NASA CONTRACTOR REPORT

NASA CR-343

NASA CR-343

FINAL REPORT ON PHASE II OF RESEARCH ON FAILURE FREE SYSTEMS

Prepared under Contract No. NASw-572 *by*
WESTINGHOUSE ELECTRIC CORPORATION
Baltimore, Md.
for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION - WASHINGTON, D. C. - DECEMBER 1965

NASA CR-343

FINAL REPORT ON PHASE II OF RESEARCH ON FAILURE FREE SYSTEMS

Prepared under Contract No. NASw-572 by
WESTINGHOUSE ELECTRIC CORPORATION
Baltimore, Md.

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – Price $5.00

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

SECTION 1
GENERAL PROJECT INFORMATION

A. PURPOSE

This report is prepared in accordance with the requirements of Contract NASw-572, "Research on Failure Free Systems", between the National Aeronautics and Space Administration and the Westinghouse Electric Corporation (reference WGD-38521). The research that is reported herein has the general objective of the advancement of the state-of-the-art in the design of highly reliable electronic systems which can be expected to be associated with the national space effort. The design objectives which are studied are those which permit the proper operation of systems to be relatively insensitive to the effects of individual internal component or subsystem failures.

The scope of this program has included the development of new techniques for organizing and implementing systems which more efficiently use redundant equipment to insure system operation and the development of procedures for testing a variety of redundant systems. The research performed during this program has been divided into four major task areas:

1. Statistical Measure of Quality
2. Implementation of an Adaptive Voter
3. Failure Responsive System Organizations
4. Medium Communication for Subsystem Reorganization

The four remaining sections and the Appendix of this report describe the work which has been done in each of these major task areas. Because the details of the work in each of these areas is relatively independent of the work in the other areas, each of the sections is self contained and may be read as a separate report. The possible exception to this condition is Section 4 which contains extensive cross reference to the Appendix, a previously published report describing earlier stages of work on the same task.

The remaining portions of Section 1 provide:

1. A brief summary of contents of each of the other sections including the Appendix.
2. The conclusions and recommendations drawn from each of the major task efforts.
3. A list of the personnel directly contributing to the project.

B. INDIVIDUAL TASK SUMMARIES, CONCLUSIONS, AND RECOMMENDATIONS

1. Task I - Statistical Measure of Quality

Several investigators have shown that the reliability of electronic systems can be greatly increased through the proper use of "redundant" equipment to provide alternate signal paths. Other investigators studied the problem of determining the optimum interval between system checkout. Relatively little analytical work has been done in the development of procedures for estimating system reliability based on results obtained from testing only parts of the system. Similarly, little attention has been given to the allocation of test points among subsystems of redundant networks subject to the mission requirements and the decision criteria of the system user.

The task has included the development and analytical derivation of several techniques for performing test point allocation and system reliability estimation incorporating a broad range of system parameters. These techniques are all formulated in a manner that permits the usage of dynamic programming techniques for their solution. Common to all formulations and the basis for the development of solutions, are the expressions for the conditional probabilities of success based on the time of the test and test results. The techniques and methodologies employed in the derivations described in this report serve as illumination to answer the question of what to consider and how one might combine the critical parameters associated with this type of testing.

2. Task II - Implementation of an Adaptive Voter

One of the most effective practical techniques for introducing redundancy into digital systems is employed in multiple-line voted systems. In this configuration, a system is divided into a group of identifiable subsystems, which are replicated two or more times and interspersed with redundant voting circuits. A typical voting circuit examines the set of nominally identical signals at its inputs, and, based on this input information provides an estimate of what the correct output signal from the subsystem set should be.

The most common restoring network is a "majority voter". In order to make a correct estimate of the output for a set of subsystems, the majority voter requires that no $\frac{n+1}{2}$* of its inputs be failed to the same state. Although this network is effective when n = 3, it is very inefficient when $n > 3$. This ineffectiveness exists because the percentage of the redundant subsystem which must be operating correctly to permit a correct vote is undesirably large.

---

* n is the number of inputs to the restoring network, i.e., the "order of redundancy".

In order to realize the advantages of voters which can operate correctly with less than a majority of correct inputs, some adaption scheme for "deweighting" faulty inputs must be provided. In studies at Stanford Research Laboratories and the Westinghouse Research Laboratories, Dr. W.H. Pierce has devised several schemes for optimally weighting inputs as a function of their past history of errors.

As part of this Failure Free Systems study, Westinghouse has been attempting to bridge the gap between Pierce's theoretical studies and the construction and use of a practical adaptive voter. Inherent in the basic design of an adaptive voter is the requirement for an electrically variable conductance (or weight) device which performs integration and displays relatively permanent memory of the established weight. These special characteristics have stimulated considerable effort toward the development of suitable adaptive components. The devices of this type which have been proposed generally utilize phenomena involving atomic translation or rotation.

During the first phase of this contract, an extensive survey of the more promising of these devices was made. At the completion of the survey, and at the beginning of the present phase of the contract, the mercury cell integrator with photoelectric readout appeared to offer the most attractive approach because of its simplicity, stability in time and general compatibility with conventional circuitry. Because the output is essentially a variable resistance proportional to the integral of the control input current, the device can be easily interfaced with more standard circuitry.

In concurrence with this finding, several of the mercury cell integrators were procured for evaluation. The remaining effort on this task has been concerned with the design and construction of an adaptive voter which employed these devices in an operational model. The specific purpose in designing and constructing this model was to determine the actual usefulness of such devices in an adaptive voter configuration.

The breadboard model of an adaptive voter which has been constructed for this program, consists of a hybrid combination of analog and digital circuitry and an on-line general purpose computer. The computer generates simulated input data for the voter and performs the feedback adaption control function inherent to the operation of the voter.

In the first portion of this dual role, the computer has been programmed to inject into a random data stream a variety of different error patterns. By selecting the proper error pattern the investigator has the capability to modify the statistical properties of the voter's input data to fit the requirement of almost any desired test. The use of the computer to perform the feedback control function offers the investigator an additional degree of

flexibility. To statistically test any proposed adaption scheme, a relatively simple subroutine must be prepared for the computer and inserted into the existing main program. To perform a test, the investigator needs only to supply the computer with the particular adaption subroutine to be considered and the information required to establish the simulated data characteristics.

