module and decision element and is given by the expression

$$R = R_m R_v = e^{-n_m \lambda t} e^{-n_v \lambda t} = e^{-\lambda t (n_m + n_v)} \qquad (A.10)$$

where λ is the average component failure rate, t is the operating time, and n_m, n_v are the number of parts in the module and decision element, respectively. Letting $a = \frac{n_v}{n_m}$ and $n_m = \frac{N_T}{m}$, Equation (A.10) becomes

$$R = e^{-\frac{(1+a)N_T \lambda t}{m}} \qquad (A.11)$$

and Equation (A.9) can be written as

$$P = \left\{ 1 - \left(1 - e^{-\frac{(1+a)N_T\lambda t}{m}} \right)^{n-1} \left[n - (n-1) \left(1 - e^{-\frac{(1+a)N_T\lambda t}{m}} \right) \right] \right\}^{m} . \tag{A.12}$$

This function must be respected, because considerable difficulty arises when an attempt is made to integrate it in closed form with respect to time between the limits of 0 and ∞. Two specific cases, $n = 3$ and $n = 4$, will be treated before an attempt is made to obtain a generalized solution to this problem.

For $n = 3$, Equation (A.12) may be written as

$$P = \left[3e^{-\frac{2(1+a)N_T\lambda t}{m}} - 2e^{-\frac{3(1+a)N_T\lambda t}{m}} \right]^{m} . \tag{A.13}$$

Expansion of Equation (A.13) yields

$$\begin{aligned} P = e^{-2(1+a)N_T\lambda t} \Bigg\{ & 3^m - m\, 3^{m-1} \left(2e^{-\frac{(1+a)N_T\lambda t}{m}} \right) \\ & + m(m-1)\, 3^{m-2} \left(2e^{-\frac{(1+a)N_T\lambda t}{m}} \right)^2 + \ldots\ldots\ldots \\ & + 2^{n-1}\, m(m-1)\,(m-2) \ldots\ldots (m-n+2)\, 3^{m-n+1}\, e^{-\frac{(n-1)\,(1+a)N_T\lambda t}{m}} \Bigg\} \end{aligned} \tag{A.14}$$

This series can be integrated term by term since it converges absolutely for any value of m . When this is done, the result is

$$\tau_R = \frac{1}{(1+a)\lambda N_T} \left\{ \frac{3^m}{2} - \frac{2m^2\, 3^{m-1}}{2m+1} + \frac{2^2\, m^2 (m-1)\, 3^{m-2}}{2\,!\, (2m+2)} + \cdots\cdots + \frac{(-1)^{n-1}\, 2^{n-1}\, mm\,!\, 3^{m-n+1}}{(n-1)\,!\, (m-n+1)\,!\, [2m+(n-1)]} \right\} \tag{A.15}$$

or in general

$$\tau_R = \frac{1}{(1+a)\lambda N_T} \sum_{k=0}^{m} \frac{(-1)^k\, 2^k\, mm\,!\, 3^{m-k}}{k\,!\, (m-k)\,!\, (2m+k)} \quad . \tag{A.16}$$

The summation in Equation (A.16) in many respects resembles a Bessel function of the first kind; however, an unfruitful effort resulted from an attempt to define the equation in these terms.

For n = 4 , the reliability is given by the series expansion

$$P = e^{-2(1+a)N_T\lambda t} \left\{ 6^m - m6^{m-1} \left(8e^{-\frac{(1+a)N_T\lambda t}{m}} - 3e^{-\frac{2(1+a)N_T\lambda t}{m}} \right) + m(m-1)\, 6^{m-2} \left(8e^{-\frac{(1+a)N_T\lambda t}{m}} - 3e^{-\frac{2(1+a)N_T\lambda t}{m}} \right)^2 - m(m-1)\,(m-2)\, 6^{m-3} \left(8e^{-\frac{(1+a)N_T\lambda t}{m}} - 3e^{-\frac{2(1+a)N_T\lambda t}{m}} \right)^3 + \cdots\cdots \right\} \tag{A.17}$$

Equation (A.17) can be written in the form

$$P = e^{-2kt}\left[a - be^{-\frac{kt}{m}} + ce^{-\frac{2kt}{m}} - de^{-\frac{3kt}{m}} + \dots\dots\dots\right] \qquad \text{(A.18)}$$

where

$$k = (1+a)N_T\lambda t$$

$$a = 6^m$$

$$b = 8m\ 6^{m-1}$$

$$c = 3m\ 6^{m-1} + 32\ m(m-1)\ 6^{m-2}$$

$$d = 24(m)\ (m-1)\ 6^{m-2} - \frac{256}{3}\ m(m-1)\ (m-2)\ 6^{m-3}$$

or Equation (A.18) can be written as

$$P = ae^{-\frac{2kt}{m}} - be^{-kt\left(\frac{2m+1}{m}\right)} + ce^{-kt\left(\frac{2m+2}{m}\right)} - de^{-kt\left(\frac{2m+3}{m}\right)} + \dots . \qquad \text{(A.19)}$$

Since Equation (A.19) has a finite number of terms $(m+1)$ for any $m < \infty$, it converges absolutely and can be integrated term by term between the limits of 0 and ∞ yielding

$$\tau_R = \frac{1}{k}\left[\frac{a}{2} - \frac{m}{2(2m+1)}\ b + \frac{m}{2(2m+2)}\ c - \frac{m}{2(m+3)}\ d + \dots\right] . \qquad \text{(A.20)}$$

Thus, the mtbf of a redundant system can be expressed as a function of $k = (1+a)\lambda N_T$. This is important because numerical methods can be used to evaluate the integral given in Equation (A.12) by assuming some value of k. For adequate precision, numerical integration is necessary rather than an evaluation of the series given in Equation (A.16). For example, utilizing a Univac 1108 digital computer and double precision arithmetic, m is restricted to be less than 20 when the series evaluation approach is taken. The functional form of Equations (A.16) and (A.20) is given by

$$\tau_R = \frac{1}{k} f(m,n) \qquad \text{or} \qquad f(m,n) = k\tau_R \quad . \tag{A.21}$$

The number of terms in the series expansion of f(m,n) is determined by n while m establishes the value of the function for any given n.

The integral of Equation (A.12) between the limits of 0 and ∞ can be found utilizing one of several numerical techniques. Simpson's rule has been used herein, and the mtbf is given approximately by

$$\tau_R = \frac{\Delta t}{3} [P(t=0) + 4\,P(t=1) + 2\,P(t=2) + 4\,P(t=3) + \ldots\ldots. \tag{A.22}$$
$$+ 2\,P(t=998) + 4\,P(t=999) + P(t=1000)] \quad .$$

Equation (A.12) is therefore evaluated at $t = 0, 1, \ldots, 1000$, and Equation (A.22) is evaluated at these points. The term k/m was chosen such that $.002 \leq k/m \leq 0.003$. This choice assures that the integral converges and also gives reasonable accuracy.

The error of the approximation is given by

$$AM \leq \epsilon \leq AM'$$

where

$$A = \frac{\Delta t^4 \, (b-a)}{180} \quad .$$

M' and M are the largest and smallest values, respectively, of the fourth derivative of the function P within the limits of integration a and b. M' and M can be found by the Gregory-Newton interpolation formula given by

$$
\begin{aligned}
p_j &= \sum_{i=0}^{n} \binom{j}{i} \Delta^i p_o = p_o + \binom{j}{1} \Delta p_o + \binom{j}{2} \Delta^2 p_o + \binom{j}{3} \Delta^3 p_o + \binom{j}{4} \Delta^4 p_o + \ldots \\
p_j'(x) &= \frac{1}{\Delta t}\left[\Delta p_o + \left(j - \frac{1}{2}\right) \Delta^2 p_o + \left(\frac{3j^2 - 6j + 2}{6}\right) \Delta^3 p_o + \left(\frac{2j^3 - 9j^2 + 11j - 3}{12}\right) \Delta^4 p_o + \ldots\right] \\
p_j^{(2)}(x) &= \frac{1}{\Delta t^2}\left[\Delta^2 p_o + (j-1)\,\Delta^3 p_o + \left(\frac{6j^2 - 18j + 11}{12}\right) \Delta^4 p_o + \ldots\right] \\
p_j^{(3)}(x) &= \frac{1}{\Delta t^3}\left[\Delta^3 p_o + \left(\frac{2j-3}{3}\right) \Delta^4 p_o + \ldots\right] \\
p_j^{(4)}(x) &= \frac{1}{\Delta t^4}\left[\Delta^4 p_o + \ldots\right]
\end{aligned}
\tag{A.23}
$$

The actual error depends on the specific values of n and m used; however, the maximum error over the range of values considered here is approximately $-0.0011 \leq e_{max} \leq +0.0017$.

Before discussion of the results of the evaluation of Equation (A.22), it is desirable to proceed with the derivation of the number of failures expected in the redundant system before system failure. The total number of equivalent modules in the redundant system is given by

$$C = nm(1+a) \tag{A.24}$$

where n and m are the degree and level of redundancy employed and a is as previously defined. The reliability of one module is given by

$$R_m = e^{-n_m \lambda t} \quad . \tag{A.25}$$

Letting

$$n_m = \frac{N_T}{m}$$

and

$$t = \tau_R = \frac{f(m,n)}{k} = \frac{f(m,n)}{(1+a)\lambda N_T} \quad ,$$

Equation (A.25) can be written as

$$R_m = e^{-\frac{f(m,n)}{m(1+a)}} \quad . \tag{A.26}$$

Thus, Equation (A.26) gives the reliability of a single module at the mtbf of the redundant system. Equation (A.24) gives the equivalent number of modules employed in the redundant system.

The number of failures which can be expected to have occurred by time τ_R is simply the mean or expected value of the binomial distribution and is given by

$$\begin{aligned} \mu_f &= C(1 - R_m) \\ &= nm(1+a)\left[1 - e^{-\frac{f(m,n)}{m(1+a)}}\right] \quad . \end{aligned} \tag{A.27}$$

The standard deviation in the number of expected failures with a binomial distribution is

$$\sigma_f = \sqrt{nm(1+a)\left(1 - e^{-\frac{f(m,n)}{m(1+a)}}\right) e^{-\frac{f(m,n)}{m(1+a)}}} \quad . \tag{A.28}$$

Notice that since

$$f(m,n) = k\,\tau_R = (1+a)\lambda N_T\,\tau_R \tag{A.29}$$

and because the mtbf in the nonredundant is given by

$$\tau_S = \frac{1}{\lambda N_T} \quad ,$$

Equation (A.29) can be expressed as

$$f(m,n) = (1+a)\frac{\tau_R}{\tau_S} \quad . \tag{A.30}$$

Equation (A.30) gives more of an intuitive feeling for the function $f(m,n)$ than the previous equations.

It is shown in Chapter III that for optimum design

$$a \approx n - 2$$

and Equation (A.27) can be written as

$$\mu_f \approx n(n-1)m \left[1 - e^{-\frac{f(m,n)}{m(n-1)}}\right] \tag{A.31}$$

or by substitution of Equation (A.30) as

$$\mu_f = n(n-1)m \left[1 - e^{-\frac{\tau_R}{m\tau_S}}\right] \quad .$$

When the product $m(n-1)$ is large, the standard deviation of the estimate is given by approximately $\sqrt{\mu_f}$.

From the numerical evaluation of $f(m,n)$, the ratio of $\frac{\tau_R}{\tau_S}$ can be found from Equation (A.30). Figure A.1 shows this ratio plotted as a function of n for several values of m when optimum system design is assumed; i.e., when $a = \frac{n_v}{n_m} \approx n-2$. It is noted that for

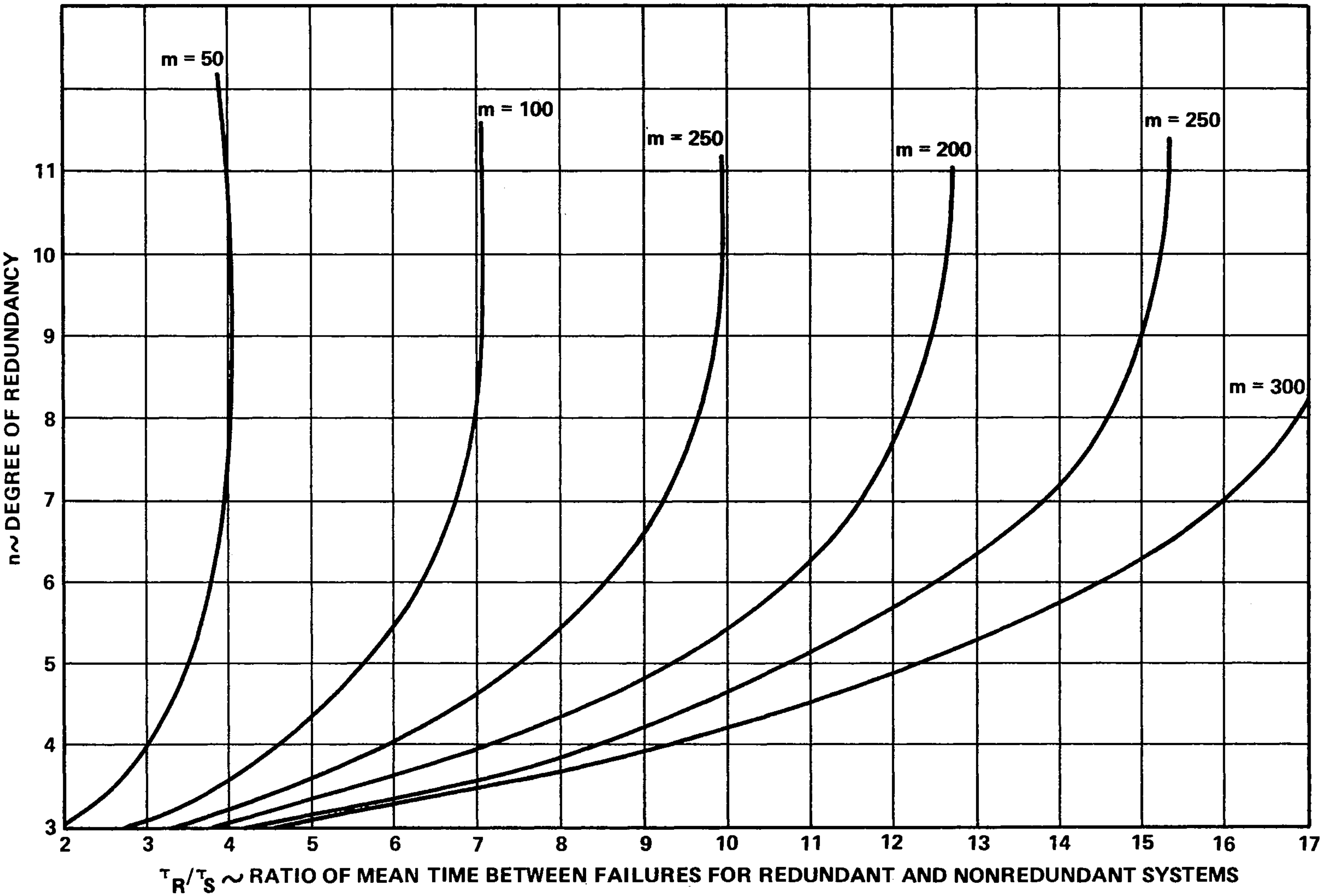


Figure A.1. n Versus Ratio τ_R/τ_S Where $a = n_v/n_m \approx n-2$

a given m, a maximum $\frac{\tau_R}{\tau_S}$ results from a specific n; e.g., for m = 50, a maximum $\frac{\tau_R}{\tau_S} = 4.05$ is noted at approximately n = 9. Increasing n further causes $\frac{\tau_R}{\tau_S}$ to decrease slowly. Although it is not shown in the figure, the ratio $\frac{\tau_R}{\tau_S}$ is less than one for any $m \leq 7$ and $n \leq 11$.

Figure A.2 shows the number of failures expected in the redundant system, μ_f, plotted as a function of m for several values of n. Again, it has been assumed that the system has been optimally designed or that $a = n-2$. Redundant system complexity is given by Equation (A.24); e.g., in a system which has been optimally designed with m = 100 and n = 5, approximately 110 failures could be expected to have occurred before the redundant system fails. However, the nonredundant system contains 100 modules and the redundant system contains

$$C = nm(1+a) = n(n-1)m = (5)(4)(100) = 2,000$$

equivalent modules. Thus, the redundant system can only tolerate approximately 5.5-percent failure in the total system before failure.

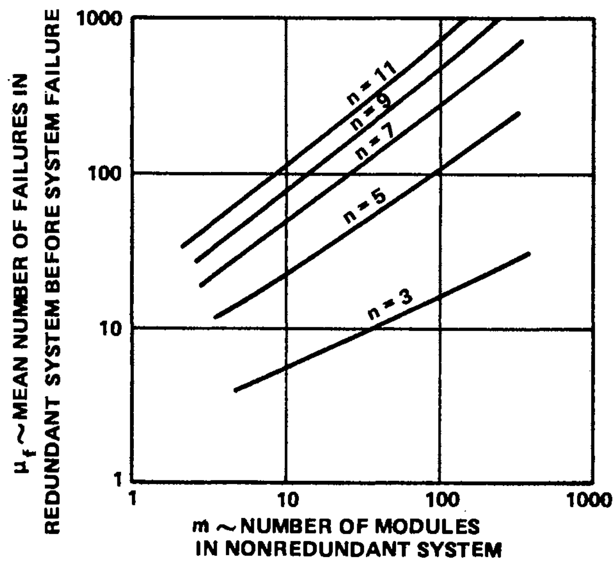


Figure A.2. μ_f Versus m Where $a = n_v/n_m \approx n-2$

The factor $\tau_R k$ has been plotted as a function of m for several values of n in Figure A.3. Nothing new can be obtained from this figure and it has been presented only as a means of quickly determining τ_R for any value of a, λ, and N_T; e.g., for $m = 100$ and $n = 5$, a value of 22.5 is shown in Figure A.3 with

$$a = 1$$

$$\lambda = 10^{-8} \text{ failures/hr}$$

$$N_T = 25 \times 10^3$$

$$\tau_R = \frac{22.5}{(1+a)\lambda N_T} = \frac{22.5}{2(10^{-8})(25 \times 10^3)} = 45 \times 10^3 \text{ hrs/failure.}$$

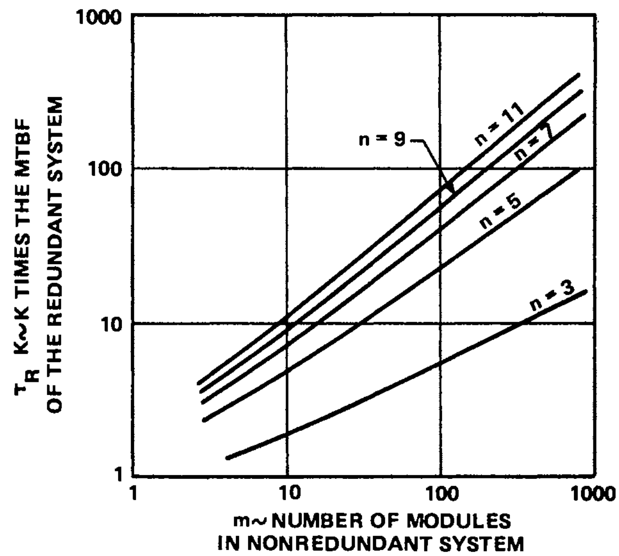


Figure A.3. $\tau_R k$ Versus m Where $k = (1+a)\lambda N_T$

APPENDIX B

COMPUTER PROGRAM FLOW DIAGRAMS AND DETAILED RESULTS OF THE OPTIMIZATION PROCESSES

This appendix presents detailed information concerning the computer programs and mathematical computations which were utilized in the optimization processes considered in Chapter IV. The appendix is divided into three sections: the first treats the optimization process employing the criterion function $(\Delta P)_{max} = \left(\frac{\Delta p_i}{p_i}\right)_{max}$;

the second the function $\left(\frac{\Delta P}{\Delta C}\right)_{max} = \left(\frac{\Delta p_i}{\Delta n_i \, c_i \, p_i}\right)_{max}$;

and the third modifies the second function to include the decision element and is given explicitly by the function

$$\left(\frac{\Delta P}{\Delta C}\right)_{max} = \left[\frac{\Delta p_i}{\left(\Delta n_i \, c_i + \frac{3\Delta n_v}{N_T}\right) p_i}\right]_{max} .$$

In each section, the dual problems of organizing a system to optimize reliability within given constraints and of achieving a reliability goal with minimum resources are treated.