The portion of the voter which has been implemented as actual circuitry consists of digital control equipment which increments the variable input weighting devices, the analog weighting devices, and output threshold and squaring circuits. The entire voter operation is described in Section 3.

3. Task III - Failure Responsive System Organizations

This task has been a continuation of the "Self-Repairing Systems" study which began during the first year of this contract. It was shown in that study that systems which have the capability to partially reorganize their redundant subsystems in response to existing internal failure patterns may be more resistant to early life system failures than comparable fixed redundant systems. The first goal of this study has been to develop design rules which will make such systems practicable. The second goal has been to design a specific study vehicle which can be used to demonstrate the feasibility of such systems.

The development of a set of design rules has been accomplished by comparing the relative effectiveness of a wide variety of system organizations through the use of a computer simulation program. The simulation program written for the previous phase of this study was not considered to have adequate flexibility to accomplish the goals of the present program. As a result, a new program has been written using many of the concepts of the original program, but based primarily on a "spare list" technique for determining system reorganization capability. This technique permits a much broader range of system parameters to be tested. A detailed description of the program is presented in the Appendix. The following presents a brief summary of the program operation.

A system organization to be simulated is represented in the computer by a three-dimensional matrix with one dimension corresponding to the stages in the system, a second dimension corresponding to the order of redundancy found within the system and a third dimension corresponding to the data words to be remembered about each individual subsystem. In the new program the data words associated with each subsystem include a complete "spare list". This list specifies the set of subsystems which can be sequentially called from the rest of the system to replace each failed subsystem that becomes a part of a strategic pattern. The data is read into the computer as a simple list of subsystem identification numbers.

Using this listing method, almost any conceivable sequence of spares can be established by simply modifying the "spare list" input data. This is in marked contrast to the reprogramming previously required to generate new sequences.

In addition to the input "spare list" a number of other control variables are read into the computer to simulate specific system organizations. These variables affect system characteristics such as the capability of certain subsystems to perform multiple repairs, the minimum amount of instantaneous failure masking required, and the relative reliability of the subsystems and the peripheral switching circuitry.

The simulation study also has been extended to include orders of redundancy different from order-three. This includes fractional and even orders of redundancy. As an example of the results obtained from this portion of the study, it has been shown that three-and-one-half order* failure responsive systems are potentially much more reliable than order-five multiple-line majority-voted systems.

The use of reorganizational strategies which employ a "pool" of spare subsystems in an initially "off-line" operation has been avoided for a number of reasons. The most important reason is that no automatic checkout is provided for the spares in this pool, and, as a result, spares which are already failed can be called into use. This system could allow two failed subsystems to control the majority vote of a stage, thus inadvertently failing the entire system.

Because the Mean Time Before Failure is not a particularly meaningful reliability measure for redundant systems with relatively short but vital missions, a new measure was adopted for comparing failure responsive systems. The measure is the time at which the reliability of the systems falls below some predetermined level. For this study, the .90 level has been used.

A new criterion for evaluation has also been developed for this study which permits a single value measure of effectiveness to be associated with each system organization under test. Provisions have been made for incorporating the calculations required to determine the value of this measure into the computer program. This facilitates a quick and relatively accurate evaluation of system performance.

The remainder of the effort performed on this task has been oriented toward the design of the study vehicle. The design rules developed through the use of the comparison study described above have been used as a basis for the study vehicle design. The switching

* A "three-and-one-half" order system is a system initially having half its stages order-three and the other half order-four redundant.

strategy incorporated in the design is the "step list" pattern. This is one of the better strategies developed during the simulation study. Three "spares" are provided for each stage in the system, with all spares having exactly the same mobility.

After an extensive investigation it was decided that the type of system which would best demonstrate the feasibility of the failure response techniques would be a special purpose arithmetic unit.

The specific study vehicle which has been selected is the arithmetic section of a beam steering computer used in a phased array radar. This unit receives data, performs a number of arithmetic operations on the data, temporarily stores, and then reads out the results. The arithmetic unit, which must provide a given sequence of operations properly timed in relation to inputs and outputs, plus storage of intermediate results during computation, was chosen to tax both the reorganizational strategy theories and the implementation techniques which have been developed.

The beam steering computer consists of four identical subsystems, each consisting of two adder-subtractors, two shift registers, and one full adder. The circuitry required to implement a single subsystem consists of 33 gates and 31 flip-flops. The system operates on a three phase cycle: input data read-in, arithmetic computation and storage, and results output.

4. Task IV - Medium Communication for Module Reorganization

In Task III of the Failure Free Systems study, it has been found that systems which have the capability to partially reorganize the connection pattern linking their individual subsystems have significantly longer useful life spans than systems with a fixed subsystem configuration. The reorganization capability allows these systems to avoid the use of failed subsystems and to maintain a uniform distribution of the operating redundant subsystems. The two inherent primary difficulties with systems having this capability are (1) the need for relatively complex interconnection switching circuitry and (2) the need for highly homgeneous subsystems.

Any system in which every subsystem has the capability to communicate with every other system is often referred to as a "strongly connected system". Systems may be strongly connected through a system of individual signal channels such as wires or through a common medium such as a gas, a liquid or a block of solid material. Subsystems communicating over individual signal channels may require as many as N^2 unidirectional channels or at least $N^2/_2$ bidirectional channels with the associated channel selection circuitry available at each subsystem. If, however, a medium is used as a central mode through which all data

passes, the characteristics of a strongly connected system are retained, but the number of channels is reduced to 2N unidirectional channels or N bidirectional channels with all or most of the channel selection circuitry confined to the medium.

A typical, although rather mundane, example of the economy of using a central medium to contain the channel selection circuitry is the telephone system. In this case the switchboard fills the role of a medium which performs all the channel selection functions for the system. This example also illustrates that a large system might profitably be broken into segments organized around individual media each of which communicates with the other media. Although the question is academic at this point, the question of whether the individual media should communicate directly or through a "higher rank" media is one which must be solved before the very large, complex computing systems could be implemented.

This task has been concerned with the initial investigation of structures which can easily be reorganized through the use of a medium communication channel. The investigation has led to the formulation of a general type of computer organization which fulfills the objective of this task. As the formulation exists, the system function would be performed by a group of subsystems communicating through a common medium. The medium also stores any information which each subsystem would normally contain in any storage which was not controlled by the inputs. Subsystem outputs are stored (and voted on) in the medium. The channeling of the information would be accomplished either by tagging the data with some time or frequency code, by providing an instruction program within each subsystem or by providing a central controller which would be associated directly with the medium.

C. CONCLUSIONS

1. Task I - Statistical Measure of Quality

The objective of this task has been to develop a procedure for optimally allocating test points to the subsystems in a redundant system subject to one or more limiting criteria. This has been done for a particular system model.

If the assumption is made that the operation or failure of every stage is statistically independent of the operation or failure of all subsystems outside the stage, the present analysis technique may be extended to a fairly broad class of systems of predominantly serial configurations. These systems may include feedback loops, feedforward loops, diverging branches and converging branches as described in great detail in Nemhauser's work. If this assumption of independence cannot be made, a more sophisticated analysis procedure must be used. It is recommended that future work in this area include the investigation of existing techniques of this latter type and the determination of the applicability of the allocation procedure.

Regardless of the configuration of the system model, the technique developed here can handle any number of test points that a single unit may require. Here, the implication is that the total number of test points necessary to verify the success or failure of a subsystem be grouped as a "unit test point." No allocations of partial unit test points are permitted. If only partial success or failure information can be obtained on a unit then each element of the E (V) expression must be rederived. The assumption must also be made that the cost functions associated with adding test points within the stages must be monotonically non-decreasing.

For most practical systems, the number of computations required to complete the allocation procedure is quite large. It is recommended that before solution of any realizable system be attempted, the entire allocation procedure, including a generalized reliability analysis method, be implemented as a computer program which is amenable to solution on a large scale digital computer.

2. Task II - Implementation of An Adaptive Voter

The results of this project have shown that adaptive voters can be constructed using mercury cell integrators as the variable input weighting devices. The results have also shown that the implementation of such voters using voting schemes which require computation of optimal weight values require relatively complex feedback adaption control loops. This last requirement, combined with the problems involved in using the presently available models of the mercury cell integrators, leads to the conclusion that adaptive voters of this type are presently an impractical means of improving the reliability of redundant systems.

Despite this rather negative conclusion, the potential improvement in system reliability offered by the use of adaptive voters cannot be ignored. In order to advance the art toward the realization of practical adaptive voting techniques, the following recommendations for future work in this area are made:

a. The search for suitable weighting devices should be actively continued, and consideration should be given to making physical changes in the mercury cell integrators to eliminate some of the present problems.

b. A broad range of adaption schemes including those proposed by Pierce should be examined on a comparative basis. The objective in performing this comparison would be to determine the cost, in terms of lost reliability, of using simple, easy to implement schemes rather than the more sophisticated "optimal" adaption schemes. Although many adaption schemes are not amenable to mathematical analysis, the comparison would be relatively easy

to perform through the use of a computer simulation program similar to the one used in the present project. In this case, however, the entire voter would be simulated rather than just the feedback adaption control loop.

c. Using the results of the comparison study recommended above,and if suitable weighting devices are available which have the characteristics required by the particular adaption schemes under consideration, complete breadboard models of the more promising schemes should be constructed and tested.

3. Task III - Failure Responsive System Organizations

Using the computer simulation program prepared for this task, an extensive series of tests were made to determine the most effective switching patterns which could be used in the fabrication of failure responsive systems.

The results of these tests show that the members of one particular class of response strategies are the most efficient of all the well-ordered strategies which were tested. The observance of this characteristic and the recognition of the value of "rescan" capability leads to the following general conclusions:

1. The capability of individual subsystems to move to new locations should be as evenly distributed among the subsystems as possible.

2. The subsystems which are available for use as spares (or replacements) to any two stages should be chosen so that the mutual dependence by these stages on the same spares is minimized.

3. The systems should be so organized that, in normal circumstances, a subsystem will not move to the aid of a critically failed stage if its movement will leave the stage in which it is presently operating vulnerable to a single failure. A critically failed stage should have the "authority", however, to demand the movement of a spare subsystem if the movement of all of the spare subsystems available to this stage are restricted as above.

It can also be concluded from the test results, that order-two-and-one-half redundant failure responsive systems may effectively replace order-three redundant multiple-line systems in applications where instantaneous failure masking is not important. Conversely, applications with either high instantaneous failure masking or exceptionally long life requirements may be benefitted by employing order-three-and-one-half or order-four redundant failure responsive systems to replace order-three, or even order-five, multiple-line systems.

From other results obtained in the study, it may be concluded that the beneficial effects obtained from failure responsive capability appear to more than offset the disadvantages inherent in the relatively complicated circuitry required for system implementation. These curves show that the useful lives of the example systems have been significantly increased over those of the corresponding examples of multiple-line systems. These increases have been realized despite relatively pessimistic assumptions regarding the reliability of the error detection and switching circuitry.

Finally, it may be concluded that the optimum number of spare subsystems which should be made available to any stage is a function of the failure rate of the peripheral circuitry relative to the failure rate of the subsystems. It can be seen from the results of the study that for systems having relatively simple subsystems the optimum number of available spare subsystems per stage will be around three to five.

The design of a practicable system having failure responsive capability has been accomplished. This design has shown that such systems can be implemented using standard combinational logic circuits to form the various error detection, error correction and "repair" switching functions which are required.

Although the amount of peripheral circuitry required to implement the functions mentioned above does not seem excessive, it may be concluded that the subsystems must be at least as complex as those described for the beam-steering computer considered here. The successful design of this particular study vehicle demonstrates the applicability of failure responsive system techniques to systems containing input-controlled memory. In addition, the design has shown that subsystem with multiple inputs can be handled with the reorganizational capability of these systems.

The present design of the study vehicle contains no specific provisions to protect the system against all failure modes of the peripheral circuitry. In many cases, failures in this circuitry will be treated as subsystem failures. The natural extention of this work would be to continue the design effort to provide protection against all peripheral circuit failure modes.

The existing computer simulation program does not provide for the simulation of failures in redundant peripheral circuitry or in peripheral circuitry which affects the operation of more than one subsystem. It is recommended that the program be modified to include this simulation capability.