B.1 COMPUTATIONS FOR CRITERION FUNCTION

$$(\Delta P)_{max} = \left(\frac{\Delta p_i}{p_i}\right)_{max}$$

Figure B.1.1 illustrates the logical flow diagram of the computer program utilized in the optimization process with the criterion function $(\Delta P)_{max} = \left(\frac{\Delta p_i}{p_i}\right)_{max}$. The flow diagram is straightforward and requires little explanation except possibly definition of some of the terms used. The system inputs are defined as follows:

N Number of stages or modules into which a nonredundant system has been divided.

$\left.\begin{matrix} u \\ v \\ w \end{matrix}\right\}$ Parameters of each module, taken here to be cost, weight, and power. Thus, in the example used, there are six each of these.

$\bar{R}_i$ Failure probability of the i^{th} module.

$\left.\begin{matrix} u_c \\ v_c \\ w_c \end{matrix}\right\}$ Constraints in redundant system cost, weight, and power, respectively.

I_c This is a bit which determines the dual problem to be solved; i.e., maximize system reliability within given constraints or achieve a reliability goal with minimum resources.

P_L Reliability goal to be achieved when I_c is set to a logical "1."

$\left.\begin{array}{l} a \\ b \\ c \end{array}\right\}$ Weighting factors which can be applied to cost, weight, and power, respectively.

Detailed computer printouts showing the results of the dual problem at each step in the process are shown in Tables B.1.1 and B.1.2.

B.2 COMPUTATIONS FOR CRITERION FUNCTION

$$\left(\frac{\Delta P}{\Delta C}\right)_{max} = \left(\frac{\Delta p_i}{\Delta n_i \; c_i \; p_i}\right)_{max}$$

The logical flow diagram of the computer program is shown in Figure B.2.1 and is very similar to that used in the previous section. The major difference is the specific calculations which are made at each step.

Detailed results of each step are given in Tables B.2.1 and B.2.2.

B.3 COMPUTATIONS FOR CRITERION FUNCTION

$$\left(\frac{\Delta P}{\Delta C}\right)_{max} = \left[\frac{\Delta p_i}{\left(\Delta n_i\, c_i + \frac{3\Delta n_{v_i}}{N_T}\right) p_i}\right]_{max}$$

The logical program for system optimization when consideration is given to the decision element is shown in Figure B.3.1. Again, the logical developments are similar to those used previously, the primary difference being in specific calculations used.

Tables B.3.1 and B.3.2 give detailed results obtained at each step in the optimization process. Table B.3.1 is applicable to optimizing system reliability in the presence of constraints while Table B.3.2 presents the results obtained in achieving a reliability goal with minimum resources.

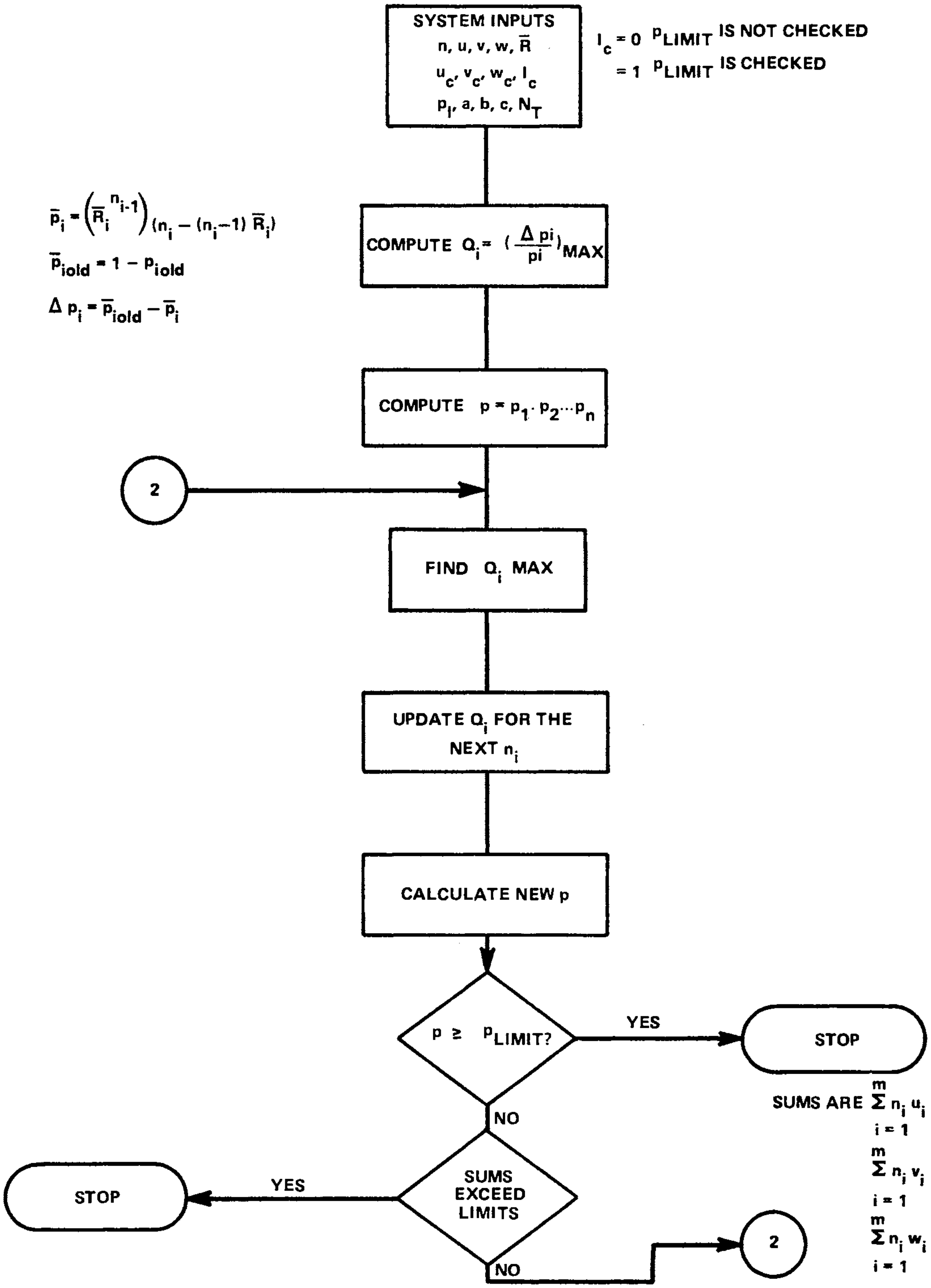


Figure B.1.1. Logic Diagram of Computer Program for Optimization Process Utilizing

$$(\Delta p)_{max} = \left(\frac{\Delta p_i}{p_i}\right)_{max}$$

TABLE B.1.1

SYSTEM OPTIMIZATION USING $(\Delta P)_{max} = \left(\frac{\Delta p_i}{p_i}\right)_{max}$

AND CONSTRAINTS $C_u = 99$, $C_v = 57$, AND $C_w = 83$

```
                    INPUT DATA

N =       6
ICØN =      0
PLIMIT VALUE =          ,99950000
A =            ,33333000   B =            ,33333000   C =            ,33333000
U( 1) =      1,00000000
U( 2) =      2,00000000
U( 3) =      3,00000000
U( 4) =      4,00000000
U( 5) =      5,00000000
U( 6) =      6,00000000
V( 1) =      1,00000000
V( 2) =      1,00000000
V( 3) =      2,00000000
V( 4) =      2,00000000
V( 5) =      3,00000000
V( 6) =      3,00000000
W( 1) =      1,00000000
W( 2) =      3,00000000
W( 3) =      2,00000000
W( 4) =      2,00000000
W( 5) =      4,00000000
W( 6) =      5,00000000
RBAR( 1) =        ,00057000
RBAR( 2) =        ,05936000
RBAR( 3) =        ,11815000
RBAR( 4) =        ,17694000
RBAR( 5) =        ,23573000
RBAR( 6) =        ,29452000
CU =     99,00000000     CV =     57,00000000     CW =     83,00000000


NI =      1
SUM NI*UNI =   ,21000000E 02
SUM NI*VNI =   ,12000000E 02
SUM NI*WNI =   ,17000000E 02

INITIAL P VALUE =    ,36790319E 00
```

```
  NEW NI =        3
  P =     ,41373126E 00
  SUM NI*UNI =   ,31000000E 02
  SUM NI*VNI =   ,18000000E 02
  SUM NI*WNI =   ,25000000E 02
  PI( 1) =     ,99943000E 00
  PI( 2) =     ,94064000E 00
  PI( 3) =     ,88185000E 00
  PI( 4) =     ,82306000E 00
  PI( 5) =     ,85949249E 00
  PI( 6) =     ,70548000E 00
1 DELTA PI( 1)/PI( 1) =      ,56935020E-03
1 DELTA PI( 2)/PI( 2) =      ,52312781E-01
1 DELTA PI( 3)/PI( 3) =      ,90231155E-01
1 DELTA PI( 4)/PI( 4) =      ,11432447E 00
3 DELTA PI( 5)/PI( 5) =      ,11329285E 00
1 DELTA PI( 6)/PI( 6) =      ,12103594E 00


  NEW NI =        3
  P =     ,46381882E 00
  SUM NI*UNI =   ,43000000E 02
  SUM NI*VNI =   ,24000000E 02
  SUM NI*WNI =   ,35000000E 02
  PI( 1) =     ,99943000E 00
  PI( 2) =     ,94064000E 00
  PI( 3) =     ,88185000E 00
  PI( 4) =     ,82306000E 00
  PI( 5) =     ,85949249E 00
  PI( 6) =     ,79086843E 00
1 DELTA PI( 1)/PI( 1) =      ,56935020E-03
1 DELTA PI( 2)/PI( 2) =      ,52312781E-01
1 DELTA PI( 3)/PI( 3) =      ,90231155E-01
1 DELTA PI( 4)/PI( 4) =      ,11432447E 00
3 DELTA PI( 5)/PI( 5) =      ,11329285E 00
3 DELTA PI( 6)/PI( 6) =      ,16376308E 00
```

```
  NEW NI =        4
  P =     ,53977522E 00
  SUM NI*UNI =   ,49000000E 02
  SUM NI*VNI =   ,27000000E 02
  SUM NI*WNI =   ,40000000E 02
  PI( 1) =     ,99943000E 00
  PI( 2) =     ,94064000E 00
  PI( 3) =     ,88185000E 00
  PI( 4) =     ,82306000E 00
  PI( 5) =     ,85949249E 00
  PI( 6) =     ,92038349E 00
1 DELTA PI( 1)/PI( 1) =      ,56935020E-03
1 DELTA PI( 2)/PI( 2) =      ,52312781E-01
1 DELTA PI( 3)/PI( 3) =      ,90231155E-01
1 DELTA PI( 4)/PI( 4) =      ,11432447E 00
3 DELTA PI( 5)/PI( 5) =      ,11329285E 00
4 DELTA PI( 6)/PI( 6) =      ,55259247E-01


  NEW NI =        3
  P =     ,60146474E 00
  SUM NI*UNI =   ,57000000E 02
  SUM NI*VNI =   ,31000000E 02
  SUM NI*WNI =   ,44000000E 02
```

TABLE B.1.1. (Continued)

```
    PI( 1) =    .99943000E 00
    PI( 2) =    .94064000E 00
    PI( 3) =    .88185000E 00
    PI( 4) =    .91715590E 00
    PI( 5) =    .85949249E 00
    PI( 6) =    .92038349E 00
1   DELTA PI( 1)/PI( 1) =    .56935020E-03
1   DELTA PI( 2)/PI( 2) =    .52312781E-01
1   DELTA PI( 3)/PI( 3) =    .90231155E-01
3   DELTA PI( 4)/PI( 4) =    .69373424E-01
3   DELTA PI( 5)/PI( 5) =    .11329285E 00
4   DELTA PI( 6)/PI( 6) =    .55259247E-01
```

```
    NEW NI =      4
    P =    .66962866E 00
    SUM NI*UNI =   .62000000E 02
    SUM NI*VNI =   .34000000E 02
    SUM NI*WNI =   .48000000E 02
    PI( 1) =    .99943000E 00
    PI( 2) =    .94064000E 00
    PI( 3) =    .88185000E 00
    PI( 4) =    .91715590E 00
    PI( 5) =    .95686654E 00
    PI( 6) =    .92038349E 00
1   DELTA PI( 1)/PI( 1) =    .56935020E-03
1   DELTA PI( 2)/PI( 2) =    .52312781E-01
1   DELTA PI( 3)/PI( 3) =    .90231155E-01
3   DELTA PI( 4)/PI( 4) =    .69373424E-01
4   DELTA PI( 5)/PI( 5) =    .31985024E-01
4   DELTA PI( 6)/PI( 6) =    .55259247E-01
```

```
    NEW NI =      3
    P =    .73005003E 00
    SUM NI*UNI =   .68000000E 02
    SUM NI*VNI =   .38000000E 02
    SUM NI*WNI =   .52000000E 02
    PI( 1) =    .99943000E 00
    PI( 2) =    .94064000E 00
    PI( 3) =    .96142034E 00
    PI( 4) =    .91715590E 00
    PI( 5) =    .95686684E 00
    PI( 6) =    .92038349E 00
1   DELTA PI( 1)/PI( 1) =    .56935020E-03
1   DELTA PI( 2)/PI( 2) =    .52312781E-01
3   DELTA PI( 3)/PI( 3) =    .33873872E-01
3   DELTA PI( 4)/PI( 4) =    .69373424E-01
4   DELTA PI( 5)/PI( 5) =    .31985024E-01
4   DELTA PI( 6)/PI( 6) =    .55259247E-01
```

```
    NEW NI =      4
    P =    .78069610E 00
    SUM NI*UNI =   .72000000E 02
    SUM NI*VNI =   .40000000E 02
    SUM NI*WNI =   .54000000E 02
    PI( 1) =    .99943000E 00
    PI( 2) =    .94064000E 00
    PI( 3) =    .96142034E 00
    PI( 4) =    .98078215E 00
    PI( 5) =    .95686684E 00
    PI( 6) =    .92038349E 00
1   DELTA PI( 1)/PI( 1) =    .55935020E-03
1   DELTA PI( 2)/PI( 2) =    .52312781E-01
3   DELTA PI( 3)/PI( 3) =    .33873872E-01
4   DELTA PI( 4)/PI( 4) =    .15304830E-01
4   DELTA PI( 5)/PI( 5) =    .31985024E-01
4   DELTA PI( 6)/PI( 6) =    .55259247E-01
```

```
    NEW NI =      5
    P =    .82383678E 00
    SUM NI*UNI =   .78000000E 02
    SUM NI*VNI =   .43000000E 02
    SUM NI*WNI =   .59000000E 02
    PI( 1) =    .99943000E 00
    PI( 2) =    .94064000E 00
    PI( 3) =    .96142034E 00
    PI( 4) =    .98078215E 00
    PI( 5) =    .95686684E 00
    PI( 6) =    .97124319E 00
1   DELTA PI( 1)/PI( 1) =    .55935020E-03
1   DELTA PI( 2)/PI( 2) =    .52312781E-01
3   DELTA PI( 3)/PI( 3) =    .33873872E-01
4   DELTA PI( 4)/PI( 4) =    .15304830E-01
4   DELTA PI( 5)/PI( 5) =    .31985024E-01
5   DELTA PI( 6)/PI( 6) =    .19276383E-01
```

```
    NEW NI =      3
    P =    .86693397E 00
    SUM NI*UNI =   .82000000E 02
    SUM NI*VNI =   .45000000E 02
    SUM NI*WNI =   .65000000E 02
    PI( 1) =    .99943000E 00
    PI( 2) =    .98984749E 00
    PI( 3) =    .96142034E 00
    PI( 4) =    .98078215E 00
    PI( 5) =    .95686684E 00
    PI( 6) =    .97124319E 00
```

TABLE B.1.1. (Concluded)

```
1  DELTA PI( 1)/PI( 1) =    ,56935020E-03
3  DELTA PI( 2)/PI( 2) =    ,94490389E-02
3  DELTA PI( 3)/PI( 3) =    ,33873872E-01
4  DELTA PI( 4)/PI( 4) =    ,15304830E-01
4  DELTA PI( 5)/PI( 5) =    ,31985024E-01
5  DELTA PI( 6)/PI( 6) =    ,19278383E-01


   NEW NI =      4
   P =    ,89630038E 00
   SUM NI*UNI =   ,85000000E 02
   SUM NI*VNI =   ,47000000E 02
   SUM NI*WNI =   ,67000000E 02
   PI( 1) =    ,99943000E 00
   PI( 2) =    ,98984749E 00
   PI( 3) =    ,99398737E 00
   PI( 4) =    ,98078215E 00
   PI( 5) =    ,95686684E 00
   PI( 6) =    ,97124319E 00
1  DELTA PI( 1)/PI( 1) =    ,56935020E-03
3  DELTA PI( 2)/PI( 2) =    ,94490389E-02
4  DELTA PI( 3)/PI( 3) =    ,51614264E-02
4  DELTA PI( 4)/PI( 4) =    ,15304830E-01
4  DELTA PI( 5)/PI( 5) =    ,31985024E-01
5  DELTA PI( 6)/PI( 6) =    ,19278383E-01
```

```
   NEW NI =      5
   P =    ,92496857E 00
   SUM NI*UNI =   ,90000000E 02
   SUM NI*VNI =   ,50000000E 02
   SUM NI*WNI =   ,71000000E 02
   PI( 1) =    ,99943000E 00
   PI( 2) =    ,98984749E 00
   PI( 3) =    ,99398737E 00
   PI( 4) =    ,98078215E 00
   PI( 5) =    ,98747225E 00
   PI( 6) =    ,97124319E 00
1  DELTA PI( 1)/PI( 1) =    ,56935020E-03
3  DELTA PI( 2)/PI( 2) =    ,94490389E-02
4  DELTA PI( 3)/PI( 3) =    ,51614264E-02
4  DELTA PI( 4)/PI( 4) =    ,15304830E-01
5  DELTA PI( 5)/PI( 5) =    ,91326782E-02
5  DELTA PI( 6)/PI( 6) =    ,19278383E-01

   NEW NI =      6
   P =    ,94230046E 00
   SUM NI*UNI =   ,96000000E 02
   SUM NI*VNI =   ,53000000E 02
   SUM NI*WNI =   ,76000000E 02
   PI( 1) =    ,99943000E 00
   PI( 2) =    ,98984749E 00
   PI( 3) =    ,99398737E 00
   PI( 4) =    ,98078215E 00
   PI( 5) =    ,98747225E 00
   PI( 6) =    ,98996718E 00
1  DELTA PI( 1)/PI( 1) =    ,56935020E-03
3  DELTA PI( 2)/PI( 2) =    ,94490389E-02
4  DELTA PI( 3)/PI( 3) =    ,51614264E-02
4  DELTA PI( 4)/PI( 4) =    ,15304830E-01
5  DELTA PI( 5)/PI( 5) =    ,91326782E-02
6  DELTA PI( 6)/PI( 6) =    ,66845753E-02


LIMIT ØN U EXCEEDED, ELIMINATE MØDULE( 4)
```

```
   NEW NI =      4
   P =    ,95170902E 00
   SUM NI*UNI =   ,98000000E 02
   SUM NI*VNI =   ,54000000E 02
   SUM NI*WNI =   ,79000000E 02
   PI( 1) =    ,99943000E 00
   PI( 2) =    ,99920060E 00
   PI( 3) =    ,99398737E 00
   PI( 4) =    ,98078215E 00
   PI( 5) =    ,98747225E 00
   PI( 6) =    ,98996718E 00
1  DELTA PI( 1)/PI( 1) =    ,56935020E-03
4  DELTA PI( 2)/PI( 2) =    ,74085952E-03
4  DELTA PI( 3)/PI( 3) =    ,51614264E-02
4  DELTA PI( 4)/PI( 4) =    ,00000000E 00
5  DELTA PI( 5)/PI( 5) =    ,91326782E-02
6  DELTA PI( 6)/PI( 6) =    ,66845753E-02


LIMIT ØN U EXCEEDED, ELIMINATE MØDULE( 5)
LIMIT ØN U EXCEEDED, ELIMINATE MØDULE( 6)
LIMIT ØN U EXCEEDED, ELIMINATE MØDULE( 3)
LIMIT ØN U EXCEEDED, ELIMINATE MØDULE( 2)
LIMIT ØN U EXCEEDED, ELIMINATE MØDULE( 1)

 ALL MØDULES ELIMINATED
```

TABLE B.1.2

SYSTEM OPTIMIZATION USING $(\Delta P)_{max} = \left(\frac{\Delta p_i}{p_i}\right)_{max}$