4. Task IV - Medium Communication for Module Reorganization

The results of this exploratory investigation have led to the conclusion that the system structures employing a medium communications channel offer the following potential advantages to users of ultra-reliable computing systems.

a. High Subsystem Mobility

1. The prime advantage is that this type system offers one means for realizing the potential benefits of failure-responsive systems. Indeed, in the system where each subsystem can perform all the functions,the maximum "mobility" of the subsystems has been achieved because every subsystem may replace any other subsystem as the failure pattern occurs. In this system two of the main restrictions on mobility have been removed. The first restriction is that all subsystems perform the same function. Subsystems are proposed which have the capability to change function when they change position. As a result the homogeneity difficulties inherent in fixed function subsystems do not arise. The second major obstacle to mobility occurs when a subsystem contains a fixed memory - i.e. a memory not set up within a few cycles by the data stream. Subsystems which are otherwise identical but which contain different information in their memory are not interchangeable. However, using a memory medium, memory which was formerly a part of the subsystem is now a part of the medium. Any subsystem may now associate itself with the part of the medium containing this memory and perform in this position just as well as any other subsystem.

2. Graceful Degradation

Graceful degradation of the system performance is inherently available in the system. This concept of graceful degradation assumes that the system is not used for only one purpose at a fixed data rate. In that case, a fixed computational capacity would be required. To have greater capability would be wasteful, and to have less capability would constitute complete failure.

The system proposed has this desirable property. For example, the system may first have twenty subsystems. If each subsystem is identical - can accomplish all the functions, then eighteen subsystems can fail (assuming two out of three voting) and the computer will still be able to do everything that was possible initially, but take ten times as long to do the same task.

3. Optimal Non-Redundant Operation

One of the most common objections to redundant systems is that they use three times the number of components without increasing computer capability. On the other hand, it may be desirable to operate three computers in parallel when failure is very expensive. As these two operational modes imply, having three individual computers gives one a choice between capability and reliability. This choice is available in an even more useful form with the medium system. By reprogramming the system the subsystems may do each process only once, increasing the power of the computer by a factor of two or three. This option may be very desirable for ground testing equipment before take off, or in any mode where reliability is not as important as speed.

Operation in the non-redundant mode is not alien to the system design and interweaving non-redundant and redundant operation is quite possible. This may be done even in the same computation if certain parts are not as significant as others.

4. Asynchronous Operation of Subsystems

The units in a processing system with memory may operate asynchronously. This autonomy relieves the programmer of timing problems and increases the efficiency of computation since subsystems need not wait for each other. As this condition implies, the memory serves both as a central medium and as a buffer store for each subsystem.

5. Efficient Use of Time Shared Subsystems

Because the subsystems are not restricted to a single functional location, but rather perform the next in a sequence of functions as they are needed, the subsystems may be shared between different problems. Priority interrupt is effected by placing the interrupting routine within the main program. The subsystems operate in parallel; hence, each subsystem is used to its full capacity.

A new development program such as this creates many new study areas. One approach to developing the organization in more detail would be to assume some properties for the subsystems and then write the programs to make them function as desired. This configuration could then be simulated on a general purpose digital computer. Such a procedure would insure that realistic solutions are found in each problem area.

The basic concept of a system implementation which employs a memory medium as a communication linkage between subsystems has been formulated. The investigators strongly recommend that this study be continued. The objectives of continuing this task should be the further development of a medium communication system implementation.

The next step toward reaching this objective should be accomplished by seeking answers to the following questions:

a. What type of voters should accompany a system of this type?

b. How many of the voters should be provided in relation to the number of functions being performed by the system, the operating speed of the voters, the data rate of the system, etc.

c. Should the system program be stored in the medium or in the subsystems?

d. Are there reasonable alternatives to the "drum" medium?

e. What "tag" information should accompany all data words stored in the medium?

f. In what pattern or order should the data be stored in the medium?

g. How many different functions should each subsystem be capable of performing?

h. How complex should the functions be?

D. PROJECT TEAM

This contract effort has been performed by the Advanced Development Subdivision of the Surface Division of the Westinghouse Electric Corporation. The effort has been one part of a broad program conducted by this subdivision in the development of techniques for constructing ultra-reliable electronic systems.

Mr. Sidney E. Lomax, Director of Development, has been responsible for the management of this contract since the inception in 1963. Mr. C. G. Masters, Jr., as project engineer, has been responsible for the technical direction of all tasks performed under the project. The performance of the individual tasks have been the responsibility of the following engineers.

Task I. - William A. Lutts
Task II. - James E. Thompson
Task III. - Joseph M. Hannigan
Charles G. Masters, Jr.
Task IV. - Kevin P. Shambrook

In addition to this principle team, several other engineers from the Advanced Development Subdivision have supported the individual task efforts in various specialized roles.

Henry F. DeFrancesco
Faris J. Kahwajy
William C. Mann
Karl C. Wehr
Thomas A. Woolverton

SECTION 2
STATISTICAL MEASURE OF QUALITY

2-1 INTRODUCTION

The steadily increasing sophistication of space missions has been reflected in an increased complexity of spaceborne electronic data processing and control systems. This increase in complexity tends to lower the reliability of systems which normally operate in an environment where the cost of system failure is extremely high. In many cases, this cost may include the loss of human life in addition to the loss of a space vehicle and an aborted mission.

Several teams of investigators have shown that the reliability of electronic systems can be greatly increased through the proper use of "redundant" equipment to provide alternate signal paths. By far the largest portion of work that has been done concerning the use of redundant equipment has been concentrated on the development of synthesis techniques and procedures for analyzing the initial reliability of systems implemented in a redundant configuration. Relatively little work has been done in the development of procedures for testing redundant systems and for estimating their reliability after they have been fabricated and released to the user.

The user of spaceborne electronic equipment has three different situations in which he may wish to test the equipment. The first situation exists when the equipment is being examined in a shop environment prior to being mounted in the space vehicle. In this situation, time is usually not of essence and exhaustive testing is desirable to the limit permitted by the physical design of the equipment. The second situation exists when the equipment has been mounted in the vehicle, and a test is to be made prior to launch. In this case, time is of the essence and an exhaustive test of an entire redundant network would usually be prohibitive. The third situation exists during long term, multi-phase space missions where tests are made at several preplanned intervals to determine which of a possible set of alternatives should be followed either during the next immediate phase or the remainder of the mission. In most cases the decision to be made is simply whether to continue or terminate the mission depending on the probability of successfully completing the next phase. In this case, both time of test and the complexity of test equipment are of vital interest.

In the latter two situations, there exists an obvious need for a technique to facilitate making an accurate estimate of the probability of successfully completing the mission based on information gained from testing only part of the system before or during the mission. A similar need often exists in the shop situation because the use of tightly packaged microminiature circuitry may severely limit the amount of individual subsystem testing which can be performed. This is true regardless of the time permitted for the test or the availability of sophisticated test equipment.

In answer to the questions of what portion of the equipment should be tested and how should the partial test results be utilized to obtain an estimate of the probability of mission success, an analysis has been made of one idealized system. (See figure 2-1.) This system was analyzed to determine the probability of mission success both without any testing and with only partial testing. These results were then used as a basis for developing procedures that might be utilized for optimally allocating a limited number of test points within a redundant system. In this case, an optimum allocation is one which provides more information concerning the operational states of the system than any other test point allocation having either the same number of test points and/or the same total cost of the test points. Thus, the problem of test point allocation may be limited by one or several constraints on the quantity of test points to be some number less than or equal to some fixed limit L_T and/or the total cost of placing the test points to be less than or equal to some fixed limit C_T.

2-2 DEFINITIONS

Several terms which are utilized in the discussion must be defined:

Unit. The smallest independently operating block of equipment in a stage.

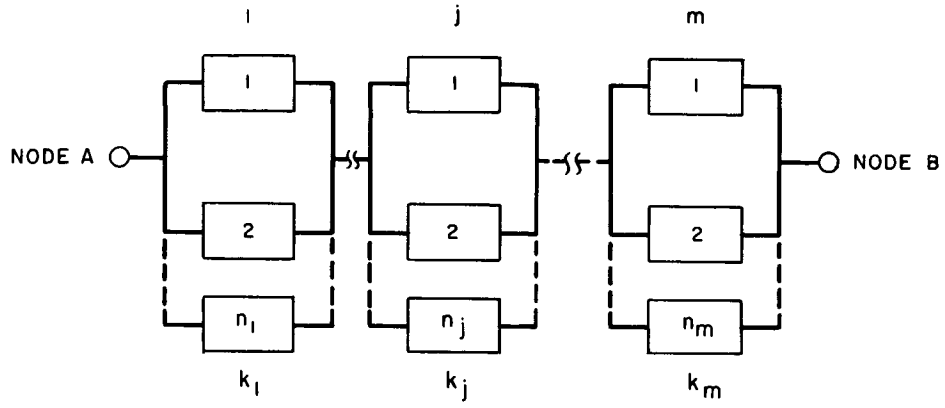
Stage. A set of identical units connected in parallel in such a manner that each unit's operation or failure is independent of that of the others in the stage.

System. A set of series connected stages where each stage is isolated from failures of every other stage.

Unit Successful State. A unit is capable of and operates in the manner for which it was designed.

Unit Failed State. Unit does not have the capability to operate in a manner for which it was designed.

Stage Successful State. A stage which has at least k_j out of n_j units in unit successful states.



n_j = number of units in stage j

k_j = least number of units required for stage it to be in a successful state.

$p_j(t)$ = probability of success for a unit in stage j at any time t.

$q_j(t)$ = probability of failure for a unit in stage j at any time t.

$p_j(t) + q_j(t) = 1$

j = 1, 2, ------- m

$n = \sum_{j=1}^{m} n_j$

Figure 2-1. Idealized System Model

Stage Failed State. A stage which has less than k_j units in unit successful states.

System Successful State. Each stage j out of m total stages has at least k_j out of n_j units in unit successful states simultaneously.

System Failed State. At least one stage j out of m total stages has less than k_j units in unit successful state.

Reliability. The probability of continuous successful operation over a specified period of time.

Unit Test Point. An item of equipment which determines the existence of the successful or failed state of a unit.

Unit Test Point Cost. The total dollar cost of obtaining the information on the successful or failed state of a unit.

2-3 PROBLEM FORMULATION

A block diagram of the system model under consideration is the one shown in figure 1. This system has a total of (m) stages and (n) individual units. Signals enter at node A and are processed in parallel by the units of each stage. The resultant output leaves the system at node B. The specific values of n_j, k_j and $p_j(t)$ at each stage j are allowed to be different for all stages. All failure modes of each unit will be assumed to have no effects on the status of the other units in the system. This means that the failure of a unit within a stage has no effect on the failure rate of other units within that stage and that the stages are independent of one another. The significant points to be made about this configuration are as follows:

1. Generality is preserved to the extent that n_j, the number of subsystems in stage j, and k_j, the minimum number of subsystems required for success, are allowed to vary from stage to stage.
2. The function $p_j(t)$ for the probability of unit success is allowed to vary from stage to stage.
3. All stages must be operating for the system to operate. This is an active redundant system. Each unit operates continuously at each stage until that unit enters a failed state.

A. Fixed Configuration Without Testing

The analysis problem is to determine the reliability expression for a fixed configuration such as that described above. For this configuration n_j, k_j, n, m and $p_j(t)$ are given. In the development of this reliability expression certain characteristics of the system should be noted.

1. For any unit there are two possible states, successful or failed. For a total of n units there are 2^n possible states that the system might assume at any point in time given that this system is allowed to operate until all units fail.
2. In any one stage there are $\sum_{k=k_j}^{n_j} \binom{n_j}{k}$ possible successful states. Since the summation represents the total number of ways of having success in j^{th} stage, the total number of ways of having system success becomes $\prod_{j=1}^{m} \sum_{k=k_j}^{n_j} \binom{n_j}{k}$.

3. From 2 above and the fact that for each stage all n_j units are identical, there are $(n_j - k_j + 1)$ distinguishable stage successful states. For the system then, there are $\prod_{j=1}^{m} (n_j - k_j + 1)$ possible system successful states which are distinguishable.

4. Letting $P_{s_j}(t)$ be the probability of stage success, then for any stage j the expression is a simple form of the binomial:

$$P_{s_j}(t) = \sum_{k=k_j}^{n_j} \binom{n_j}{k} p_j^k(t)\, q_j^{n_j-k}(t)$$

$$= \sum_{k=k_j}^{n_j} \binom{n_j}{k} p_j^k(t) \left[1 - p_j^k(t)\right]^{n_j-k}$$

$$P_{s_j}(t) = \left[1 - p_j(t)\right]^{n_j} \sum_{k=k_j}^{n_j} \binom{n_j}{k} \left[\frac{P_j(t)}{1 - P_j(t)}\right]^k$$

5. For the entire system then the reliability expression for any point in time is:

$$P_s(t) = \prod_{j=1}^{m} P_{s_j}(t) = \prod_{j=1}^{m} \left[(1 - p_j(t))^{n_j} \sum_{k=k_j}^{n_j} \binom{n_j}{k} \left[\frac{P_j(t)}{1 - p_j(t)}\right]^k \right]$$

It should be noted here that this expression holds at any time t given that no tests are conducted after some reference time t_o.