TO ACHIEVE A RELIABILITY GOAL OF 0.9995

```
INPUT DATA
N =      6
ICØN =     1
PLIMIT VALUE =        ,99950000
A =        ,33333000    B =        ,33333000    C =        ,33333000
U( 1) =        1,00000000
U( 2) =        2,00000000
U( 3) =        3,00000000
U( 4) =        4,00000000
U( 5) =        5,00000000
U( 6) =        6,00000000
V( 1) =        1,00000000
V( 2) =        1,00000000
V( 3) =        2,00000000
V( 4) =        2,00000000
V( 5) =        3,00000000
V( 6) =        3,00000000
W( 1) =        1,00000000
W( 2) =        3,00000000
W( 3) =        2,00000000
W( 4) =        2,00000000
W( 5) =        4,00000000
W( 6) =        5,00000000
RBAR( 1) =        ,00057000
RBAR( 2) =        ,05935000
RBAR( 3) =        ,11815000
RBAR( 4) =        ,17694000
RBAR( 5) =        ,23573000
RBAR( 6) =        ,29452000
CU =     99,00000000     CV =     57,00000000     CW =     83,00000000

NI =    1
SUM NI*UNI =   ,21000000E 02
SUM NI*VNI =   ,12000000E 02
SUM NI*WNI =   ,17000000E 02

INITIAL P VALUE =    ,36790319E 00
```

```
NEW NI =       3
P =    ,41374126E 00
SUM NI*UNI =    ,31000000E 02
SUM NI*VNI =    ,18000000E 02
SUM NI*WNI =    ,25000000E 02
PI( 1) =    ,99943000E 00
PI( 2) =    ,94064000E 00
PI( 3) =    ,88185000E 00
PI( 4) =    ,82306000E 00
PI( 5) =    ,85949249E 00
PI( 6) =    ,70548000E 00
1  DELTA PI( 1)/PI( 1) =    ,56935020E-03
1  DELTA PI( 2)/PI( 2) =    ,52312781E-01
1  DELTA PI( 3)/PI( 3) =    ,90231155E-01
1  DELTA PI( 4)/PI( 4) =    ,11432447E 00
3  DELTA PI( 5)/PI( 5) =    ,11329285E 00
1  DELTA PI( 6)/PI( 6) =    ,12103594E 00
```

```
NEW NI =       3
P =    ,46381682E 00
SUM NI*UNI =    ,43000000E 02
SUM NI*VNI =    ,24000000E 02
SUM NI*WNI =    ,35000000E 02
PI( 1) =    ,99943000E 00
PI( 2) =    ,94064000E 00
PI( 3) =    ,88185000E 00
PI( 4) =    ,82306000E 00
PI( 5) =    ,85949249E 00
PI( 6) =    ,79086843E 00
1  DELTA PI( 1)/PI( 1) =    ,56935020E-03
1  DELTA PI( 2)/PI( 2) =    ,52312781E-01
1  DELTA PI( 3)/PI( 3) =    ,90231155E-01
1  DELTA PI( 4)/PI( 4) =    ,11432447E 00
3  DELTA PI( 5)/PI( 5) =    ,11329285E 00
3  DELTA PI( 6)/PI( 6) =    ,16376308E 00
```

```
NEW NI =       4
P =    ,53977522E 00
SUM NI*UNI =    ,49000000E 02
SUM NI*VNI =    ,27000000E 02
SUM NI*WNI =    ,40000000E 02
PI( 1) =    ,99943000E 00
PI( 2) =    ,94064000E 00
PI( 3) =    ,88185000E 00
PI( 4) =    ,82306000E 00
PI( 5) =    ,85949249E 00
PI( 6) =    ,92038349E 00
1  DELTA PI( 1)/PI( 1) =    ,56935020E-03
1  DELTA PI( 2)/PI( 2) =    ,52312781E-01
1  DELTA PI( 3)/PI( 3) =    ,90231155E-01
1  DELTA PI( 4)/PI( 4) =    ,11432447E 00
3  DELTA PI( 5)/PI( 5) =    ,11329285E 00
4  DELTA PI( 6)/PI( 6) =    ,55259247E-01
```

```
NEW NI =       3
P =    ,60148474E 00
SUM NI*UNI =    ,57000000E 02
SUM NI*VNI =    ,31000000E 02
SUM NI*WNI =    ,44000000E 02
```

TABLE B.1.2. (Continued)

```
PI( 1) =  ,99943000E 00
PI( 2) =  ,94064000E 00
PI( 3) =  ,88185000E 00
PI( 4) =  ,91715590E 00
PI( 5) =  ,85949249E 00
PI( 6) =  ,92038349E 00
1  DELTA PI( 1)/PI( 1) =  ,56935020E-03
1  DELTA PI( 2)/PI( 2) =  ,52312781E-01
1  DELTA PI( 3)/PI( 3) =  ,90231155E-01
3  DELTA PI( 4)/PI( 4) =  ,69373424E-01
3  DELTA PI( 5)/PI( 5) =  ,11329285E 00
4  DELTA PI( 6)/PI( 6) =  ,55259247E-01
```

```
NEW NI =      4
P =   ,66962366E 00
SUM NI*UNI =  ,62000000E 02
SUM NI*VNI =  ,34000000E 02
SUM NI*WNI =  ,48000000E 02
PI( 1) =  ,99943000E 00
PI( 2) =  ,94064000E 00
PI( 3) =  ,88185000E 00
PI( 4) =  ,91715590E 00
PI( 5) =  ,95686684E 00
PI( 6) =  ,92038349E 00
1  DELTA PI( 1)/PI( 1) =  ,56935020E-03
1  DELTA PI( 2)/PI( 2) =  ,52312781E-01
1  DELTA PI( 3)/PI( 3) =  ,90231155E-01
3  DELTA PI( 4)/PI( 4) =  ,69373424E-01
4  DELTA PI( 5)/PI( 5) =  ,31985024E-01
4  DELTA PI( 6)/PI( 6) =  ,55259247E-01
```

```
NEW NI =      3
P =   ,73005903E 00
SUM NI*UNI =  ,68000000E 02
SUM NI*VNI =  ,38000000E 02
SUM NI*WNI =  ,52000000E 02
PI( 1) =  ,99943000E 00
PI( 2) =  ,94064000E 00
PI( 3) =  ,96142034E 00
PI( 4) =  ,91715590E 00
PI( 5) =  ,95686684E 00
PI( 6) =  ,92038349E 00
1  DELTA PI( 1)/PI( 1) =  ,56935020E-03
1  DELTA PI( 2)/PI( 2) =  ,52312781E-01
3  DELTA PI( 3)/PI( 3) =  ,33873872E-01
3  DELTA PI( 4)/PI( 4) =  ,69373424E-01
4  DELTA PI( 5)/PI( 5) =  ,31985024E-01
4  DELTA PI( 6)/PI( 6) =  ,55259247E-01
```

```
NEW NI =      4
P =   ,78069610E 00
SUM NI*UNI =  ,72000000E 02
SUM NI*VNI =  ,40000000E 02
SUM NI*WNI =  ,54000000E 02
PI( 1) =  ,99943000E 00
PI( 2) =  ,94064000E 00
PI( 3) =  ,96142034E 00
PI( 4) =  ,98078215E 00
PI( 5) =  ,95686684E 00
PI( 6) =  ,92038349E 00
1  DELTA PI( 1)/PI( 1) =  ,56935020E-03
1  DELTA PI( 2)/PI( 2) =  ,52312781E-01
3  DELTA PI( 3)/PI( 3) =  ,33873872E-01
4  DELTA PI( 4)/PI( 4) =  ,15304830E-01
4  DELTA PI( 5)/PI( 5) =  ,31985024E-01
4  DELTA PI( 6)/PI( 6) =  ,55259247E-01
```

```
NEW NI =      5
P =   ,82383678E 00
SUM NI*UNI =  ,78000000E 02
SUM NI*VNI =  ,43000000E 02
SUM NI*WNI =  ,59000000E 02
PI( 1) =  ,99943000E 00
PI( 2) =  ,94064000E 00
PI( 3) =  ,96142034E 00
PI( 4) =  ,98078215E 00
PI( 5) =  ,95686684E 00
PI( 6) =  ,97124319E 00
1  DELTA PI( 1)/PI( 1) =  ,56935020E-03
1  DELTA PI( 2)/PI( 2) =  ,52312781E-01
3  DELTA PI( 3)/PI( 3) =  ,33873872E-01
4  DELTA PI( 4)/PI( 4) =  ,15304830E-01
4  DELTA PI( 5)/PI( 5) =  ,31985024E-01
5  DELTA PI( 6)/PI( 6) =  ,19278383E-01
```

```
NEW NI =      3
P =   ,86693397E 00
SUM NI*UNI =  ,82000000E 02
SUM NI*VNI =  ,45000000E 02
SUM NI*WNI =  ,65000000E 02
PI( 1) =  ,99943000E 00
PI( 2) =  ,98984749E 00
PI( 3) =  ,96142034E 00
PI( 4) =  ,98078215E 00
PI( 5) =  ,95686684E 00
PI( 6) =  ,97124319E 00
```

TABLE B. 1. 2. (Continued)

```
1   DELTA PI( 1)/PI( 1) =    ,56935020E-03
3   DELTA PI( 2)/PI( 2) =    ,94490389E-02
3   DELTA PI( 3)/PI( 3) =    ,33873872E-01
4   DELTA PI( 4)/PI( 4) =    ,15304830E-01
4   DELTA PI( 5)/PI( 5) =    ,31985024E-01
5   DELTA PI( 6)/PI( 6) =    ,19278383E-01
```

```
    NEW NI =      4
    P =    ,89630038E 00
    SUM NI*UNI =  ,85000000E 02
    SUM NI*VNI =  ,47000000E 02
    SUM NI*WNI =  ,67000000E 02
    PI( 1) =   ,99943000E 00
    PI( 2) =   ,98984749E 00
    PI( 3) =   ,99398737E 00
    PI( 4) =   ,98078215E 00
    PI( 5) =   ,95686684E 00
    PI( 6) =   ,97124319E 00
1   DELTA PI( 1)/PI( 1) =    ,56935020E-03
3   DELTA PI( 2)/PI( 2) =    ,94490389E-02
4   DELTA PI( 3)/PI( 3) =    ,51614264E-02
4   DELTA PI( 4)/PI( 4) =    ,15304830E-01
4   DELTA PI( 5)/PI( 5) =    ,31985024E-01
5   DELTA PI( 6)/PI( 6) =    ,19278383E-01
```

```
    NEW NI =      5
    P =    ,92496857E 00
    SUM NI*UNI =  ,90000000E 02
    SUM NI*VNI =  ,50000000E 02
    SUM NI*WNI =  ,71000000E 02
    PI( 1) =   ,99943000E 00
    PI( 2) =   ,98984749E 00
    PI( 3) =   ,99398737E 00
    PI( 4) =   ,98078215E 00
    PI( 5) =   ,98747225E 00
    PI( 6) =   ,97124319E 00
1   DELTA PI( 1)/PI( 1) =    ,56935020E-03
3   DELTA PI( 2)/PI( 2) =    ,94490389E-02
4   DELTA PI( 3)/PI( 3) =    ,51614264E-02
4   DELTA PI( 4)/PI( 4) =    ,15304830E-01
5   DELTA PI( 5)/PI( 5) =    ,91326782E-02
5   DELTA PI( 6)/PI( 6) =    ,19278383E-01
```

```
    NEW NI =      6
    P =    ,94280046E 00
    SUM NI*UNI =  ,96000000E 02
    SUM NI*VNI =  ,53000000E 02
    SUM NI*WNI =  ,76000000E 02
    PI( 1) =   ,99943000E 00
    PI( 2) =   ,98984749E 00
    PI( 3) =   ,99398737E 00
    PI( 4) =   ,98078215F 00
    PI( 5) =   ,98747225E 00
    PI( 6) =   ,98996718E 00
1   DELTA PI( 1)/PI( 1) =    ,56935020E-03
3   DELTA PI( 2)/PI( 2) =    ,94490389E-02
4   DELTA PI( 3)/PI( 3) =    ,51614264E-02
4   DELTA PI( 4)/PI( 4) =    ,15304830E-01
5   DELTA PI( 5)/PI( 5) =    ,91326782E-02
6   DELTA PI( 6)/PI( 6) =    ,66845753E-02
```

```
    NEW NI =      5
    P =    ,95722987E 00
    SUM NI*UNI =  ,10000000E 03
    SUM NI*VNI =  ,55000000E 02
    SUM NI*WNI =  ,78000000E 02
    PI( 1) =   ,99943000E 00
    PI( 2) =   ,98984749E 00
    PI( 3) =   ,99398737E 00
    PI( 4) =   ,99579285E 00
    PI( 5) =   ,98747225E 00
    PI( 6) =   ,98996718E 00
1   DELTA PI( 1)/PI( 1) =    ,56935020E-03
3   DELTA PI( 2)/PI( 2) =    ,94490389E-02
4   DELTA PI( 3)/PI( 3) =    ,51614264E-02
5   DELTA PI( 4)/PI( 4) =    ,33340191E-02
5   DELTA PI( 5)/PI( 5) =    ,91326782E-02
6   DELTA PI( 6)/PI( 6) =    ,66845753E-02
```

```
    NEW NI =      4
    P =    ,96627477E 00
    SUM NI*UNI =  ,10200000E 03
    SUM NI*VNI =  ,56000000E 02
    SUM NI*WNI =  ,81000000E 02
    PI( 1) =   ,99943000E 00
    PI( 2) =   ,99920060E 00
    PI( 3) =   ,99398737E 00
    PI( 4) =   ,99579285E 00
    PI( 5) =   ,98747225E 00
    PI( 6) =   ,98996718E 00
1   DELTA PI( 1)/PI( 1) =    ,56935020E-03
4   DELTA PI( 2)/PI( 2) =    ,74085952E-03
4   DELTA PI( 3)/PI( 3) =    ,51614264E-02
5   DELTA PI( 4)/PI( 4) =    ,33340191E-02
5   DELTA PI( 5)/PI( 5) =    ,91326782E-02
6   DELTA PI( 6)/PI( 6) =    ,66845753E-02
```

TABLE B.1.2. (Continued)

	NEW NI =	6
	P =	.97509944E 00
	SUM NI*UNI =	.10700000E 03
	SUM NI*VNI =	.59000000E 02
	SUM NI*WNI =	.85000000E 02
	PI(1) =	.99943000E 00
	PI(2) =	.99920060E 00
	PI(3) =	.99398737E 00
	PI(4) =	.99579285E 00
	PI(5) =	.99649052E 00
	PI(6) =	.98996718E 00
1	DELTA PI(1)/PI(1) =	.56935020E-03
4	DELTA PI(2)/PI(2) =	.74085952E-03
4	DELTA PI(3)/PI(3) =	.51614264E-02
5	DELTA PI(4)/PI(4) =	.33340191E-02
6	DELTA PI(5)/PI(5) =	.25600355E-02
6	DELTA PI(6)/PI(6) =	.66845753E-02

	NEW NI =	5
	P =	.98668412E 00
	SUM NI*UNI =	.11600000E 03
	SUM NI*VNI =	.64000000E 02
	SUM NI*WNI =	.92000000E 02
	PI(1) =	.99943000E 00
	PI(2) =	.99920060E 00
	PI(3) =	.99911777E 00
	PI(4) =	.99579285E 00
	PI(5) =	.99649052E 00
	PI(6) =	.99658469E 00
1	DELTA PI(1)/PI(1) =	.56935020E-03
4	DELTA PI(2)/PI(2) =	.74085952E-03
5	DELTA PI(3)/PI(3) =	.75836392E-03
5	DELTA PI(4)/PI(4) =	.33340191E-02
6	DELTA PI(5)/PI(5) =	.25600355E-02
7	DELTA PI(6)/PI(6) =	.22816130E-02

	NEW NI =	7
	P =	.99250811E 00
	SUM NI*UNI =	.12500000E 03
	SUM NI*VNI =	.69000000E 02
	SUM NI*WNI =	.98000000E 02
	PI(1) =	.99943000E 00
	PI(2) =	.99920060E 00
	PI(3) =	.99911777E 00
	PI(4) =	.99911284E 00
	PI(5) =	.99904157E 00
	PI(6) =	.99658469E 00
1	DELTA PI(1)/PI(1) =	.56935020E-03
4	DELTA PI(2)/PI(2) =	.74085952E-03
5	DELTA PI(3)/PI(3) =	.75836392E-03
6	DELTA PI(4)/PI(4) =	.70555328E-03
7	DELTA PI(5)/PI(5) =	.70225889E-03
7	DELTA PI(6)/PI(6) =	.22816130E-02

	NEW NI =	7
	P =	.98161757E 00
	SUM NI*UNI =	.11300000E 03
	SUM NI*VNI =	.62000000E 02
	SUM NI*WNI =	.90000000E 02
	PI(1) =	.99943000E 00
	PI(2) =	.99920060E 00
	PI(3) =	.99398737E 00
	PI(4) =	.99579285E 00
	PI(5) =	.99649052E 00
	PI(6) =	.99658469E 00
1	DELTA PI(1)/PI(1) =	.56935020E-03
4	DELTA PI(2)/PI(2) =	.74085952E-03
4	DELTA PI(3)/PI(3) =	.51614264E-02
5	DELTA PI(4)/PI(4) =	.33340191E-02
6	DELTA PI(5)/PI(5) =	.25600355E-02
7	DELTA PI(6)/PI(6) =	.22816130E-02

	NEW NI =	6
	P =	.98997374E 00
	SUM NI*UNI =	.12000000E 03
	SUM NI*VNI =	.66000000E 02
	SUM NI*WNI =	.94000000E 02
	PI(1) =	.99943000E 00
	PI(2) =	.99920060E 00
	PI(3) =	.99911777E 00
	PI(4) =	.99911284E 00
	PI(5) =	.99649052E 00
	PI(6) =	.99658469E 00
1	DELTA PI(1)/PI(1) =	.56935020E-03
4	DELTA PI(2)/PI(2) =	.74085952E-03
5	DELTA PI(3)/PI(3) =	.75836392E-03
6	DELTA PI(4)/PI(4) =	.70555328E-03
6	DELTA PI(5)/PI(5) =	.25600355E-02
7	DELTA PI(6)/PI(6) =	.22816130E-02

	NEW NI =	8
	P =	.99477263E 00
	SUM NI*UNI =	.13100000E 03
	SUM NI*VNI =	.72000000E 02
	SUM NI*WNI =	.10300000E 03
	PI(1) =	.99943000E 00
	PI(2) =	.99920060E 00
	PI(3) =	.99911777E 00
	PI(4) =	.99911284E 00
	PI(5) =	.99904157E 00
	PI(6) =	.99885852E 00
1	DELTA PI(1)/PI(1) =	.56935020E-03
4	DELTA PI(2)/PI(2) =	.74085952E-03
5	DELTA PI(3)/PI(3) =	.75836392E-03
6	DELTA PI(4)/PI(4) =	.70555328E-03
7	DELTA PI(5)/PI(5) =	.70225889E-03
8	DELTA PI(6)/PI(6) =	.76622967E-03