B. Fixed Configuration With Testing

For the problem of conducting a test at some point in time t_1 and utilizing the results to determine the probability of mission success at some future time $t_m > t_1$, the expression for $P_s(t)$ is different. The test of the system is such that two general questions must be answered:

1. Is the system in any one of the system successful states?
2. Which units out of those tested are in a successful state and which are in a failed state?

If the answer to the first question is no, then the probability of mission success is zero. If the answer is yes, the user employs the results of the second question to make a better estimate of mission success.

Let:

s_j = the total number of units operating out of ℓ_j in stage j at some test time t_1.

ℓ_j = total number of test points in stage j.

z_j = the exact number of units out of $n_j - \ell_j$ that are operating at some point in time t_1.

s = the exact number of units out of s_j that are operating at time t_m.

Depending on the relative values of $n_j - \ell_j$, s_j and k_j, the new estimate of the probability of mission success of a stage based on the test results may be divided into three general cases.

For notational convenience the following convention will be utilized to represent the binomial for any set of variables r, i and values of t.

$$p_j^* (t_k - t_u) = P_j^j (t_k - t_u)\, q_j^{r-i} (t_k - t_u)$$

Case I $s_j = 0$ The number of units with test points that are successfully operating in stage j at (t_1) is zero.

(1) If $n_j - \ell_j < k_j$, the new probability of mission success $P_{s_j} = 0$ since the number of units operating in the stage j is less than the minimum required k_j.

(2) If $n_j - \ell_j \geq k_j$, the value of $P'_s (t_m)$ depends on the value of z_j where $z_j = k_j, k_j + 1 \ldots\ldots\ldots\ldots n_j - \ell_j$ From the results of a test, it is known that $s_j = 0$ and that the system is operating. This latter information implies that the successful units are of the set $n_j - \ell_j$ and the value of z_j is indeed greater than or equal to k_j.

Let:

K = The condition exists that $k_j \leq z_j \leq (n_j - \ell_j)$

k = the number of units that survive until t_m

Then

$$P'_{s_j} (t_m) = \sum_{z_j = k_j}^{n_j - \ell_j} P (z_j / K)\, P (k_j \leq k \leq z_j)$$

where

$$P(k_j \le k \le z_j) = \sum_{k=k_j}^{z_j} \binom{z_j}{k} P_j^{*} (t_m - t_1)$$

$$P(z_j/K) = \frac{\binom{n_j-\ell_j}{z_j} P_j^{*} (t_1 - t_0)}{\sum_{\ell=k_j}^{n_j-\ell_j} \binom{n_j-\ell_j}{\ell} P_j^{*} (t_1 - t_0)}$$

Combining:

$$P'_{s_j}(t_m) = \frac{\sum_{z_j=k_j}^{n_j-\ell_j} \left[\binom{n_j-\ell_j}{z_j} P_j^{*} (t_1-t_0) \sum_{k=k_j}^{z_j} \binom{z_j}{k} P_j^{*} (t_m-t_1) \right]}{\sum_{\ell=k_j}^{n_j-\ell_j} \binom{n_j-\ell_j}{\ell} P_j^{*} (t_1-t_0)}$$

Case II $0 < s_j < k_j$ The number of units with test points that are successfully operating in stage j is less than k_j and greater than zero.

(1) If $n_j-\ell_j = 0$, the new probability of mission success $P'_s(t_m) = 0$ since all n_j units have test points and the number operating at t_1, namely s_j, is less than the minimum required k_j.

(2) If $(n_j-\ell_j+s_j) < k_j$, the new probability of mission success $P'_s(t_m) = 0$ since the total of $s_j+z_j < k_j$.

(3) If $0<(n_j-\ell_j)<k_j$ and $(n_j-\ell_j+s_j)\ge k_j$, then there is at least one combination $(s + z_j)\ge k_j$ at time t_1. In this case we are given that the answer to question two for stage j is that there are s_j units in successful states and ℓ_j-s_j in unit failed states. Since it is known exactly which s_j units are operating, the probability that the s_j units are operating, the probability that the s_j units survived until t_1 at t_1 is one. Thus, the user of the test results knows that there must be at least k_j-s_j units operating out of the $n_j-\ell_j$ because of the result of question one.

Since $n_j-\ell_j < k_j$, the minimum number of units from the set s_j, that are required to operate for the period t_m-t_1 given that all $n_j-\ell_j$ are operating for that same period, is $s = k_j-(n_j-\ell_j)$. Thus, the lower limit on z_j becomes $z_j = k_j-s$ for any s in the range $k_j-(n_j-\ell_j) \le s \le s_j$. In this case, the range of z_j becomes $k_j-s \le z_j \le n_j-\ell_j$.

Formulating the probability $P'_{s_j}(t_m)$ for this set of circumstances yields:

$$P'_{s_j}(t_m) = \frac{\sum_{s=k_j-(n_j-l_j)}^{s_j} \left[\binom{s_j}{s} P_j^*(t_m-t_1) \left[\sum_{z_j=k_j-s}^{n_j-l_j} \left[\binom{n_j-l_j}{z_j} P_j^*(t_1-t_0) \sum_{k=k_j-s}^{z_j} \binom{z_j}{k} P_j^*(t_m-t_1) \right] \right] \right]}{\sum_{l=k_j-s_j}^{n_j-l_j} \binom{n_j-l_j}{l} P_j^*(t_1-t_0)}$$

(4) If $n_j-l_j \geq k_j$, there is at least one combination $s+z_j \geq k_j$ at time t_1 which satisfies the successful operation criterion. In this case we are given that there are s_j units in successful states and l_j-s_j in unit failed states. In this case the minimum number of units from the s_j, that are required to operate for the period t_m-t_1, given that at least k_m of the n_j-l_j are in unit successful states, is zero.

Incorporating this into the expression for (3) above yields:

$$P'_{s_j}(t_m) = \frac{\sum_{s=0}^{s_j} \left[\binom{s_j}{s} P_j^*(t_m-t_1) \sum_{z_j=k_j-s}^{n_j-l_j} \left[\binom{n_j-l_j}{z_j} P_j^*(t_1-t_0) \sum_{k=k_j-s}^{z_j} \binom{z_j}{k} P_j^*(t_m-t_1) \right] \right]}{\sum_{l=k_j-s_j}^{n_j-l_j} \binom{n_j-l_j}{l} P_j^*(t_1-t_0)}$$

Case III $s_j \geq k_j$ The number of units with test points that are successfully operating in stage j is greater than or equal to k_j.

(1) If $n_j-l_j=0$, the number of units required for success, $s+z_j \geq k_j$, will reduce to $s \geq k_j$, thus, the expression for $P'_{s_j}(t_m)$ becomes

$$P'_{s_j}(t_m) = \sum_{s=k_j}^{s_j} \binom{s_j}{s} P_j^*(t_m-t_1)$$

(2) If $n_j-l_j<k_j$, there is at least one combination $s+z_j \geq k_j$ at the time t_1 which satisfies the successful operation criterion. In this case the user knows that there are l_j units that are successful but knows nothing about the other n_j-l_j. The ranges for s and z_j in the expression for $P'_{s_j}(t_m)$ then depend on the condition of the n_j-l_j units. Since z_j can have the range $0 \leq z_j \leq n_j-l_j$, then s must range

from $k_j - (n_j - \ell_j) \le s \le s_j$. For the determination of $P'_{s_j}(t_m)$, the lower limit on z_j will become $z_j = k_j - s$ thus reflecting the dependence of stage success on z_j. Combining the above results yields the same expression for $P'_{s_j}(t_m)$ as that found in Case II(3).

(3) If $n_j - \ell_j \ge k_j$, again, there is at least one combination $s + z_j \ge k_j$ at time t_1 which satisfies the successful operation criterion. In this case the user knows that there are s_j units successful but still knows nothing about the other $n_j - \ell_j$. However all the comments from (2) above hold with the exception that the quantity $k_j-(n_j-\ell_j)$ will be less than or equal to zero, thus allowing s to take on all values from 0 through s_j. Therefore the expression for $P'_{s_j}(t)$ is the same as that found in Case II(4).

In summary, the following table represents the ranges of the indices s and z for all the above cases which yield an $P'_{s_j}(t_m) \neq 0$.

	$n_j-\ell_j=0$	$0<n_j-\ell_j<k_j$	$n_j-\ell_j\ge k_j$
	——	——	$s=0$
$s_j=0$	——	——	$k_j\le z_j\le n_j-\ell_j$
$0<s_j<k_j$	——	$k_j-(n_j-\ell_j)\le s\le s_j$	$0\le s\le s_j$
$s_j+n_j-\ell_j\ge k_j$	——	$k_j-s\le z_j\le n_j-\ell_j$	$k_j-s\le z_j\le n_j-\ell_j$
$k_j\le s_j\le \ell_j$	$k_j\le s\le s_j$	$k_j-(n_j-\ell_j)\le s\le s_j$	$0\le s\le s_j$
	$z=0$	$k_j-s\le z_j\le n_j-\ell_j$	$k_j-s\le z_j\le n_j-\ell_j$

Table 1

Taking table 1, the general expression for $P_s(t_m)$ for all m stages may be written as follows:

$$P'_s(t_m) = \prod_{j=1}^{m} \left[\frac{\sum_{s=k_j-(n_j-\ell_j)}^{s_j} \binom{s_j}{s} P_j^*(t_m-t_1) \left[\sum_{z_j=k_j-s}^{n_j-\ell_j} \binom{n_j-\ell_j}{s} P_j^*(t_1-t_0) \sum_{k=k_j-s}^{z_j} \binom{z_j}{k} P_j^*(t_m-t_1) \right] \right]}{\sum_{\ell=k_j-s_j}^{n_j-\ell_j} \binom{n_j-\ell_j}{\ell} P_j^*(t_1-t_0)} \right]$$

2.4 TEST POINT ALLOCATION

In the previous sections, definitions have been stated and the system reliability estimates have been developed for configurations with and without test results. However, the allocation of test points and the reasoning behind their placement has not been answered. In the placement of any single test point, due consideration must be given to the "value" of locating it with any specific unit.

A. Value Function

For any decision to locate a single test point, the ultimate gain to the user of the test results is the incremental change in the estimate of system reliability. This ultimate payoff will be considered as a meaningful and quantifiable measure of test point value.

Let $\Delta P_{s_j}(\ell_j)$ represent the incremental change in the estimate of system reliability resulting from the addition of each test point to stage J. One of the following three results may occur:

1. $\Delta P_{s_j}(\ell_j) = 0$
2. $\Delta P_{s_j}(\ell_j) > 0$
3. $\Delta P_{s_j}(\ell_j) < 0$

For one, a zero value results since the only conclusion the user can draw is that his system is following the predicted curve determined at t_o and all decisions made about the system at t_o need not be changed. For either two or three, the resulting difference may cause the user to deviate from his initial set of decisions depending on the magnitude and sign of ΔP_{s_j}. (To speculate on what magnitude and sign of ΔP_{s_j} should cause a change in an initial decision at t_o depends on some predetermined set of intervals of values of ΔP_{s_j} and the set of alternative courses of action associated with each band. Since it is not the purpose of this study to establish these intervals or the courses of action, neither will be mentioned any further.) A non-zero value results in either two or three above and is determined by the configuration of test points and the ultimate benefit of allowing the system user to make better mission decisions.

Since it is the existence of a difference and not the sign of the difference that yields a value, the value of any stage j will be defined as $V_j = \left|\Delta P_{s_j}(\ell_j)\right|$ for any allocation of (ℓ_j) test points in that stage.