TABLE B.1.2. (Continued)

```
    NEW NI =       9
    P =     ,99553485E 00
    SUM NI*UNI =  ,13700000E 03
    SUM NI*VNI =  ,75000000E 02
    SUM NI*WNI =  ,10800000E 03
    PI( 1) =      ,99943000E 00
    PI( 2) =      ,99920060E 00
    PI( 3) =      ,99911777E 00
    PI( 4) =      ,99911284E 00
    PI( 5) =      ,99904157E 00
    PI( 6) =      ,99962387E 00
1   DELTA PI( 1)/PI( 1) =     ,56935020E-03
4   DELTA PI( 2)/PI( 2) =     ,74085952E-03
5   DELTA PI( 3)/PI( 3) =     ,75636392E-03
6   DELTA PI( 4)/PI( 4) =     ,70555328E-03
7   DELTA PI( 5)/PI( 5) =     ,70225889E-03
9   DELTA PI( 6)/PI( 6) =     ,25368433E-03
```

```
    NEW NI =       5
    P =     ,99702794E 00
    SUM NI*UNI =  ,14200000E 03
    SUM NI*VNI =  ,78000000E 02
    SUM NI*WNI =  ,11300000E 03
    PI( 1) =      ,99943000E 00
    PI( 2) =      ,99994087E 00
    PI( 3) =      ,99987546E 00
    PI( 4) =      ,99911284E 00
    PI( 5) =      ,99904157E 00
    PI( 6) =      ,99962387E 00
1   DELTA PI( 1)/PI( 1) =     ,56935020E-03
5   DELTA PI( 2)/PI( 2) =     ,54931080E-04
6   DELTA PI( 3)/PI( 3) =     ,10743936E-03
6   DELTA PI( 4)/PI( 4) =     ,70555328E-03
7   DELTA PI( 5)/PI( 5) =     ,70225889E-03
9   DELTA PI( 6)/PI( 6) =     ,25368433E-03
```

```
    NEW NI =       8
    P =     ,99643206E 00
    SUM NI*UNI =  ,15100000E 03
    SUM NI*VNI =  ,83000000E 02
    SUM NI*WNI =  ,11900000E 03
    PI( 1) =      ,99943000E 00
    PI( 2) =      ,99994087E 00
    PI( 3) =      ,99987546E 00
    PI( 4) =      ,99981777E 00
    PI( 5) =      ,99974316E 00
    PI( 6) =      ,99962387E 00
1   DELTA PI( 1)/PI( 1) =     ,56935020E-03
5   DELTA PI( 2)/PI( 2) =     ,54931080E-04
6   DELTA PI( 3)/PI( 3) =     ,10743936E-03
7   DELTA PI( 4)/PI( 4) =     ,14554467E-03
8   DELTA PI( 5)/PI( 5) =     ,18905979E-03
9   DELTA PI( 6)/PI( 6) =     ,25368433E-03
```

```
    NEW NI =       6
    P =     ,99628983E 00
    SUM NI*UNI =  ,14000000E 03
    SUM NI*VNI =  ,77000000E 02
    SUM NI*WNI =  ,11000000E 03
    PI( 1) =      ,99943000E 00
    PI( 2) =      ,99920060E 00
    PI( 3) =      ,99987546E 00
    PI( 4) =      ,99911284E 00
    PI( 5) =      ,99904157E 00
    PI( 6) =      ,99962387E 00
1   DELTA PI( 1)/PI( 1) =     ,56935020E-03
4   DELTA PI( 2)/PI( 2) =     ,74085952E-03
6   DELTA PI( 3)/PI( 3) =     ,10743936E-03
6   DELTA PI( 4)/PI( 4) =     ,70555328E-03
7   DELTA PI( 5)/PI( 5) =     ,70225889E-03
9   DELTA PI( 6)/PI( 6) =     ,25368433E-03
```

```
    NEW NI =       7
    P =     ,99773140E 00
    SUM NI*UNI =  ,14600000E 03
    SUM NI*VNI =  ,80000000E 02
    SUM NI*WNI =  ,11500000E 03
    PI( 1) =      ,99943000E 00
    PI( 2) =      ,99994087E 00
    PI( 3) =      ,99987546E 00
    PI( 4) =      ,99981777E 00
    PI( 5) =      ,99904157E 00
    PI( 6) =      ,99962387E 00
1   DELTA PI( 1)/PI( 1) =     ,56935020E-03
5   DELTA PI( 2)/PI( 2) =     ,54931080E-04
6   DELTA PI( 3)/PI( 3) =     ,10743936E-03
7   DELTA PI( 4)/PI( 4) =     ,14554467E-03
7   DELTA PI( 5)/PI( 5) =     ,70225889E-03
9   DELTA PI( 6)/PI( 6) =     ,25368433E-03
```

```
    NEW NI =       3
    P =     ,99900052E 00
    SUM NI*UNI =  ,15300000E 03
    SUM NI*VNI =  ,85000000E 02
    SUM NI*WNI =  ,12100000E 03
    PI( 1) =      ,99999903E 00
    PI( 2) =      ,99994087E 00
    PI( 3) =      ,99987546E 00
    PI( 4) =      ,99981777E 00
    PI( 5) =      ,99974316E 00
    PI( 6) =      ,99962387E 00
3   DELTA PI( 1)/PI( 1) =     ,97359011E-06
5   DELTA PI( 2)/PI( 2) =     ,54931080E-04
6   DELTA PI( 3)/PI( 3) =     ,10743936E-03
7   DELTA PI( 4)/PI( 4) =     ,14554467E-03
8   DELTA PI( 5)/PI( 5) =     ,18905979E-03
9   DELTA PI( 6)/PI( 6) =     ,25368433E-03
```

TABLE B. 1. 2. (Concluded)

```
      NEW NI =       10
      P =     ,99925395E 00
      SUM NI*UNI =   ,15900000E 03
      SUM NI*VNI =   ,88000000E 02
      SUM NI*WNI =   ,12600000E 03
      PI( 1) =     ,99999903E 00
      PI( 2) =     ,99994087E 00
      PI( 3) =     ,99987546E 00
      PI( 4) =     ,99981777E 00
      PI( 5) =     ,99974316E 00
      PI( 6) =     ,99987746E 00
  3   DELTA PI( 1)/PI( 1) =     ,97359011E-06
  5   DELTA PI( 2)/PI( 2) =     ,54931080E-04
  6   DELTA PI( 3)/PI( 3) =     ,10743936E-03
  7   DELTA PI( 4)/PI( 4) =     ,14554467E-03
  8   DELTA PI( 5)/PI( 5) =     ,18905979E-03
 10   DELTA PI( 6)/PI( 6) =     ,82995732E-04
```

```
      NEW NI =       9
      P =     ,99944287E 00
      SUM NI*UNI =   ,16400000E 03
      SUM NI*VNI =   ,91000000E 02
      SUM NI*WNI =   ,13000000E 03
      PI( 1) =     ,99999903E 00
      PI( 2) =     ,99994087E 00
      PI( 3) =     ,99987546E 00
      PI( 4) =     ,99981777E 00
      PI( 5) =     ,99993217E 00
      PI( 6) =     ,99987746E 00
  3   DELTA PI( 1)/PI( 1) =     ,97359011E-06
  5   DELTA PI( 2)/PI( 2) =     ,54931080E-04
  6   DELTA PI( 3)/PI( 3) =     ,10743936E-03
  7   DELTA PI( 4)/PI( 4) =     ,14554467E-03
  9   DELTA PI( 5)/PI( 5) =     ,50128469E-04
 10   DELTA PI( 6)/PI( 6) =     ,82995732E-04
```

```
      NEW NI =       8
      P =     ,99958833E 00
      SUM NI*UNI =   ,16800000E 03
      SUM NI*VNI =   ,93000000E 02
      SUM NI*WNI =   ,13200000E 03
      PI( 1) =     ,99999903E 00
      PI( 2) =     ,99994087E 00
      PI( 3) =     ,99987546E 00
      PI( 4) =     ,99996329E 00
      PI( 5) =     ,99993217E 00
      PI( 6) =     ,99987746E 00
  3   DELTA PI( 1)/PI( 1) =     ,97359011E-06
  5   DELTA PI( 2)/PI( 2) =     ,54931080E-04
  6   DELTA PI( 3)/PI( 3) =     ,10743936E-03
  8   DELTA PI( 4)/PI( 4) =     ,29427345E-04
  9   DELTA PI( 5)/PI( 5) =     ,50128469E-04
 10   DELTA PI( 6)/PI( 6) =     ,82995732E-04


CØNATRAINT LIMIT REACHED
```

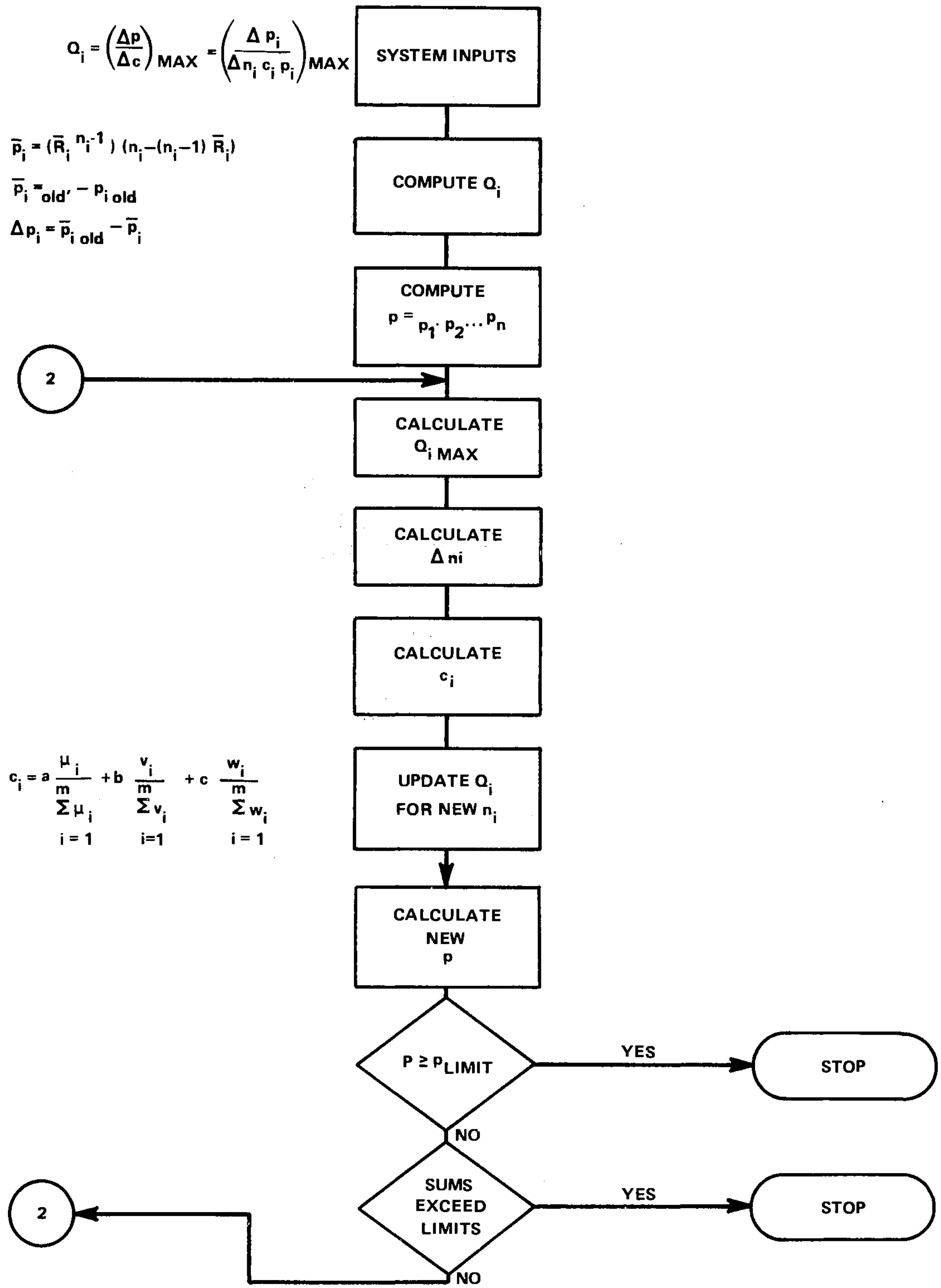


Figure B.2.1. Logic Diagram of Computer Program for Optimization Process Utilizing

$$\left(\frac{\Delta P}{\Delta C}\right)_{max} = \left(\frac{\Delta p_i}{\Delta n_i\, c_i\, p_i}\right)_{max}$$

TABLE B.2.1

SYSTEM OPTIMIZATION USING $\left(\frac{\Delta P}{\Delta C}\right)_{max} = \left(\frac{\Delta p_i}{\Delta n_i \, c_i \, p_i}\right)_{max}$

AND CONSTRAINTS $C_u = 99$, $C_v = 57$, AND $C_w = 83$

```
                    INPUT DATA

N =        6
ICØN =       0
PLIMIT VALUE =          ,99950000
A =          ,33333000    B =          ,33333000    C =          ,33333000
U( 1) =         1,00000000
U( 2) =         2,00000000
U( 3) =         3,00000000
U( 4) =         4,00000000
U( 5) =         5,00000000
U( 6) =         6,00000000
V( 1) =         1,00000000
V( 2) =         1,00000000
V( 3) =         2,00000000
V( 4) =         2,00000000
V( 5) =         3,00000000
V( 6) =         3,00000000
W( 1) =         1,00000000
W( 2) =         3,00000000
W( 3) =         2,00000000
W( 4) =         2,00000000
W( 5) =         4,00000000
W( 6) =         5,00000000
RBAR( 1) =           ,00057000
RBAR( 2) =           ,05936000
RBAR( 3) =           ,11815000
RBAR( 4) =           ,17694000
RBAR( 5) =           ,23573000
RBAR( 6) =           ,29452000
CU =      99,00000000      CV =      57,00000000      CW =      83,00000000

NI =      1
SUM NI*UNI =   ,21000000E 02
SUM NI*VNI =   ,12000000E 02
SUM NI*WNI =   ,17000000E 02

INITIAL P VALUE =      ,36790319E 00

Q = DELTA PI/DELTA NI*CI*PI

NEW NI =         3
P =      ,40996353E 00
SUM NI*UNI =   ,29000000E 02
SUM NI*VNI =   ,16000000E 02
SUM NI*WNI =   ,21000000E 02
PI( 1) =     ,99943000E 00
PI( 2) =     ,94064000E 00
PI( 3) =     ,88185000E 00
PI( 4) =     ,91715590E 00
PI( 5) =     ,76427000E 00
PI( 6) =     ,70548000E 00
 C( 1) =     ,63258004E-01
 C( 2) =     ,11834616E 00
 C( 3) =     ,14238887E 00
 C( 4) =     ,15826172E 00
 C( 5) =     ,24112737E 00
 C( 6) =     ,27660788E 00
1  Q( 1) =   ,45002226E-02
1  Q( 2) =   ,22101597E 00
1  Q( 3) =   ,31684765E 00
3  Q( 4) =   ,43834619E 00
1  Q( 5) =   ,25835460E 00
1  Q( 6) =   ,21878614E 00

NEW NI =         4
P =      ,43840411E 00
SUM NI*UNI =   ,33000000E 02
SUM NI*VNI =   ,18000000E 02
SUM NI*WNI =   ,23000000E 02
PI( 1) =     ,99943000E 00
PI( 2) =     ,94064000E 00
PI( 3) =     ,88185000E 00
PI( 4) =     ,98078215E 00
PI( 5) =     ,76427000E 00
PI( 6) =     ,70548000E 00
C( 1) =     ,63258004E-01
C( 2) =     ,11834616E 00
C( 3) =     ,14238887E 00
C( 4) =     ,15826172E 00
C( 5) =     ,24112737E 00
C( 6) =     ,27660788E 00
1  Q( 1) =   ,45002226E-02
1  Q( 2) =   ,22101597E 00
1  Q( 3) =   ,31684765E 00
4  Q( 4) =   ,96705820E-01
1  Q( 5) =   ,25835460E 00
1  Q( 6) =   ,21878614E 00

NEW NI =         3
P =      ,47796182E 00
SUM NI*UNI =   ,39000000E 02
SUM NI*VNI =   ,22000000E 02
SUM NI*WNI =   ,27000000E 02
PI( 1) =     ,99943000E 00
PI( 2) =     ,94064000E 00
PI( 3) =     ,96142034E 00
PI( 4) =     ,98078215E 00
PI( 5) =     ,76427000E 00
PI( 6) =     ,70548000E 00
C( 1) =     ,63258004E-01
C( 2) =     ,11834616E 00
C( 3) =     ,14238887E 00
C( 4) =     ,15826172E 00
C( 5) =     ,24112737E 00
C( 6) =     ,27660788E 00
1  Q( 1) =   ,45002226E-02
1  Q( 2) =   ,22101597E 00
3  Q( 3) =   ,23789691E 00
4  Q( 4) =   ,96705820E-01
1  Q( 5) =   ,25835460E 00
1  Q( 6) =   ,21878614E 00
```

TABLE B.2.1. (Continued)

	NEW NI =	3
	P =	.53751238E 00
	SUM NI*UNI =	.49000000E 02
	SUM NI*VNI =	.28000000E 02
	SUM NI*WNI =	.35000000E 02
	PI(1) =	.99943000E 00
	PI(2) =	.94064000E 00
	PI(3) =	.96142034E 00
	PI(4) =	.98078215E 00
	PI(5) =	.85949249E 00
	PI(6) =	.70548000E 00
	C(1) =	.63258004E-01
	C(2) =	.11834616E 00
	C(3) =	.14238887E 00
	C(4) =	.15826172E 00
	C(5) =	.24112737E 00
	C(6) =	.27660788E 00
1	Q(1) =	.45002226E-02
1	Q(2) =	.22101597E 00
3	Q(3) =	.23789691E 00
4	Q(4) =	.96705820E-01
3	Q(5) =	.46984649E 00
1	Q(6) =	.21878614E 00

	NEW NI =	4
	P =	.61867911E 00
	SUM NI*UNI =	.57000000E 02
	SUM NI*VNI =	.33000000E 02
	SUM NI*WNI =	.41000000E 02
	PI(1) =	.99943000E 00
	PI(2) =	.94064000E 00
	PI(3) =	.99398737E 00
	PI(4) =	.98078215E 00
	PI(5) =	.95686684E 00
	PI(6) =	.70548000E 00
	C(1) =	.63258004E-01
	C(2) =	.11834616E 00
	C(3) =	.14238887E 00
	C(4) =	.15826172E 00
	C(5) =	.24112737E 00
	C(6) =	.27660788E 00
1	Q(1) =	.45002226E-02
1	Q(2) =	.22101597E 00
4	Q(3) =	.36248806E-01
4	Q(4) =	.96705820E-01
4	Q(5) =	.13264783E 00
1	Q(6) =	.21878614E 00

	NEW NI =	3
	P =	.72984365E 00
	SUM NI*UNI =	.73000000E 02
	SUM NI*VNI =	.41000000E 02
	SUM NI*WNI =	.57000000E 02
	PI(1) =	.99943000E 00
	PI(2) =	.98984749E 00
	PI(3) =	.99398737E 00
	PI(4) =	.98078215E 00
	PI(5) =	.95686684E 00
	PI(6) =	.79086843E 00
	C(1) =	.63258004E-01
	C(2) =	.11834616E 00
	C(3) =	.14238887E 00
	C(4) =	.15826172E 00
	C(5) =	.24112737E 00
	C(6) =	.27660788E 00
1	Q(1) =	.45002226E-02
3	Q(2) =	.79842382E-01
4	Q(3) =	.36248806E-01
4	Q(4) =	.96705820E-01
4	Q(5) =	.13264783E 00
3	Q(6) =	.59204056E 00

	NEW NI =	5
	P =	.89630038E 00
	SUM NI*UNI =	.85000000E 02
	SUM NI*VNI =	.47000000E 02
	SUM NI*WNI =	.67000000E 02
	PI(1) =	.99943000E 00
	PI(2) =	.98984749E 00
	PI(3) =	.99398737E 00
	PI(4) =	.98078215E 00
	PI(5) =	.95686684E 00
	PI(6) =	.97124319E 00
	C(1) =	.63258004E-01
	C(2) =	.11834616E 00
	C(3) =	.14238887E 00
	C(4) =	.15826172E 00
	C(5) =	.24112737E 00
	C(6) =	.27660788E 00
1	Q(1) =	.45002226E-02
3	Q(2) =	.79842382E-01
4	Q(3) =	.36248806E-01
4	Q(4) =	.96705820E-01
4	Q(5) =	.13264783E 00
5	Q(6) =	.69695711E-01

	NEW NI =	4
	P =	.59840870E 00
	SUM NI*UNI =	.54000000E 02
	SUM NI*VNI =	.31000000E 02
	SUM NI*WNI =	.39000000E 02
	PI(1) =	.99943000E 00
	PI(2) =	.94064000E 00
	PI(3) =	.96142034E 00
	PI(4) =	.98078215E 00
	PI(5) =	.95686684E 00
	PI(6) =	.70548000E 00
	C(1) =	.63258004E-01
	C(2) =	.11834616E 00
	C(3) =	.14238887E 00
	C(4) =	.15826172E 00
	C(5) =	.24112737E 00
	C(6) =	.27660788E 00
1	Q(1) =	.45002226E-02
1	Q(2) =	.22101597E 00
3	Q(3) =	.23789691E 00
4	Q(4) =	.96705820E-01
4	Q(5) =	.13264783E 00
1	Q(6) =	.21878614E 00

	NEW NI =	3
	P =	.65104394E 00
	SUM NI*UNI =	.61000000E 02
	SUM NI*VNI =	.35000000E 02
	SUM NI*WNI =	.47000000E 02
	PI(1) =	.99943000E 00
	PI(2) =	.98984749E 00
	PI(3) =	.99398737E 00
	PI(4) =	.98078215E 00
	PI(5) =	.95686684E 00
	PI(6) =	.70548000E 00
	C(1) =	.63258004E-01
	C(2) =	.11834616E 00
	C(3) =	.14238887E 00
	C(4) =	.15826172E 00
	C(5) =	.24112737E 00
	C(6) =	.27660788E 00
1	Q(1) =	.45002226E-02
3	Q(2) =	.79842382E-01
4	Q(3) =	.36248806E-01
4	Q(4) =	.96705820E-01
4	Q(5) =	.13264783E 00
1	Q(6) =	.21878614E 00

	NEW NI =	4
	P =	.84936510E 00
	SUM NI*UNI =	.79000000E 02
	SUM NI*VNI =	.44000000E 02
	SUM NI*WNI =	.62000000E 02
	PI(1) =	.99943000E 00
	PI(2) =	.98984749E 00
	PI(3) =	.99398737E 00
	PI(4) =	.98078215E 00
	PI(5) =	.95686684E 00
	PI(6) =	.92038349E 00
	C(1) =	.63258004E-01
	C(2) =	.11834616E 00
	C(3) =	.14238887E 00
	C(4) =	.15826172E 00
	C(5) =	.24112737E 00
	C(6) =	.27660788E 00
1	Q(1) =	.45002226E-02
3	Q(2) =	.79842382E-01
4	Q(3) =	.36248806E-01
4	Q(4) =	.96705820E-01
4	Q(5) =	.13264783E 00
4	Q(6) =	.19977467E 00

	NEW NI =	5
	P =	.92496857E 00
	SUM NI*UNI =	.90000000E 02
	SUM NI*VNI =	.50000000E 02
	SUM NI*WNI =	.71000000E 02
	PI(1) =	.99943000E 00
	PI(2) =	.98984749E 00
	PI(3) =	.99398737E 00
	PI(4) =	.98078215E 00
	PI(5) =	.96747225E 00
	PI(6) =	.97124319E 00
	C(1) =	.63258004E-01
	C(2) =	.11834616E 00
	C(3) =	.14238887E 00
	C(4) =	.15826172E 00
	C(5) =	.24112737E 00
	C(6) =	.27660788E 00
1	Q(1) =	.45002226E-02
3	Q(2) =	.79842382E-01
4	Q(3) =	.36248806E-01
4	Q(4) =	.96705820E-01
5	Q(5) =	.37874912E-01
5	Q(6) =	.69695711E-01

TABLE B.2.1. (Concluded)

```
   NEW NI  =       5
   P  =     .93912505E 00
   SUM NI*UNI  =   .94000000E 02
   SUM NI*VNI  =   .52000000E 02
   SUM NI*WNI  =   .73000000E 02
   PI( 1)  =     .99943000E 00
   PI( 2)  =     .98964749E 00
   PI( 3)  =     .99398737E 00
   PI( 4)  =     .99579285E 00
   PI( 5)  =     .98747225E 00
   PI( 6)  =     .97124319E 00
   C( 1)  =    .63258004E-01
   C( 2)  =    .11834616E 00
   C( 3)  =    .14238887E 00
   C( 4)  =    .15826172E 00
   C( 5)  =    .24112737E 00
   C( 6)  =    .27660788E 00
1  Q( 1)  =    .45002226E-02
3  Q( 2)  =    .79842382E-01
4  Q( 3)  =    .36248806E-01
5  Q( 4)  =    .21066491E-01
5  Q( 5)  =    .37874912E-01
5  Q( 6)  =    .69695711E-01
```

```
   NEW NI  =       4
   P  =     .94799888E 00
   SUM NI*UNI  =   .96000000E 02
   SUM NI*VNI  =   .53000000E 02
   SUM NI*WNI  =   .76000000E 02
   PI( 1)  =     .99943000E 00
   PI( 2)  =     .99920060E 00
   PI( 3)  =     .99398737E 00
   PI( 4)  =     .99579285E 00
   PI( 5)  =     .98747225E 00
   PI( 6)  =     .97124319E 00
   C( 1)  =    .63258004E-01
   C( 2)  =    .11834616E 00
   C( 3)  =    .14238887E 00
   C( 4)  =    .15826172E 00
   C( 5)  =    .24112737E 00
   C( 6)  =    .27660788E 00
1  Q( 1)  =    .45002226E-02
4  Q( 2)  =    .62601064E-02
4  Q( 3)  =    .36248806E-01
5  Q( 4)  =    .21066491E-01
5  Q( 5)  =    .37874912E-01
5  Q( 6)  =    .69695711E-01
```

```
LIMIT ØN U EXCEEDED, ELIMINATE MØDULE( 6)
LIMIT ØN U EXCEEDED, ELIMINATE MØDULE( 5)

   NEW NI  =       5
   P  =     .95289191E 00
   SUM NI*UNI  =   .99000000E 02
   SUM NI*VNI  =   .55000000E 02
   SUM NI*WNI  =   .78000000E 02
   PI( 1)  =     .99943000E 00
   PI( 2)  =     .99920060E 00
   PI( 3)  =     .99911777E 00
   PI( 4)  =     .99579285E 00
   PI( 5)  =     .98747225E 00
   PI( 6)  =     .97124319E 00
   C( 1)  =    .63258004E-01
   C( 2)  =    .11834616E 00
   C( 3)  =    .14238887E 00
   C( 4)  =    .15826172E 00
   C( 5)  =    .24112737E 00
   C( 6)  =    .27660788E 00
1  Q( 1)  =    .45002226E-02
4  Q( 2)  =    .62601064E-02
5  Q( 3)  =    .53260058E-02
5  Q( 4)  =    .21066491E-01
5  Q( 5)  =    .00000000E 00
5  Q( 6)  =    .00000000E 00

LIMIT ØN U EXCEEDED, ELIMINATE MØDULE( 4)
LIMIT ØN U EXCEEDED, ELIMINATE MØDULE( 2)
LIMIT ØN U EXCEEDED, ELIMINATE MØDULE( 3)
LIMIT ØN U EXCEEDED, ELIMINATE MØDULE( 1)

 ALL MØDULES ELIMINATED
```

TABLE B.2.2

SYSTEM OPTIMIZATION USING $\left(\frac{\Delta P}{\Delta C}\right)_{max} = \left(\frac{\Delta p_i}{\Delta n_i\, c_i\, p_i}\right)_{max}$

TO ACHIEVE A RELIABILITY GOAL OF 0.9995

```
                    INPUT DATA

N =      6
ICØN =      1
PLIMIT VALUE =          ,99950000
A =           ,33333000    B =           ,33333000    C =           ,33333000
U( 1) =         1,00000000
U( 2) =         2,00000000
U( 3) =         3,00000000
U( 4) =         4,00000000
U( 5) =         5,00000000
U( 6) =         6,00000000
V( 1) =         1,00000000
V( 2) =         1,00000000
V( 3) =         2,00000000
V( 4) =         2,00000000
V( 5) =         3,00000000
V( 6) =         3,00000000
W( 1) =         1,00000000
W( 2) =         3,00000000
W( 3) =         2,00000000
W( 4) =         2,00000000
W( 5) =         4,00000000
W( 6) =         5,00000000
RBAR( 1) =          ,00057000
RBAR( 2) =          ,05936000
RBAR( 3) =          ,11815000
RBAR( 4) =          ,17694000
RBAR( 5) =          ,23573000
RBAR( 6) =          ,29452000
CU =      99,00000000      CV =      57,00000000      CW =      83,00000000

NI =      1
SUM NI*UNI =    ,21000000E 02
SUM NI*VNI =    ,12000000E 02
SUM NI*WNI =    ,17000000E 02

INITIAL P VALUE =     ,36790319E 00

Q = DELTA PI/DELTA NI*CI*PI
```

```
   NEW NI =      3
   P =     ,40996353E 00
   SUM NI*UNI =    ,29000000E 02
   SUM NI*VNI =    ,16000000E 02
   SUM NI*WNI =    ,21000000E 02
   PI( 1) =     ,99943000E 00
   PI( 2) =     ,94064000E 00
   PI( 3) =     ,88185000E 00
   PI( 4) =     ,91715590E 00
   PI( 5) =     ,76427000E 00
   PI( 6) =     ,70548000E 00
   C( 1) =     ,63258004E-01
   C( 2) =     ,11834616E 00
   C( 3) =     ,14238887E 00
   C( 4) =     ,15826172E 00
   C( 5) =     ,24112737E 00
   C( 6) =     ,27660788E 00
1  Q( 1) =     ,45002226E-02
1  Q( 2) =     ,22101597E 00
1  Q( 3) =     ,31684765E 00
3  Q( 4) =     ,43834619E 00
1  Q( 5) =     ,25835460E 00
1  Q( 6) =     ,21878614E 00

   NEW NI =      4
   P =     ,43840411E 00
   SUM NI*UNI =    ,33000000E 02
   SUM NI*VNI =    ,18000000E 02
   SUM NI*WNI =    ,23000000E 02
   PI( 1) =     ,99943000E 00
   PI( 2) =     ,94064000E 00
   PI( 3) =     ,88185000E 00
   PI( 4) =     ,98078215E 00
   PI( 5) =     ,76427000E 00
   PI( 6) =     ,70548000E 00
   C( 1) =     ,63258004E-01
   C( 2) =     ,11834616E 00
   C( 3) =     ,14238887E 00
   C( 4) =     ,15826172E 00
   C( 5) =     ,24112737E 00
   C( 6) =     ,27660788E 00
```

```
1  Q( 1) =     ,45002226E-02
1  Q( 2) =     ,22101597E 00
1  Q( 3) =     ,31684765E 00
4  Q( 4) =     ,96705820E-01
1  Q( 5) =     ,25835460E 00
1  Q( 6) =     ,21878614E 00

   NEW NI =      3
   P =     ,47796182E 00
   SUM NI*UNI =    ,39000000E 02
   SUM NI*VNI =    ,22000000E 02
   SUM NI*WNI =    ,27000000E 02
   PI( 1) =     ,99943000E 00
   PI( 2) =     ,94064000E 00
   PI( 3) =     ,96142034E 00
   PI( 4) =     ,98078215E 00
   PI( 5) =     ,76427000E 00
   PI( 6) =     ,70548000E 00
   C( 1) =     ,63258004E-01
   C( 2) =     ,11834616E 00
   C( 3) =     ,14238887E 00
   C( 4) =     ,15826172E 00
   C( 5) =     ,24112737E 00
   C( 6) =     ,27660788E 00
1  Q( 1) =     ,45002226E-02
1  Q( 2) =     ,22101597E 00
3  Q( 3) =     ,23789691E 00
4  Q( 4) =     ,96705820E-01
1  Q( 5) =     ,25835460E 00
1  Q( 6) =     ,21878614E 00
```

TABLE B. 2. 2. (Continued)

```
     NEW NI =       3
     P =     .53751238E 00
     SUM NI*UNI =   .49000000E 02
     SUM NI*VNI =   .28000000E 02
     SUM NI*WNI =   .35000000E 02
     PI( 1) =     .99943000E 00
     PI( 2) =     .94064000E 00
     PI( 3) =     .96142034E 00
     PI( 4) =     .98078215E 00
     PI( 5) =     .85949249E 00
     PI( 6) =     .70548000E 00
     C( 1) =    .63258004E-01
     C( 2) =    .11834616E 00
     C( 3) =    .14238887E 00
     C( 4) =    .15826172E 00
     C( 5) =    .24112737E 00
     C( 6) =    .27660788E 00
1    Q( 1) =    .45002226E-02
1    Q( 2) =    .22101597E 00
3    Q( 3) =    .23789691E 00
4    Q( 4) =    .96705820E-01
3    Q( 5) =    .46984649E 00
1    Q( 6) =    .21878614E 00
```

```
     NEW NI =       4
     P =     .61867911E 00
     SUM NI*UNI =   .57000000E 02
     SUM NI*VNI =   .33000000E 02
     SUM NI*WNI =   .41000000E 02
     PI( 1) =     .99943000E 00
     PI( 2) =     .94064000E 00
     PI( 3) =     .99398737E 00
     PI( 4) =     .98078215E 00
     PI( 5) =     .95686684E 00
     PI( 6) =     .70548000E 00
     C( 1) =    .63258004E-01
     C( 2) =    .11834616E 00
     C( 3) =    .14238887E 00
     C( 4) =    .15826172E 00
     C( 5) =    .24112737E 00
     C( 6) =    .27660788E 00
1    Q( 1) =    .45002226E-02
1    Q( 2) =    .22101597E 00
4    Q( 3) =    .36248806E-01
4    Q( 4) =    .96705820E-01
4    Q( 5) =    .13264783E 00
1    Q( 6) =    .21878614E 00
```

```
     NEW NI =       3
     P =     .72984365E 00
     SUM NI*UNI =   .73000000E 02
     SUM NI*VNI =   .41000000E 02
     SUM NI*WNI =   .57000000E 02
     PI( 1) =     .99943000E 00
     PI( 2) =     .98984749E 00
     PI( 3) =     .99398737E 00
     PI( 4) =     .98078215E 00
     PI( 5) =     .95686684E 00
     PI( 6) =     .79086843E 00
     C( 1) =    .63258004E-01
     C( 2) =    .11834616E 00
     C( 3) =    .14238887E 00
     C( 4) =    .15826172E 00
     C( 5) =    .24112737E 00
     C( 6) =    .27660788E 00
1    Q( 1) =    .45002226E-02
3    Q( 2) =    .79642382E-01
4    Q( 3) =    .36248806E-01
4    Q( 4) =    .96705820E-01
4    Q( 5) =    .13264783E 00
3    Q( 6) =    .59204056E 00
```

```
     NEW NI =       5
     P =     .89630038E 00
     SUM NI*UNI =   .85000000E 02
     SUM NI*VNI =   .47000000E 02
     SUM NI*WNI =   .67000000E 02
     PI( 1) =     .99943000E 00
     PI( 2) =     .98984749E 00
     PI( 3) =     .99398737E 00
     PI( 4) =     .98078215E 00
     PI( 5) =     .95686684E 00
     PI( 6) =     .97124319E 00
     C( 1) =    .63258004E-01
     C( 2) =    .11834616E 00
     C( 3) =    .14238887E 00
     C( 4) =    .15826172E 00
     C( 5) =    .24112737E 00
     C( 6) =    .27660788E 00
1    Q( 1) =    .45002226E-02
3    Q( 2) =    .79842382E-01
4    Q( 3) =    .36248806E-01
4    Q( 4) =    .96705820E-01
4    Q( 5) =    .13264783E 00
5    Q( 6) =    .69695711E-01
```

```
     NEW NI =       4
     P =     .59840870E 00
     SUM NI*UNI =   .54000000E 02
     SUM NI*VNI =   .31000000E 02
     SUM NI*WNI =   .39000000E 02
     PI( 1) =     .99943000E 00
     PI( 2) =     .94064000E 00
     PI( 3) =     .96142034E 00
     PI( 4) =     .98078215E 00
     PI( 5) =     .95686684E 00
     PI( 6) =     .70548000E 00
     C( 1) =    .63258004E-01
     C( 2) =    .11834616E 00
     C( 3) =    .14238887E 00
     C( 4) =    .15826172E 00
     C( 5) =    .24112737E 00
     C( 6) =    .27660788E 00
1    Q( 1) =    .45002226E-02
1    Q( 2) =    .22101597E 00
3    Q( 3) =    .23789691E 00
4    Q( 4) =    .96705820E-01
4    Q( 5) =    .13264783E 00
1    Q( 6) =    .21878614E 00
```

```
     NEW NI =       3
     P =     .65104394E 00
     SUM NI*UNI =   .61000000E 02
     SUM NI*VNI =   .35000000E 02
     SUM NI*WNI =   .47000000E 02
     PI( 1) =     .99943000E 00
     PI( 2) =     .98984749E 00
     PI( 3) =     .99398737E 00
     PI( 4) =     .98078215E 00
     PI( 5) =     .95686684E 00
     PI( 6) =     .70548000E 00
     C( 1) =    .63258004E-01
     C( 2) =    .11834616E 00
     C( 3) =    .14238887E 00
     C( 4) =    .15826172E 00
     C( 5) =    .24112737E 00
     C( 6) =    .27660788E 00
1    Q( 1) =    .45002226E-02
3    Q( 2) =    .79842382E-01
4    Q( 3) =    .36248806E-01
4    Q( 4) =    .96705820E-01
4    Q( 5) =    .13264783E 00
1    Q( 6) =    .21878614E 00
```

```
     NEW NI =       4
     P =     .84936510E 00
     SUM NI*UNI =   .79000000E 02
     SUM NI*VNI =   .44000000E 02
     SUM NI*WNI =   .62000000E 02
     PI( 1) =     .99943000E 00
     PI( 2) =     .98984749E 00
     PI( 3) =     .99398737E 00
     PI( 4) =     .98078215E 00
     PI( 5) =     .95686684E 00
     PI( 6) =     .92038349E 00
     C( 1) =    .63258004E-01
     C( 2) =    .11834616E 00
     C( 3) =    .14238887E 00
     C( 4) =    .15826172E 00
     C( 5) =    .24112737E 00
     C( 6) =    .27660788E 00
1    Q( 1) =    .45002226E-02
3    Q( 2) =    .79842382E-01
4    Q( 3) =    .36248806E-01
4    Q( 4) =    .96705820E-01
4    Q( 5) =    .13264783E 00
4    Q( 6) =    .19977467E 00
```

```
     NEW NI =       5
     P =     .92496857E 00
     SUM NI*UNI =   .90000000E 02
     SUM NI*VNI =   .50000000E 02
     SUM NI*WNI =   .71000000E 02
     PI( 1) =     .99943000E 00
     PI( 2) =     .98984749E 00
     PI( 3) =     .99398737E 00
     PI( 4) =     .98078215E 00
     PI( 5) =     .98747225E 00
     PI( 6) =     .97124319E 00
     C( 1) =    .63258004E-01
     C( 2) =    .11834616E 00
     C( 3) =    .14238887E 00
     C( 4) =    .15826172E 00
     C( 5) =    .24112737E 00
     C( 6) =    .27660788E 00
1    Q( 1) =    .45002226E-02
3    Q( 2) =    .79842382E-01
4    Q( 3) =    .36248806E-01
4    Q( 4) =    .96705820E-01
5    Q( 5) =    .37874912E-01
5    Q( 6) =    .69695711E-01
```

TABLE B. 2. 2. (Continued)

```
     NEW NI =       5
     P =    ,93912505E 00
     SUM NI*UNI =   ,94000000E 02
     SUM NI*VNI =   ,52000000E 02
     SUM NI*WNI =   ,73000000E 02
     PI( 1) =      ,99943000E 00
     PI( 2) =      ,98984749E 00
     PI( 3) =      ,99398737E 00
     PI( 4) =      ,99579285E 00
     PI( 5) =      ,98747225E 00
     PI( 6) =      ,97124319E 00
     C( 1) =      ,63258004E-01
     C( 2) =      ,11834616E 00
     C( 3) =      ,14238887E 00
     C( 4) =      ,15826172E 00
     C( 5) =      ,24112737E 00
     C( 6) =      ,27660788E 00
1    Q( 1) =      ,45002226E-02
3    Q( 2) =      ,79842382E-01
4    Q( 3) =      ,36248806E-01
5    Q( 4) =      ,21066491E-01
5    Q( 5) =      ,37874912E-01
5    Q( 6) =      ,69695711E-01
```

```
     NEW NI =       6
     P =    ,96627477E 00
     SUM NI*UNI =   ,10200000E 03
     SUM NI*VNI =   ,56000000E 02
     SUM NI*WNI =   ,81000000E 02
     PI( 1) =      ,99943000E 00
     PI( 2) =      ,99920060E 00
     PI( 3) =      ,99398737E 00
     PI( 4) =      ,99579285E 00
     PI( 5) =      ,98747225E 00
     PI( 6) =      ,98996718E 00
     C( 1) =      ,63258004E-01
     C( 2) =      ,11834616E 00
     C( 3) =      ,14238887E 00
     C( 4) =      ,15826172E 00
     C( 5) =      ,24112737E 00
     C( 6) =      ,27660788E 00
1    Q( 1) =      ,45002226E-02
4    Q( 2) =      ,62601064E-02
4    Q( 3) =      ,36248806E-01
5    Q( 4) =      ,21066491E-01
5    Q( 5) =      ,37874912E-01
6    Q( 6) =      ,24166251E-01
```

```
     NEW NI =       5
     P =    ,98013235E 00
     SUM NI*UNI =   ,11000000E 03
     SUM NI*VNI =   ,61000000E 02
     SUM NI*WNI =   ,87000000E 02
     PI( 1) =      ,99943000E 00
     PI( 2) =      ,99920060E 00
     PI( 3) =      ,99911777E 00
     PI( 4) =      ,99579285E 00
     PI( 5) =      ,99649052E 00
     PI( 6) =      ,98996718E 00
     C( 1) =      ,63258004E-01
     C( 2) =      ,11834616E 00
     C( 3) =      ,14238887E 00
     C( 4) =      ,15826172E 00
     C( 5) =      ,24112737E 00
     C( 6) =      ,27660788E 00
1    Q( 1) =      ,45002226E-02
4    Q( 2) =      ,62601064E-02
5    Q( 3) =      ,53260058E-02
5    Q( 4) =      ,21066491E-01
6    Q( 5) =      ,10616943E-01
6    Q( 6) =      ,24166251E-01
```

```
     NEW NI =       6
     P =    ,98997374E 00
     SUM NI*UNI =   ,12000000E 03
     SUM NI*VNI =   ,66000000E 02
     SUM NI*WNI =   ,94000000E 02
     PI( 1) =      ,99943000E 00
     PI( 2) =      ,99920060E 00
     PI( 3) =      ,99911777E 00
     PI( 4) =      ,99911284E 00
     PI( 5) =      ,99649052E 00
     PI( 6) =      ,99658469E 00
     C( 1) =      ,63258004E-01
     C( 2) =      ,11834616E 00
     C( 3) =      ,14238887E 00
     C( 4) =      ,15826172E 00
     C( 5) =      ,24112737E 00
     C( 6) =      ,27660788E 00
1    Q( 1) =      ,45002226E-02
4    Q( 2) =      ,62601064E-02
5    Q( 3) =      ,53260058E-02
6    Q( 4) =      ,44581423E-02
6    Q( 5) =      ,10616943E-01
7    Q( 6) =      ,82485468E-02
```

```
     NEW NI =       4
     P =    ,94799888E 00
     SUM NI*UNI =   ,96000000E 02
     SUM NI*VNI =   ,53000000E 02
     SUM NI*WNI =   ,76000000E 02
     PI( 1) =      ,99943000E 00
     PI( 2) =      ,99920060E 00
     PI( 3) =      ,99398737E 00
     PI( 4) =      ,99579285E 00
     PI( 5) =      ,98747225E 00
     PI( 6) =      ,97124319E 00
     C( 1) =      ,63258004E-01
     C( 2) =      ,11834616E 00
     C( 3) =      ,14238887E 00
     C( 4) =      ,15826172E 00
     C( 5) =      ,24112737E 00
     C( 6) =      ,27660788E 00
1    Q( 1) =      ,45002226E-02
4    Q( 2) =      ,62601064E-02
4    Q( 3) =      ,36248806E-01
5    Q( 4) =      ,21066491E-01
5    Q( 5) =      ,37874912E-01
5    Q( 6) =      ,69695711E-01
```

```
     NEW NI =       6
     P =    ,97509944E 00
     SUM NI*UNI =   ,10700000E 03
     SUM NI*VNI =   ,59000000E 02
     SUM NI*WNI =   ,85000000E 02
     PI( 1) =      ,99943000E 00
     PI( 2) =      ,99920060E 00
     PI( 3) =      ,99398737E 00
     PI( 4) =      ,99579285E 00
     PI( 5) =      ,99649052E 00
     PI( 6) =      ,98996718E 00
     C( 1) =      ,63258004E-01
     C( 2) =      ,11834616E 00
     C( 3) =      ,14238887E 00
     C( 4) =      ,15826172E 00
     C( 5) =      ,24112737E 00
     C( 6) =      ,27660788E 00
1    Q( 1) =      ,45002226E-02
4    Q( 2) =      ,62601064E-02
4    Q( 3) =      ,36248806E-01
5    Q( 4) =      ,21066491E-01
6    Q( 5) =      ,10616943E-01
6    Q( 6) =      ,24166251E-01
```

```
     NEW NI =       7
     P =    ,98668412E 00
     SUM NI*UNI =   ,11600000E 03
     SUM NI*VNI =   ,64000000E 02
     SUM NI*WNI =   ,92000000E 02
     PI( 1) =      ,99943000E 00
     PI( 2) =      ,99920060E 00
     PI( 3) =      ,99911777E 00
     PI( 4) =      ,99579285E 00
     PI( 5) =      ,99649052E 00
     PI( 6) =      ,99658469E 00
     C( 1) =      ,63258004E-01
     C( 2) =      ,11834616E 00
     C( 3) =      ,14238887E 00
     C( 4) =      ,15826172E 00
     C( 5) =      ,24112737E 00
     C( 6) =      ,27660788E 00
1    Q( 1) =      ,45002226E-02
4    Q( 2) =      ,62601064E-02
5    Q( 3) =      ,53260058E-02
5    Q( 4) =      ,21066491E-01
6    Q( 5) =      ,10616943E-01
7    Q( 6) =      ,82485468E-02
```

```
     NEW NI =       7
     P =    ,99250811E 00
     SUM NI*UNI =   ,12500000E 03
     SUM NI*VNI =   ,69000000E 02
     SUM NI*WNI =   ,98000000E 02
     PI( 1) =      ,99943000E 00
     PI( 2) =      ,99920060E 00
     PI( 3) =      ,99911777E 00
     PI( 4) =      ,99911284E 00
     PI( 5) =      ,99904157E 00
     PI( 6) =      ,99658469E 00
     C( 1) =      ,63258004E-01
     C( 2) =      ,11834616E 00
     C( 3) =      ,14238887E 00
     C( 4) =      ,15826172E 00
     C( 5) =      ,24112737E 00
     C( 6) =      ,27660788E 00
1    Q( 1) =      ,45002226E-02
4    Q( 2) =      ,62601064E-02
5    Q( 3) =      ,53260058E-02
6    Q( 4) =      ,44581423E-02
7    Q( 5) =      ,29123980E-02
7    Q( 6) =      ,82485468E-02
```

TABLE B.2.2. (Continued)

	NEW NI =	8
	P =	,99477263E 00
	SUM NI*UNI =	,13100000E 03
	SUM NI*VNI =	,72000000E 02
	SUM NI*WNI =	,10300000E 03
	PI(1) =	,99943000E 00
	PI(2) =	,99920060E 00
	PI(3) =	,99911777E 00
	PI(4) =	,99911284E 00
	PI(5) =	,99904157E 00
	PI(6) =	,99885852E 00
	C(1) =	,63258004E-01
	C(2) =	,11834616E 00
	C(3) =	,14238887E 00
	C(4) =	,15826172E 00
	C(5) =	,24112737E 00
	C(6) =	,27660788E 00
1	Q(1) =	,45002226E-02
4	Q(2) =	,62601064E-02
5	Q(3) =	,53260058E-02
6	Q(4) =	,44581423E-02
7	Q(5) =	,29123980E-02
8	Q(6) =	,27700934E-02

	NEW NI =	6
	P =	,99626457E 00
	SUM NI*UNI =	,13600000E 03
	SUM NI*VNI =	,75000000E 02
	SUM NI*WNI =	,10800000E 03
	PI(1) =	,99943000E 00
	PI(2) =	,99994087E 00
	PI(3) =	,99987546E 00
	PI(4) =	,99911284E 00
	PI(5) =	,99904157E 00
	PI(6) =	,99885852E 00
	C(1) =	,63258004E-01
	C(2) =	,11834616E 00
	C(3) =	,14238887E 00
	C(4) =	,15826172E 00
	C(5) =	,24112737E 00
	C(6) =	,27660788E 00
1	Q(1) =	,45002226E-02
5	Q(2) =	,46415602E-03
6	Q(3) =	,75454888E-03
6	Q(4) =	,44581423E-02
7	Q(5) =	,29123980E-02
8	Q(6) =	,27700934E-02

	NEW NI =	7
	P =	,99753511E 00
	SUM NI*UNI =	,14200000E 03
	SUM NI*VNI =	,79000000E 02
	SUM NI*WNI =	,11200000E 03
	PI(1) =	,99999903E 00
	PI(2) =	,99994087E 00
	PI(3) =	,99987546E 00
	PI(4) =	,99981777E 00
	PI(5) =	,99904157E 00
	PI(6) =	,99885852E 00
	C(1) =	,63258004E-01
	C(2) =	,11834616E 00
	C(3) =	,14238887E 00
	C(4) =	,15826172E 00
	C(5) =	,24112737E 00
	C(6) =	,27660788E 00
3	Q(1) =	,15390781E-04
5	Q(2) =	,46415602E-03
6	Q(3) =	,75454888E-03
7	Q(4) =	,91964546E-03
7	Q(5) =	,29123980E-02
8	Q(6) =	,27700934E-02

	NEW NI =	9
	P =	,99900052E 00
	SUM NI*UNI =	,15300000E 03
	SUM NI*VNI =	,85000000E 02
	SUM NI*WNI =	,12100000E 03
	PI(1) =	,99999903E 00
	PI(2) =	,99994087E 00
	PI(3) =	,99987546E 00
	PI(4) =	,99981777E 00
	PI(5) =	,99974316E 00
	PI(6) =	,99962387E 00
	C(1) =	,63258004E-01
	C(2) =	,11834616E 00
	C(3) =	,14238887E 00
	C(4) =	,15826172E 00
	C(5) =	,24112737E 00
	C(6) =	,27660788E 00
3	Q(1) =	,15390781E-04
5	Q(2) =	,46415602E-03
6	Q(3) =	,75454888E-03
7	Q(4) =	,91964546E-03
8	Q(5) =	,78406605E-03
9	Q(6) =	,91712618E-03

	NEW NI =	5
	P =	,99550961E 00
	SUM NI*UNI =	,13300000E 03
	SUM NI*VNI =	,73000000E 02
	SUM NI*WNI =	,10600000E 03
	PI(1) =	,99943000E 00
	PI(2) =	,99994087E 00
	PI(3) =	,99911777E 00
	PI(4) =	,99911284E 00
	PI(5) =	,99904157E 00
	PI(6) =	,99885852E 00
	C(1) =	,63258004E-01
	C(2) =	,11834616E 00
	C(3) =	,14238887E 00
	C(4) =	,15826172E 00
	C(5) =	,24112737E 00
	C(6) =	,27660788E 00
1	Q(1) =	,45002226E-02
5	Q(2) =	,46415602E-03
5	Q(3) =	,53260058E-02
6	Q(4) =	,44581423E-02
7	Q(5) =	,29123980E-02
8	Q(6) =	,27700934E-02

	NEW NI =	3
	P =	,99683180E 00
	SUM NI*UNI =	,13800000E 03
	SUM NI*VNI =	,77000000E 02
	SUM NI*WNI =	,11000000E 03
	PI(1) =	,99999903E 00
	PI(2) =	,99994087E 00
	PI(3) =	,99987546E 00
	PI(4) =	,99911284E 00
	PI(5) =	,99904157E 00
	PI(6) =	,99885852E 00
	C(1) =	,63258004E-01
	C(2) =	,11834616E 00
	C(3) =	,14238887E 00
	C(4) =	,15826172E 00
	C(5) =	,24112737E 00
	C(6) =	,27660788E 00
3	Q(1) =	,15390781E-04
5	Q(2) =	,46415602E-03
6	Q(3) =	,75454888E-03
6	Q(4) =	,44581423E-02
7	Q(5) =	,29123980E-02
8	Q(6) =	,27700934E-02

	NEW NI =	8
	P =	,99823564E 00
	SUM NI*UNI =	,14700000E 03
	SUM NI*VNI =	,82000000E 02
	SUM NI*WNI =	,11600000E 03
	PI(1) =	,99999903E 00
	PI(2) =	,99994087E 00
	PI(3) =	,99987546E 00
	PI(4) =	,99981777E 00
	PI(5) =	,99974316E 00
	PI(6) =	,99885852E 00
	C(1) =	,63258004E-01
	C(2) =	,11834616E 00
	C(3) =	,14238887E 00
	C(4) =	,15826172E 00
	C(5) =	,24112737E 00
	C(6) =	,27660788E 00
3	Q(1) =	,15390781E-04
5	Q(2) =	,46415602E-03
6	Q(3) =	,75454888E-03
7	Q(4) =	,91964546E-03
8	Q(5) =	,78406605E-03
8	Q(6) =	,27700934E-02

	NEW NI =	8
	P =	,99914592E 00
	SUM NI*UNI =	,15700000E 03
	SUM NI*VNI =	,87000000E 02
	SUM NI*WNI =	,12300000E 03
	PI(1) =	,99999903E 00
	PI(2) =	,99994087E 00
	PI(3) =	,99987546E 00
	PI(4) =	,99996329E 00
	PI(5) =	,99974316E 00
	PI(6) =	,99962387E 00
	C(1) =	,63258004E-01
	C(2) =	,11834616E 00
	C(3) =	,14238887E 00
	C(4) =	,15826172E 00
	C(5) =	,24112737E 00
	C(6) =	,27660788E 00
3	Q(1) =	,15390781E-04
5	Q(2) =	,46415602E-03
6	Q(3) =	,75454888E-03
8	Q(4) =	,18594101E-03
8	Q(5) =	,78406605E-03
9	Q(6) =	,91712618E-03

TABLE B.2.2. (Concluded)

```
NEW NI =     10
P =     ,99939939E 00
SUM NI*UNI =   ,16300000E 03
SUM NI*VNI =   ,90000000E 02
SUM NI*WNI =   ,12800000E 03
PI( 1) =    ,99999903E 00
PI( 2) =    ,99994087E 00
PI( 3) =    ,99987546E 00
PI( 4) =    ,99996329E 00
PI( 5) =    ,99974316E 00
PI( 6) =    ,99987746E 00
C( 1) =    ,63258004E-01
C( 2) =    ,11834616E 00
C( 3) =    ,14238887E 00
C( 4) =    ,15826172E 00
C( 5) =    ,24112737E 00
C( 6) =    ,27660788E 00
```

```
 3  Q( 1) =    ,15390781E-04
 5  Q( 2) =    ,46415602E-03
 6  Q( 3) =    ,75454888E-03
 8  Q( 4) =    ,18594101E-03
 8  Q( 5) =    ,78406605E-03
10  Q( 6) =    ,30004833E-03

    NEW NI =      9
    P =     ,99958833E 00
    SUM NI*UNI =   ,16800000E 03
    SUM NI*VNI =   ,93000000E 02
    SUM NI*WNI =   ,13200000E 03
```

```
    PI( 1) =    ,99999903E 00
    PI( 2) =    ,99994087E 00
    PI( 3) =    ,99987546E 00
    PI( 4) =    ,99996329E 00
    PI( 5) =    ,99993217E 00
    PI( 6) =    ,99987746E 00
    C( 1) =    ,63258004E-01
    C( 2) =    ,11834616E 00
    C( 3) =    ,14238887E 00
    C( 4) =    ,15826172E 00
    C( 5) =    ,24112737E 00
    C( 6) =    ,27660788E 00
 3  Q( 1) =    ,15390781E-04
 5  Q( 2) =    ,46415602E-03
 6  Q( 3) =    ,75454888E-03
 8  Q( 4) =    ,18594101E-03
 9  Q( 5) =    ,20789207E-03
10  Q( 6) =    ,30004833E-03

CØNATRAINT LIMIT REACHED
```

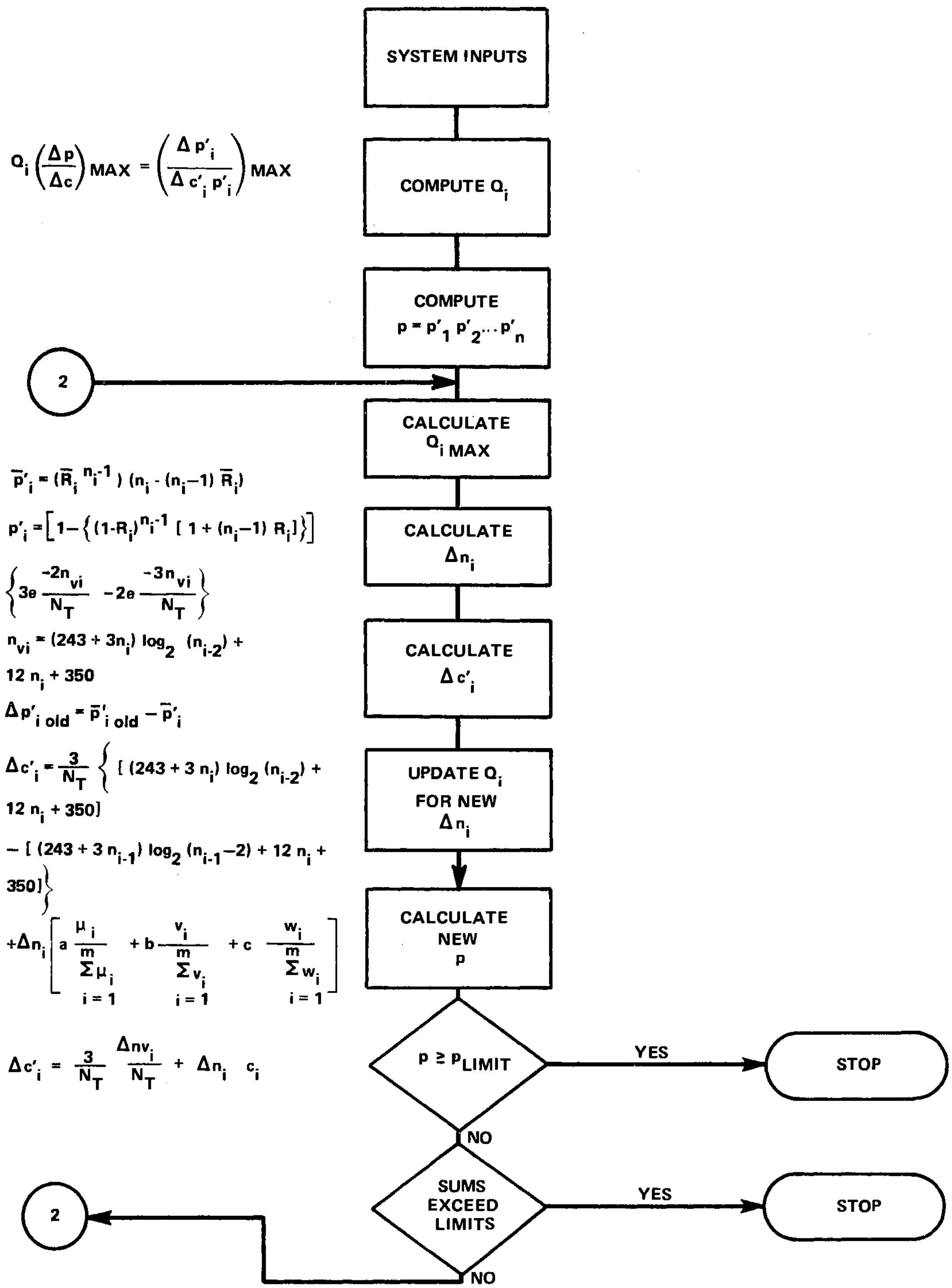


Figure B.3.1. Logic Diagram of Computer Program for Optimization Process Utilizing

$$\left(\frac{\Delta P}{\Delta C}\right)_{max} = \left[\frac{\Delta p_i}{\left(\Delta n_i \, c_i + \frac{3\Delta n_{v_i}}{N_T}\right) p_i}\right]_{max}$$

TABLE B.3.1

OPTIMIZATION OF SYSTEM WITH DECISION ELEMENTS AND C_r (constraint) = 4.78

```
INPUT DATA

N =       6
ICØN =      0
PLIMIT VALUE =          .99950000
A =       .33333000     B =       .33333000     C =       .33333000
NT  =  50000.00000000
U( 1) =        1.00000000
U( 2) =        2.00000000
U( 3) =        3.00000000
U( 4) =        4.00000000
U( 5) =        5.00000000
U( 6) =        6.00000000
V( 1) =        1.00000000
V( 2) =        1.00000000
V( 3) =        2.00000000
V( 4) =        2.00000000
V( 5) =        3.00000000
V( 6) =        3.00000000
W( 1) =        1.00000000
W( 2) =        3.00000000
W( 3) =        2.00000000
W( 4) =        2.00000000
W( 5) =        4.00000000
W( 6) =        5.00000000
RBAR( 1) =          .00057000
RBAR( 2) =          .05936000
RBAR( 3) =          .11815000
RBAR( 4) =          .17694000
RBAR( 5) =          .23573000
RBAR( 6) =          .29452000
CU =      99.00000000     CV =      57.00000000     CW =      83.00000000

NI =       1
CØNSTRAINT R =          4.78000000
SUM NI*UNI =   .21000000E 02
SUM NI*VNI =   .12000000E 02
SUM NI*WNI =   .17000000E 02
Q = DELTA PI'/DELTA    CI'*PI'
```

```
NEW NI =         1
P =     .36790319E 00
SUM NI*UNI =   .21000000E 02
SUM NI*VNI =   .12000000E 02
SUM NI*WNI =   .17000000E 02
PI( 1) =     .99943000E 00
PI( 2) =     .94064000E 00
PI( 3) =     .88185000E 00
PI( 4) =     .82306000E 00
PI( 5) =     .76427000E 00
PI( 6) =     .70548000E 00
DELTA C( 1) =     .14967601E 00
DELTA C( 2) =     .25985231E 00
DELTA C( 3) =     .30793773E 00
DELTA C( 4) =     .33968345E 00
DELTA C( 5) =     .50541475E 00
DELTA C( 6) =     .57637576E 00
CR TØTAL =        1.00000000
1  Q( 1) =     .38038842E-02
1  Q( 2) =     .20131736E 00
1  Q( 3) =     .29301754E 00
1  Q( 4) =     .33656180E 00
1  Q( 5) =     .24651583E 00
1  Q( 6) =     .20999485E 00

NEW NI =         3
P =     .40989117E 00
SUM NI*UNI =   .29000000E 02
SUM NI*VNI =   .16000000E 02
SUM NI*WNI =   .21000000E 02
PI( 1) =     .99943000E 00
PI( 2) =     .94064000E 00
PI( 3) =     .88185000E 00
PI( 4) =     .91699401E 00
PI( 5) =     .76427000E 00
PI( 6) =     .70548000E 00
DELTA C( 1) =     .14967601E 00
DELTA C( 2) =     .25985231E 00
DELTA C( 3) =     .30793773E 00
DELTA C( 4) =     .17428172E 00
DELTA C( 5) =     .50541475E 00
```

```
1  Q( 1) =     .38038842E-02
1  Q( 2) =     .20131736E 00
1  Q( 3) =     .29301754E 00
4  Q( 4) =     .89277134E-01
1  Q( 5) =     .24651583E 00
1  Q( 6) =     .20999485E 00

DELTA C( 6) =     .57637576E 00
CR TØTAL =        1.33968345
1  Q( 1) =     .38038842E-02
1  Q( 2) =     .20131736E 00
1  Q( 3) =     .29301754E 00
3  Q( 4) =     .39606388E 00
1  Q( 5) =     .24651583E 00
1  Q( 6) =     .20999485E 00

NEW NI =         4
P =     .43818460E 00
SUM NI*UNI =   .33000000E 02
SUM NI*VNI =   .18000000E 02
SUM NI*WNI =   .23000000E 02
PI( 1) =     .99943000E 00
PI( 2) =     .94064000E 00
PI( 3) =     .88185000E 00
PI( 4) =     .98029108E 00
PI( 5) =     .76427000E 00
PI( 6) =     .70548000E 00
DELTA C( 1) =     .14967601E 00
DELTA C( 2) =     .25985231E 00
DELTA C( 3) =     .30793773E 00
DELTA C( 4) =     .16821694E 00
DELTA C( 5) =     .50541475E 00
DELTA C( 6) =     .57637576E 00
CR TØTAL =        1.51396517
```

TABLE B.3.1. (Continued)

```
NEW NI =      3
P =   .47763818E 00
SUM NI*UNI =  .39000000E 02
SUM NI*VNI =  .22000000E 02
SUM NI*WNI =  .27000000E 02
PI( 1) =   .99943000E 00
PI( 2) =   .94064000E 00
PI( 3) =   .96125064E 00
PI( 4) =   .98029108E 00
PI( 5) =   .76427000E 00
PI( 6) =   .70548000E 00
DELTA C( 1) =   .14967601E 00
DELTA C( 2) =   .25985231E 00
DELTA C( 3) =   .15840887E 00
DELTA C( 4) =   .16821694E 00
DELTA C( 5) =   .50541475E 00
DELTA C( 6) =   .57637576E 00
CR TØTAL =   1.82190290
1  Q( 1) =   .38038842E-02
1  Q( 2) =   .20131736E 00
3  Q( 3) =   .21172207E 00
4  Q( 4) =   .89277134E-01
1  Q( 5) =   .24651583E 00
1  Q( 6) =   .20999485E 00

NEW NI =      3
P =   .53705362E 00
SUM NI*UNI =  .49000000E 02
SUM NI*VNI =  .28000000E 02
SUM NI*WNI =  .35000000E 02
PI( 1) =   .99943000E 00
PI( 2) =   .94064000E 00
PI( 3) =   .96125064E 00
PI( 4) =   .98029108E 00
PI( 5) =   .85934078E 00
PI( 6) =   .70548000E 00
DELTA C( 1) =   .14967601E 00
DELTA C( 2) =   .25985231E 00
DELTA C( 3) =   .15840887E 00
DELTA C( 4) =   .16821694E 00
DELTA C( 5) =   .25714737E 00
DELTA C( 6) =   .57637576E 00
CR TØTAL =   2.32731765
1  Q( 1) =   .38038842E-02
1  Q( 2) =   .20131736E 00
3  Q( 3) =   .21172207E 00
4  Q( 4) =   .89277134E-01
3  Q( 5) =   .43917183E 00
1  Q( 6) =   .20999485E 00

NEW NI =      4
P =   .59770409E 00
SUM NI*UNI =  .54000000E 02
SUM NI*VNI =  .31000000E 02
SUM NI*WNI =  .39000000E 02
PI( 1) =   .99943000E 00
PI( 2) =   .94064000E 00
PI( 3) =   .96125064E 00
PI( 4) =   .98029108E 00
PI( 5) =   .95638775E 00
PI( 6) =   .70548000E 00
DELTA C( 1) =   .14967601E 00
DELTA C( 2) =   .25985231E 00
DELTA C( 3) =   .15840887E 00
DELTA C( 4) =   .16821694E 00
DELTA C( 5) =   .25108259E 00
DELTA C( 6) =   .57637576E 00
CR TØTAL =   2.58446502
1  Q( 1) =   .38038842E-02
1  Q( 2) =   .20131736E 00
3  Q( 3) =   .21172207E 00
4  Q( 4) =   .89277134E-01
4  Q( 5) =   .12622702E 00
1  Q( 6) =   .20999485E 00

NEW NI =      4
P =   .61775028E 00
SUM NI*UNI =  .57000000E 02
SUM NI*VNI =  .33000000E 02
SUM NI*WNI =  .41000000E 02
PI( 1) =   .99943000E 00
PI( 2) =   .94064000E 00
PI( 3) =   .99348969E 00
PI( 4) =   .98029108E 00
PI( 5) =   .95638775E 00
PI( 6) =   .70548000E 00
DELTA C( 1) =   .14967601E 00
DELTA C( 2) =   .25985231E 00
DELTA C( 3) =   .15234409E 00
DELTA C( 4) =   .16821694E 00
DELTA C( 5) =   .25108259E 00
DELTA C( 6) =   .57637576E 00
CR TØTAL =   2.74287389
1  Q( 1) =   .38038842E-02
1  Q( 2) =   .20131736E 00
4  Q( 3) =   .32015615E-01
4  Q( 4) =   .89277134E-01
4  Q( 5) =   .12622702E 00
1  Q( 6) =   .20999485E 00

NEW NI =      3
P =   .69239803E 00
SUM NI*UNI =  .69000000E 02
SUM NI*VNI =  .39000000E 02
SUM NI*WNI =  .51000000E 02
PI( 1) =   .99943000E 00
PI( 2) =   .94064000E 00
PI( 3) =   .99348969E 00
PI( 4) =   .98029108E 00
PI( 5) =   .95638775E 00
PI( 6) =   .79072884E 00
DELTA C( 1) =   .14967601E 00
DELTA C( 2) =   .25985231E 00
DELTA C( 3) =   .15234409E 00
DELTA C( 4) =   .16821694E 00
DELTA C( 5) =   .25108259E 00
DELTA C( 6) =   .29262788E 00
CR TØTAL =   3.31924964
1  Q( 1) =   .38038842E-02
1  Q( 2) =   .20131736E 00
4  Q( 3) =   .32015615E-01
4  Q( 4) =   .89277134E-01
4  Q( 5) =   .12622702E 00
3  Q( 6) =   .55833966E 00

NEW NI =      4
P =   .80552600E 00
SUM NI*UNI =  .75000000E 02
SUM NI*VNI =  .42000000E 02
SUM NI*WNI =  .56000000E 02
PI( 1) =   .99943000E 00
PI( 2) =   .94064000E 00
PI( 3) =   .99348969E 00
PI( 4) =   .98029108E 00
PI( 5) =   .95638775E 00
PI( 6) =   .91992266E 00
DELTA C( 1) =   .14967601E 00
DELTA C( 2) =   .25985231E 00
DELTA C( 3) =   .15234409E 00
DELTA C( 4) =   .16821694E 00
DELTA C( 5) =   .25108259E 00
DELTA C( 6) =   .28656310E 00
CR TØTAL =   3.61187752
1  Q( 1) =   .38038842E-02
1  Q( 2) =   .20131736E 00
4  Q( 3) =   .32015615E-01
4  Q( 4) =   .89277134E-01
4  Q( 5) =   .12622702E 00
4  Q( 6) =   .19179390E 00

NEW NI =      3
P =   .84751568E 00
SUM NI*UNI =  .79000000E 02
SUM NI*VNI =  .44000000E 02
SUM NI*WNI =  .62000000E 02
PI( 1) =   .99943000E 00
PI( 2) =   .98967277E 00
PI( 3) =   .99348969E 00
PI( 4) =   .98029108E 00
PI( 5) =   .95638775E 00
PI( 6) =   .91992266E 00
DELTA C( 1) =   .14967601E 00
DELTA C( 2) =   .13436616E 00
DELTA C( 3) =   .15234409E 00
DELTA C( 4) =   .16821694E 00
DELTA C( 5) =   .25108259E 00
DELTA C( 6) =   .28656310E 00
CR TØTAL =   3.87172983
1  Q( 1) =   .38038842E-02
3  Q( 2) =   .67887181E-01
4  Q( 3) =   .32015615E-01
4  Q( 4) =   .89277134E-01
4  Q( 5) =   .12622702E 00
4  Q( 6) =   .19179390E 00

NEW NI =      5
P =   .89409604E 00
SUM NI*UNI =  .85000000E 02
SUM NI*VNI =  .47000000E 02
SUM NI*WNI =  .67000000E 02
PI( 1) =   .99943000E 00
PI( 2) =   .98967277E 00
PI( 3) =   .99348969E 00
PI( 4) =   .98029108E 00
PI( 5) =   .95638775E 00
PI( 6) =   .97048258E 00
```

TABLE B.3.1. (Concluded)

```
   DELTA C( 1) =    .14967601E 00
   DELTA C( 2) =    .13436616E 00
   DELTA C( 3) =    .15234409E 00
   DELTA C( 4) =    .16821694E 00
   DELTA C( 5) =    .25108259E 00
   DELTA C( 6) =    .28411266E 00
   CR TØTAL =       4.15829293
1  Q( 1) =    .38038842E-02
3  Q( 2) =    .67887181E-01
4  Q( 3) =    .32015615E-01
4  Q( 4) =    .89277134E-01
4  Q( 5) =    .12622702E 00
5  Q( 6) =    .66945685E-01


   NEW NI =      5
   P =   .92243299E 00
   SUM NI*UNI =   .90000000E 02
   SUM NI*VNI =   .50000000E 02
   SUM NI*WNI =   .71000000E 02
   PI( 1) =    .99943000E 00
   PI( 2) =    .98967277E 00
   PI( 3) =    .99348969E 00
   PI( 4) =    .98029108E 00
   PI( 5) =    .98669894E 00
   PI( 6) =    .97048258E 00
   DELTA C( 1) =    .14967601E 00
   DELTA C( 2) =    .13436616E 00
   DELTA C( 3) =    .15234409E 00
   DELTA C( 4) =    .16821694E 00
   DELTA C( 5) =    .24863215E 00
   DELTA C( 6) =    .28411266E 00
   CR TØTAL =       4.40937552
1  Q( 1) =    .38038842E-02
3  Q( 2) =    .67887181E-01
4  Q( 3) =    .32015615E-01
4  Q( 4) =    .89277134E-01
5  Q( 5) =    .35703277E-01
5  Q( 6) =    .66945685E-01
```

```
   NEW NI =      5
   P =   .93628602E 00
   SUM NI*UNI =   .94000000E 02
   SUM NI*VNI =   .52000000E 02
   SUM NI*WNI =   .73000000E 02
   PI( 1) =    .99943000E 00
   PI( 2) =    .98967277E 00
   PI( 3) =    .99348969E 00
   PI( 4) =    .99501302E 00
   PI( 5) =    .98669894E 00
   PI( 6) =    .97048258E 00
   DELTA C( 1) =    .14967601E 00
   DELTA C( 2) =    .13436616E 00
   DELTA C( 3) =    .15234409E 00
   DELTA C( 4) =    .16576650E 00
   DELTA C( 5) =    .24863215E 00
   DELTA C( 6) =    .28411266E 00
   CR TØTAL =       4.57759246
1  Q( 1) =    .38038842E-02
3  Q( 2) =    .67887181E-01
4  Q( 3) =    .32015615E-01
5  Q( 4) =    .18579103E-01
5  Q( 5) =    .35703277E-01
5  Q( 6) =    .66945685E-01


   NEW NI =      4
   P =   .94482657E 00
   SUM NI*UNI =   .96000000E 02
   SUM NI*VNI =   .53000000E 02
   SUM NI*WNI =   .76000000E 02
   PI( 1) =    .99943000E 00
   PI( 2) =    .99870031E 00
   PI( 3) =    .99348969E 00
   PI( 4) =    .99501302E 00
   PI( 5) =    .98669894E 00
   PI( 6) =    .97048258E 00
   DELTA C( 1) =    .14967601E 00
```

```
   DELTA C( 2) =    .12830137E 00
   DELTA C( 3) =    .15234409E 00
   DELTA C( 4) =    .16576650E 00
   DELTA C( 5) =    .24863215E 00
   DELTA C( 6) =    .28411266E 00
   CR TØTAL =       4.71195862
1  Q( 1) =    .38038842E-02
4  Q( 2) =    .35702802E-02
4  Q( 3) =    .32015615E-01
5  Q( 4) =    .18579103E-01
5  Q( 5) =    .35703277E-01
5  Q( 6) =    .66945685E-01


DELTA C CØNSTRAINT REACHED, ELIMINATE MØDULE( 6)
      NEGATIVE Q FØUND, ELIMINATE MØDULE ( 6)
DELTA C CØNSTRAINT REACHED, ELIMINATE MØDULE( 5)
      NEGATIVE Q FØUND, ELIMINATE MØDULE ( 5)
DELTA C CØNSTRAINT REACHED, ELIMINATE MØDULE( 3)
      NEGATIVE Q FØUND, ELIMINATE MØDULE ( 3)
DELTA C CØNSTRAINT REACHED, ELIMINATE MØDULE( 4)
      NEGATIVE Q FØUND, ELIMINATE MØDULE ( 4)
      NEGATIVE Q FØUND, ELIMINATE MØDULE ( 1)
      NEGATIVE Q FØUND, ELIMINATE MØDULE ( 2)

 ALL MØDULES ELIMINATED
```

TABLE B.3.2

OPTIMIZATION OF SYSTEM WITH DECISION ELEMENTS TO OBTAIN MAXIMUM RELIABILITY WITH MINIMUM EXPENDITURE OF RESOURCES

```
INPUT DATA

N =        6
ICØN =     1
PLIMIT VALUE =      ,99950000
A =       ,33333000    B =       ,33333000    C =       ,33333000
NT  =  50000,00000000
U( 1) =     1,00000000
U( 2) =     2,00000000
U( 3) =     3,00000000
U( 4) =     4,00000000
U( 5) =     5,00000000
U( 6) =     6,00000000
V( 1) =     1,00000000
V( 2) =     1,00000000
V( 3) =     2,00000000
V( 4) =     2,00000000
V( 5) =     3,00000000
V( 6) =     3,00000000
W( 1) =     1,00000000
W( 2) =     3,00000000
W( 3) =     2,00000000
W( 4) =     2,00000000
W( 5) =     4,00000000
W( 6) =     5,00000000
RBAR( 1) =      ,00057000
RBAR( 2) =      ,05936000
RBAR( 3) =      .11815000
RBAR( 4) =      .17694000
RBAR( 5) =      ,23573000
RBAR( 6) =      .29452000
CU =     99,00000000     CV =     57,00000000     CW =     83,00000000

NI =       1
CONSTRAINT R =        4,73000000
SUM NI*UNI =  ,21000000E 02
SUM NI*VNI =  ,12000000E 02
SUM NI*WNI =  ,17000000E 02
Q = DELTA PI'/DELTA    CI'*PI'
```

```
NEW NI =       1
P =     ,35790319E 00
SUM NI*UNI =  ,21000000E 02
SUM NI*VNI =  ,12000000E 02
SUM NI*WNI =  ,17000000E 02
PI( 1) =    ,99943000E 00
PI( 2) =    ,94054000E 00
PI( 3) =    ,88145000E 00
PI( 4) =    ,82306000E 00
PI( 5) =    ,76427000E 00
PI( 6) =    ,70548000E 00
DELTA C( 1) =    ,14967601E 00
DELTA C( 2) =    ,25985231E 00
DELTA C( 3) =    ,30793773E 00
DELTA C( 4) =    ,33968345E 00
DELTA C( 5) =    ,50541475E 00
DELTA C( 6) =    ,57637576E 00
CR TØTAL =       1,00000000
1  Q( 1) =    ,38038842E-02
1  Q( 2) =    ,20131736E 00
1  Q( 3) =    ,29301754E 00
1  Q( 4) =    ,33656180E 00
1  Q( 5) =    ,24651583E 00
1  Q( 6) =    ,20999485E 00

NEW NI =       3
P =     ,40989117E 00
SUM NI*UNI =  ,29000000E 02
SUM NI*VNI =  ,16000000E 02
SUM NI*WNI =  ,21000000E 02
PI( 1) =    ,99943000E 00
PI( 2) =    ,94064000E 00
PI( 3) =    ,88185000E 00
PI( 4) =    ,91699401E 00
PI( 5) =    ,76427000E 00
PI( 6) =    ,70548000E 00
DELTA C( 1) =    ,14967601E 00
DELTA C( 2) =    ,25985231E 00
DELTA C( 3) =    ,30793773E 00
DELTA C( 4) =    ,17428172E 00
```

```
DELTA C( 5) =    ,50541475E 00
DELTA C( 6) =    ,57637576E 00
CR TØTAL =       1,33968345
1  Q( 1) =    ,38038842E-02
1  Q( 2) =    ,20131736E 00
1  Q( 3) =    ,29301754E 00
3  Q( 4) =    ,39606388E 00
1  Q( 5) =    ,24651583E 00
1  Q( 6) =    ,20999485E 00

NEW NI =       4
P =     ,43818460E 00
SUM NI*UNI =  ,33000000E 02
SUM NI*VNI =  ,18000000E 02
SUM NI*WNI =  ,23000000E 02
PI( 1) =    ,99943000E 00
PI( 2) =    ,94064000E 00
PI( 3) =    ,88185000E 00
PI( 4) =    ,98029108E 00
PI( 5) =    ,76427000E 00
PI( 6) =    ,70548000E 00
DELTA C( 1) =    ,14967601E 00
DELTA C( 2) =    ,25985231E 00
DELTA C( 3) =    ,30793773E 00
DELTA C( 4) =    ,16821694E 00
DELTA C( 5) =    ,50541475E 00
```

TABLE B.3.2. (Continued)

```
    DELTA C( 6) =      ,57637576E 00
    CR TOTAL =         1,51396517
1   Q( 1) =      ,38038842E-02
1   Q( 2) =      ,20131736E 00
1   Q( 3) =      ,29301754E 00
4   Q( 4) =      ,69277134E-01
1   Q( 5) =      ,24651583E 00
1   Q( 6) =      ,20999485E 00


    NEW NI =        3
    P =     ,47763618E 00
    SUM NI*UNI =   ,39000000E 02
    SUM NI*VNI =   ,22000000E 02
    SUM NI*WNI =   ,27000000E 02
    PI( 1) =     ,99943000E 00
    PI( 2) =     ,94064000E 00
    PI( 3) =     ,96125064E 00
    PI( 4) =     ,98029108E 00
    PI( 5) =     ,76427000E 00
    PI( 6) =     ,70548000E 00
    DELTA C( 1) =      ,14967601E 00
    DELTA C( 2) =      ,25985231E 00
    DELTA C( 3) =      ,15840887E 00
    DELTA C( 4) =      ,16821694E 00
    DELTA C( 5) =      ,50541475E 00
    DELTA C( 6) =      ,57637576E 00
    CR TOTAL =         1,82190290
1   Q( 1) =      ,38038842E-02
1   Q( 2) =      ,20131736E 00
3   Q( 3) =      ,21172207E 00
4   Q( 4) =      ,69277134E-01
1   Q( 5) =      ,24651583E 00
1   Q( 6) =      ,20999485E 00


    NEW NI =        3
    P =     ,53705362E 00
    SUM NI*UNI =   ,49000000E 02
    SUM NI*VNI =   ,26000000E 02
    SUM NI*WNI =   ,35000000E 02
    PI( 1) =     ,99943000E 00
    PI( 2) =     ,94064000E 00
    PI( 3) =     ,96125064E 00
    PI( 4) =     ,98029108E 00
    PI( 5) =     ,85934078E 00
    PI( 6) =     ,70548000E 00
    DELTA C( 1) =      ,14967601E 00
    DELTA C( 2) =      ,25985231E 00
    DELTA C( 3) =      ,15840887E 00
    DELTA C( 4) =      ,16821694E 00
    DELTA C( 5) =      ,25714737E 00
    DELTA C( 6) =      ,57637576E 00
    CR TOTAL =         2,32731765
1   Q( 1) =      ,38038842E-02
1   Q( 2) =      ,20131736E 00
3   Q( 3) =      ,21172207E 00
4   Q( 4) =      ,69277134E-01
3   Q( 5) =      ,63917183E 00
1   Q( 6) =      ,20999485E 00


    NEW NI =        4
    P =     ,59770409E 00
    SUM NI*UNI =   ,54000000E 02
    SUM NI*VNI =   ,31000000E 02
    SUM NI*WNI =   ,39000000E 02
    PI( 1) =     ,99943000E 00
    PI( 2) =     ,94064000E 00
    PI( 3) =     ,96125064E 00
    PI( 4) =     ,98029108E 00
    PI( 5) =     ,95638775E 00
    PI( 6) =     ,70548000E 00
    DELTA C( 1) =      ,14967601E 00
    DELTA C( 2) =      ,25985231E 00
    DELTA C( 3) =      ,15840887E 00
    DELTA C( 4) =      ,16821694E 00
    DELTA C( 5) =      ,25108259E 00
    DELTA C( 6) =      ,57637576E 00
    CR TOTAL =         2,58446502
1   Q( 1) =      ,38038842E-02
1   Q( 2) =      ,20131736E 00
3   Q( 3) =      ,21172207E 00
4   Q( 4) =      ,69277134E-01
4   Q( 5) =      ,12622702E 00
1   Q( 6) =      ,20999485E 00


    NEW NI =        4
    P =     ,61775028E 00
    SUM NI*UNI =   ,57000000E 02
    SUM NI*VNI =   ,33000000E 02
    SUM NI*WNI =   ,41000000E 02
    PI( 1) =     ,99943000E 00
    PI( 2) =     ,94064000E 00
    PI( 3) =     ,99348969E 00
    PI( 4) =     ,98029108E 00
    PI( 5) =     ,95638775E 00
    PI( 6) =     ,70548000E 00
    DELTA C( 1) =      ,14967601E 00
    DELTA C( 2) =      ,25985231E 00
    DELTA C( 3) =      ,15234409E 00
    DELTA C( 4) =      ,16821694E 00
    DELTA C( 5) =      ,25108259E 00
    DELTA C( 6) =      ,57637576E 00
    CR TOTAL =         2,74287389
1   Q( 1) =      ,38038842E-02
1   Q( 2) =      ,20131736E 00
4   Q( 3) =      ,32015615E-01
4   Q( 4) =      ,89277134E-01
4   Q( 5) =      ,12622702E 00
1   Q( 6) =      ,20999485E 00


    NEW NI =        5
    P =     ,69239603E 00
    SUM NI*UNI =   ,59000000E 02
    SUM NI*VNI =   ,39000000E 02
    SUM NI*WNI =   ,51000000E 02
    PI( 1) =     ,99943000E 00
    PI( 2) =     ,94064000E 00
    PI( 3) =     ,99348969E 00
    PI( 4) =     ,98029108E 00
    PI( 5) =     ,95638775E 00
    PI( 6) =     ,79072884E 00
    DELTA C( 1) =      ,14967601E 00
    DELTA C( 2) =      ,25985231E 00
    DELTA C( 3) =      ,15234409E 00
    DELTA C( 4) =      ,16821694E 00
    DELTA C( 5) =      ,25108259E 00
    DELTA C( 6) =      ,29262768E 00
    CR TOTAL =         3,31921964
1   Q( 1) =      ,38038842E-02
1   Q( 2) =      ,20131736E 00
4   Q( 3) =      ,32015615E-01
4   Q( 4) =      ,89277134E-01
4   Q( 5) =      ,12622702E 00
3   Q( 6) =      ,55833956E 00


    NEW NI =        4
    P =     ,46552600E 00
    SUM NI*UNI =   ,75000000E 02
    SUM NI*VNI =   ,42000000E 02
    SUM NI*WNI =   ,56000000E 02
    PI( 1) =     ,99943000E 00
    PI( 2) =     ,94064000E 00
    PI( 3) =     ,99348969E 00
    PI( 4) =     ,98029108E 00
    PI( 5) =     ,95638775E 00
    PI( 6) =     ,91992266E 00
    DELTA C( 1) =      ,14967601E 00
    DELTA C( 2) =      ,25985231E 00
    DELTA C( 3) =      ,15234409E 00
    DELTA C( 4) =      ,16821694E 00
    DELTA C( 5) =      ,25108259E 00
    DELTA C( 6) =      ,24656310E 00
    CR TOTAL =         3,61187752
1   Q( 1) =      ,38038842E-02
1   Q( 2) =      ,20131736E 00
4   Q( 3) =      ,32015615E-01
4   Q( 4) =      ,69277134E-01
4   Q( 5) =      ,12622702E 00
4   Q( 6) =      ,19179390E 00


    NEW NI =        5
    P =     ,84751566E 00
    SUM NI*UNI =   ,79000000E 02
    SUM NI*VNI =   ,44000000E 02
    SUM NI*WNI =   ,62000000E 02
    PI( 1) =     ,99943000E 00
    PI( 2) =     ,96367277E 00
    PI( 3) =     ,99348969E 00
    PI( 4) =     ,98029108E 00
```

TABLE B.3.2. (Continued)

```
    PI( 5) =     ,95638775E 00
    PI( 6) =     ,91992266E 00
    DELTA C( 1) =     ,14967601E 00
    DELTA C( 2) =     ,13436616E 00
    DELTA C( 3) =     ,15234409E 00
    DELTA C( 4) =     ,16821694E 00
    DELTA C( 5) =     ,25108259E 00
    DELTA C( 6) =     ,23656310E 00
    CR TØTAL =        3,87172983
1   Q( 1) =     ,38038842E-02
3   Q( 2) =     ,67887181E-01
4   Q( 3) =     ,32015615E-01
4   Q( 4) =     ,89277134E-01
4   Q( 5) =     ,12622702E 00
4   Q( 6) =     ,19179390E 00


    NEW NI =        5
    P =     ,89409604E 00
    SUM NI*UNI =     ,85000000E 02
    SUM NI*VNI =     ,47000000E 02
    SUM NI*WNI =     ,67000000E 02
    PI( 1) =     ,99943000E 00
    PI( 2) =     ,98967277E 00
    PI( 3) =     ,99348969E 00
    PI( 4) =     ,98029108E 00
    PI( 5) =     ,95638775E 00
    PI( 6) =     ,97048258E 00
    DELTA C( 1) =     ,14967601E 00
    DELTA C( 2) =     ,13436616E 00
    DELTA C( 3) =     ,15234409E 00
    DELTA C( 4) =     ,16821694E 00
    DELTA C( 5) =     ,25108259E 00
    DELTA C( 6) =     ,28411266E 00
    CR TØTAL =        4,15829293
1   Q( 1) =     ,38038842E-02
3   Q( 2) =     ,67887181E-01
4   Q( 3) =     ,32015615E-01
4   Q( 4) =     ,89277134E-01
4   Q( 5) =     ,12622702E 00
5   Q( 6) =     ,66945686E-01


    NEW NI =        5
    P =     ,92243299E 00
    SUM NI*UNI =     ,90000000E 02
    SUM NI*VNI =     ,50000000E 02
    SUM NI*WNI =     ,71000000E 02
```

```
    PI( 1) =     ,99943000E 00
    PI( 2) =     ,98967277E 00
    PI( 3) =     ,99348969E 00
    PI( 4) =     ,98029108E 00
    PI( 5) =     ,98669894E 00
    PI( 6) =     ,97048258E 00
    DELTA C( 1) =     ,14967601E 00
    DELTA C( 2) =     ,13436616E 00
    DELTA C( 3) =     ,15234409E 00
    DELTA C( 4) =     ,16821694E 00
    DELTA C( 5) =     ,24863215E 00
    DELTA C( 6) =     ,28411266E 00
    CR TØTAL =        4,40937552
1   Q( 1) =     ,38038842E-02
3   Q( 2) =     ,67887181E-01
4   Q( 3) =     ,32015615E-01
4   Q( 4) =     ,89277134E-01
5   Q( 5) =     ,35703278E-01
5   Q( 6) =     ,66945685E-01


    NEW NI =        5
    P =     ,93626602E 00
    SUM NI*UNI =     ,94000000E 02
    SUM NI*VNI =     ,52000000E 02
    SUM NI*WNI =     ,73000000E 02
    PI( 1) =     ,99943000E 00
    PI( 2) =     ,98967277E 00
    PI( 3) =     ,99348969E 00
    PI( 4) =     ,99501302E 00
    PI( 5) =     ,98669894E 00
    PI( 6) =     ,97048258E 00
    DELTA C( 1) =     ,14967601E 00
    DELTA C( 2) =     ,13436616E 00
    DELTA C( 3) =     ,15234409E 00
    DELTA C( 4) =     ,16576650E 00
    DELTA C( 5) =     ,24863215E 00
    DELTA C( 6) =     ,28411266E 00
    CR TØTAL =        4,57759246
1   Q( 1) =     ,38038842E-02
3   Q( 2) =     ,67887181E-01
4   Q( 3) =     ,32015615E-01
5   Q( 4) =     ,18579104E-01
5   Q( 5) =     ,35703278E-01
5   Q( 6) =     ,66945686E-01


    NEW NI =        4
    P =     ,94482657E 00
    SUM NI*UNI =     ,96000000E 02
    SUM NI*VNI =     ,53000000E 02
```

```
    SUM NI*WNI =     ,76000000E 02
    PI( 1) =     ,99943000E 00
    PI( 2) =     ,99870031E 00
    PI( 3) =     ,99348969E 00
    PI( 4) =     ,99501302E 00
    PI( 5) =     ,98669894E 00
    PI( 6) =     ,97048258E 00
    DELTA C( 1) =     ,14967601E 00
    DELTA C( 2) =     ,12830137E 00
    DELTA C( 3) =     ,15234409E 00
    DELTA C( 4) =     ,16576650E 00
    DELTA C( 5) =     ,24863215E 00
    DELTA C( 6) =     ,28411266E 00
    CR TØTAL =        4,71195862
1   Q( 1) =     ,38038842E-02
4   Q( 2) =     ,35702797E-02
4   Q( 3) =     ,32015615E-01
5   Q( 4) =     ,18579104E-01
5   Q( 5) =     ,35703278E-01
5   Q( 6) =     ,66945686E-01


    NEW NI =        6
    P =     ,96279729E 00
    SUM NI*UNI =     ,10200000E 03
    SUM NI*VNI =     ,56000000E 02
    SUM NI*WNI =     ,81000000E 02
    PI( 1) =     ,99943000E 00
    PI( 2) =     ,99870031E 00
    PI( 3) =     ,99348969E 00
    PI( 4) =     ,99501302E 00
    PI( 5) =     ,98669894E 00
    PI( 6) =     ,98894127E 00
    DELTA C( 1) =     ,14967601E 00
    DELTA C( 2) =     ,12830137E 00
    DELTA C( 3) =     ,15234409E 00
    DELTA C( 4) =     ,16576650E 00
    DELTA C( 5) =     ,24363215E 00
    DELTA C( 6) =     ,23278722E 00
    CR TØTAL =        4,99607123
1   Q( 1) =     ,38038842E-02
4   Q( 2) =     ,35702797E-02
4   Q( 3) =     ,32015615E-01
5   Q( 4) =     ,18579104E-01
5   Q( 5) =     ,35703278E-01
6   Q( 6) =     ,22803930E-01
```

```
    NEW NI =        6
    P =     ,97134402E 00
    SUM NI*UNI =     ,10700000E 03
    SUM NI*VNI =     ,59000000E 02
    SUM NI*WNI =     ,85000000E 02
    PI( 1) =     ,99943000E 00
    PI( 2) =     ,99870031E 00
    PI( 3) =     ,99348969E 00
    PI( 4) =     ,99501302E 00
    PI( 5) =     ,99545785E 00
    PI( 6) =     ,98894127E 00
    DELTA C( 1) =     ,14967601E 00
    DELTA C( 2) =     ,12830137E 00
    DELTA C( 3) =     ,15234409E 00
    DELTA C( 4) =     ,16576650E 00
    DELTA C( 5) =     ,24730671E 00
    DELTA C( 6) =     ,28278722E 00
    CR TØTAL =        5,24470343
1   Q( 1) =     ,38038842E-02
4   Q( 2) =     ,35702797E-02
4   Q( 3) =     ,32015615E-01
5   Q( 4) =     ,18579104E-01
6   Q( 5) =     ,94016318E-02
6   Q( 6) =     ,22803930E-01


    NEW NI =        5
    P =     ,97608164E 00
    SUM NI*UNI =     ,11000000E 03
    SUM NI*VNI =     ,61000000E 02
    SUM NI*WNI =     ,67000000E 02
    PI( 1) =     ,99943000E 00
    PI( 2) =     ,99870031E 00
    PI( 3) =     ,99833533E 00
    PI( 4) =     ,99501302E 00
    PI( 5) =     ,99545785E 00
    PI( 6) =     ,98894127E 00
    DELTA C( 1) =     ,14967601E 00
    DELTA C( 2) =     ,12830137E 00
    DELTA C( 3) =     ,14989365E 00
    DELTA C( 4) =     ,16576650E 00
    DELTA C( 5) =     ,24730671E 00
    DELTA C( 6) =     ,28278722E 00
    CR TØTAL =        5,39704752
1   Q( 1) =     ,38038842E-02
4   Q( 2) =     ,35702797E-02
5   Q( 3) =     ,33676577E-02
5   Q( 4) =     ,18579104E-01
6   Q( 5) =     ,94016313E-02
6   Q( 5) =     ,22303930E-01
```

TABLE B.3.2. (Concluded)

```
    NEW NI =      7
    P =   ,98237606E 00
    SUM NI*UNI =  ,11600000E 03
    SUM NI*VNI =  ,64000000E 02
    SUM NI*WNI =  ,92000000E 02
    PI( 1) =   ,99943000E 00
    PI( 2) =   ,99870031E 00
    PI( 3) =   ,99833533E 00
    PI( 4) =   ,99501302E 00
    PI( 5) =   ,99545785E 00
    PI( 6) =   ,99531862E 00
    DELTA C( 1) =     ,14967601E 00
    DELTA C( 2) =     ,12830137E 00
    DELTA C( 3) =     ,14989365E 00
    DELTA C( 4) =     ,16576650E 00
    DELTA C( 5) =     ,24730671E 00
    DELTA C( 6) =     ,28195964E 00
    CR TØTAL =        5,67983474
1   Q( 1) =     ,38038842E-02
4   Q( 2) =     ,35702797E-02
5   Q( 3) =     ,33676577E-02
5   Q( 4) =     ,18579104E-01
6   Q( 5) =     ,94016318E-02
7   Q( 6) =     ,73044638E-02


    NEW NI =      6
    P =   ,98540158E 00
    SUM NI*UNI =  ,12000000E 03
    SUM NI*VNI =  ,66000000E 02
    SUM NI*WNI =  ,94000000E 02
    PI( 1) =   ,99943000E 00
    PI( 2) =   ,99870031E 00
    PI( 3) =   ,99833533E 00
    PI( 4) =   ,99807745E 00
    PI( 5) =   ,99545785E 00
    PI( 6) =   ,99531862E 00
    DELTA C( 1) =     ,14967601E 00
    DELTA C( 2) =     ,12830137E 00
    DELTA C( 3) =     ,14989365E 00
    DELTA C( 4) =     ,16444106E 00
    DELTA C( 5) =     ,24730671E 00
    DELTA C( 6) =     ,28195964E 00
    CR TØTAL =        5,84560124
1   Q( 1) =     ,38038842E-02
4   Q( 2) =     ,35702797E-02
5   Q( 3) =     ,33676577E-02
6   Q( 4) =     ,28644857E-02
6   Q( 5) =     ,94016318E-02
7   Q( 6) =     ,73044638E-02


    NEW NI =      7
    P =   ,98769272E 00
    SUM NI*UNI =  ,12500000E 03
    SUM NI*VNI =  ,69000000E 02
    SUM NI*WNI =  ,98000000E 02
    PI( 1) =   ,99943000E 00
    PI( 2) =   ,99870031E 00
    PI( 3) =   ,99833533E 00
    PI( 4) =   ,99807745E 00
    PI( 5) =   ,99777237E 00
    PI( 6) =   ,99531862E 00
    DELTA C( 1) =     ,14967601E 00
    DELTA C( 2) =     ,12830137E 00
    DELTA C( 3) =     ,14989365E 00
    DELTA C( 4) =     ,16444106E 00
    DELTA C( 5) =     ,24647913E 00
    DELTA C( 6) =     ,28195964E 00
    CR TØTAL =        6,09290796
1   Q( 1) =     ,38038842E-02
4   Q( 2) =     ,35702797E-02
5   Q( 3) =     ,33676577E-02
6   Q( 4) =     ,28644857E-02
7   Q( 5) =     ,19496972E-02
7   Q( 6) =     ,73044638E-02


    NEW NI =      8
    P =   ,98972694E 00
    SUM NI*UNI =  ,13100000E 03
    SUM NI*VNI =  ,72000000E 02
    SUM NI*WNI =  ,10300000E 03
    PI( 1) =   ,99943000E 00
    PI( 2) =   ,99870031E 00
    PI( 3) =   ,99833533E 00
    PI( 4) =   ,99807745E 00
    PI( 5) =   ,99777237E 00
    PI( 6) =   ,99736854E 00
    DELTA C( 1) =     ,14967601E 00
    DELTA C( 2) =     ,12830137E 00
    DELTA C( 3) =     ,14989365E 00
    DELTA C( 4) =     ,16444106E 00
    DELTA C( 5) =     ,24647913E 00
    DELTA C( 6) =     ,28139593E 00
    CR TØTAL =        6,37486759
1   Q( 1) =     ,38038842E-02
4   Q( 2) =     ,35702797E-02
5   Q( 3) =     ,33676577E-02
6   Q( 4) =     ,28644857E-02
7   Q( 5) =     ,19496972E-02
8   Q( 6) =     ,19665046E-02


    NEGATIVE Q FØUND, ELIMINATE MØDULE ( 1)
    NEGATIVE Q FØUND, ELIMINATE MØDULE ( 2)
    NEGATIVE Q FØUND, ELIMINATE MØDULE ( 3)
    NEGATIVE Q FØUND, ELIMINATE MØDULE ( 4)


    NEW NI =      9
    P =   ,99027462E 00
    SUM NI*UNI =  ,13100000E 03
    SUM NI*VNI =  ,72000000E 02
    SUM NI*WNI =  ,10300000E 03
    PI( 1) =   ,99943000E 00
    PI( 2) =   ,99870031E 00
    PI( 3) =   ,99833533E 00
    PI( 4) =   ,99807745E 00
    PI( 5) =   ,99777237E 00
    PI( 6) =   ,99792045E 00
    DELTA C( 1) =     ,79278004E-01
    DELTA C( 2) =     ,12585094E 00
    DELTA C( 3) =     ,14856821E 00
    DELTA C( 4) =     ,16361348E 00
    DELTA C( 5) =     ,24647913E 00
    DELTA C( 6) =     ,28098873E 00
    CR TØTAL =        6,65626352
1   Q( 1) =     ,00000000E 00
4   Q( 2) =     ,00000000E 00
5   Q( 3) =     ,00000000E 00
6   Q( 4) =     ,00000000E 00
7   Q( 5) =     ,19496972E-02
9   Q( 6) =     ,16770129E-03


    NEGATIVE Q FØUND, ELIMINATE MØDULE ( 5)
    NEGATIVE Q FØUND, ELIMINATE MØDULE ( 6)

ALL MØDULES ELIMINATED
```

VITA

J. B. White, Jr.

Candidate for the Degree of

Doctor of Philosophy

Thesis: A METHOD FOR THEORETICALLY OBTAINING AND PHYSICALLY REALIZING UNLIMITED RELIABILITY THROUGH REDUNDANCY

Major Field: General Engineering

Biographical:

Personal Data: Born April 3, 1933 at Ellsworth, Arkansas, the son of James B. and Sally H. White. Adolescent and teen-age years were spent in Enid, Oklahoma.

Education: Attended junior high and high school in Enid, Oklahoma, graduating from Enid High School in May 1951. Attended and graduated from Oklahoma State University in May 1956, receiving a Bachelor of Science Degree in Electrical Engineering. Pursued graduate work at the University of Alabama, Huntsville from 1957 to 1968. Completed a year residence work at Oklahoma State University in August 1969. Research was accomplished in absentia at Marshall Space Flight Center of National Aeronautics and Space Administration; the requirements for the Doctor of Philosophy Degree completed in May 1971.

Professional Experience: Employed by the Fort Worth Division of General Dynamics Corporation from June 1956 to November 1956. Spent the next two years with the U. S. Army Ordnance Corps. Employed by the U. S. Army Ballistic Missile Agency from December 1958 until July 1960 when that group was transferred in mass to the National Aeronautics and Space Agency, forming the Marshall Space Flight Center. Employed by the Marshall Space Flight Center since its inception in 1960 until the present.