The objective of this program is to find an allocation of test points for the entire system; therefore, the problem of combining the stage values into some meaningful objective function which reflects the system operations still remains. Prior to t_o, the parameters of V_j are all determined. For $t \geq t_o$, all previously fixed parameters remain fixed and a new parameter s_j enters V_j. This parameter then determines the value V_j at some point in time t_1. However, the quantity s_j is probabilistic in nature and directly depends upon the success or failure of the units with test points. Thus, a more reasonable estimate of the value V_j of any test point allocation would be the increase in the expected value of V_j. For any stage J, the new stage value function is:

$$V_j = \sum_{s_j=0}^{\ell_j} \left| \Delta P_{s_j} \right| \binom{\ell_j}{s_j} P_j^* (t_1 - t_0)$$

Keeping consistent with the previous development of V_j and utilizing the assumption that all m stages are independent, a reasonable system value function is:

$$E(V) = \sum_{j=1}^{m} V_j$$

B. Characteristics of E(V)

The basis for utilizing E(V) in the following development is the guarantee that one may restrict attention to dominate configuration for matrices. This states that as later stages are added, there is no previously rejected combination of earlier stages with an allocated set of test points that might somehow fit better with new ones. The interpretation of what exactly is meant by a dominate configuration will be developed as various decision models are formulated. The remainder of this report primarily describes the formulation of a set of decision models which might be utilized in deciding where test points should be allocated in an optimum manner.

Since the function E(V) is the objective function for all but one of the following problem statements, it is worthwhile to describe some of its characteristics.

1. If $E(V_1)$ and $E(V_2)$ are the expected values for two disjoint sets of stages, then they combine to form a larger set whose expected value $E\left[V(E(V_1), E(V_2))\right]$ is uniquely determined.

2. E(V) is monotone increasing in the sense that $E(V_1) > E(V_1')$ and $E(V_2) > E(V_2')$ implies $E\left[V(E(V_1), E(V_2))\right] \geq E\left[V(E(V_1), E(V_2))\right]$ for all expected values $E(V_1)$, $E(V_2)$, $E(V_1')$, $(E(V_2'))$.

The statements and characteristics stated thus far establish the framework for the application of dynamic programming techniques to the solution of the allocation alternatives described below.

C. Decision Models

As implied previously, the goal of a test program is the development of (1) a test procedure which will provide the most success-failure state information for some predetermined constraint on dollar cost or, conversely; (2) a fixed amount of the same information for minimal dollar costs. Inherent to both of these problems is the concept of partial system testing to be carried out with some constraint on the quantity of test points to be utilized in the system.

For one above, cases I through IV, which follow, describe four possible constraints on cost with three variations on the quantity of test points available for each cost constraints. For two above, a case V will be developed which is very similar to one developed by Kettle (1).

Let:

C_L = the total cost for allocating L test points in the system

C_T = Constraint on available dollar cost for any allocation of test points in the system

c_j, c_k = Unit test point cost for a single unit in any arbitrary stages j or k.

Case I

$$\left[\underset{j}{\text{Max}}\,(c_j(n_j-1))\right] + \sum_{\substack{k=1 \\ k\neq j}}^{m} c_k n_k \leq C_T < \sum_{j=1}^{m} c_j n_j$$

This case is equivalent to the decision model of allocating a quantity of test points to a system with no constraint on dollar cost. The amount C_T specified allows for any complete allocation of the maximum of n-1 test points allowed under the partial testing criteria previously described. Under this case, any one of the following three conditions could exist:

1. Maximize E (V)

   subject to:

$$\sum_{j=1}^{m} \ell_j = L$$

$$\ell_j \leq n_j \text{ for all } j$$

For this problem, there are $\binom{n}{L}$ feasible solutions which could yield the maximum E (V). Further observation of E (V) reveals that in any one stage for any k there are $\binom{n_j}{k}$ k = 0, 1, ... n_j, arrangements of test points which yield the same value V_j. Utilizing the fact that L may vary from one to n-1 and the results of the previous observation, the range for search for a maximum E(V) is limited between a minimum of two and maximum of $(n_j + 1)$ terms per stage. Two terms occur when the number of test points L is equal to (n-1) and $(n_j + 1)$ terms occur whenever L is less than the quantity (n-minimum n_j).

With the above results, the dynamic programming techniques in Nemhauser[2] and several other tests may be utilized to find the optimum value of E(V) for any fixed L.

For a given group of test point allocations, defined over a set of stages, a configuration is said to dominate another if it yields a larger E(V) for any other allocation of the same quantity of test points.

2. Maximize E(V)

   subject to:

$$\sum_{j=1}^{m} \ell_j < L$$

where $\ell_j \leq n_j$ for all j

In developing the answer for fixed L above, one also has the ability to obtain a much broader and possibly more meaningful set of results as expressed in the inequality constraint on L. In this case, a decision may be made based on the results of maximizing for each value of ℓ $(1 \leq \ell \leq L)$ and observing a plot similar to that found in figure 2-2.

This technique is much more laborious than (1) but is the straight forward dynamic programming approach with the total number of terms per stage being $n_j + 1$ for the maximum value of L equal to n-1.

3. Maximize E(V)

   subject to: L variable

$$\ell_j \leq n_j$$

Introducing the constraint of L variable is the same as stating that the only sure decision is that the total number of test points to be allocated is some number less than or equal to n-1. Therefore all discussion of (2) above applies in this instance.

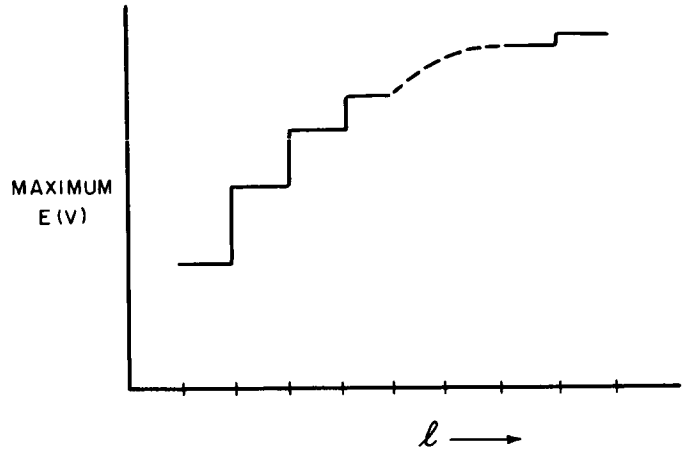


Figure 2-2. Maximum E(V) Vs Quantity of Test Point

Case II C_T Fixed

The total cost C_L of the allocation of test points is approximately equal to or exactly equal to some fixed limit C_T. Because the information gained from a test is a monotonically non-decreasing function of the number of test points, the allocation which yields the maximum E(V) will be as close to the value C_T as possible without exceeding it. The necessity for specifying C_L relative to C_T is caused by the fact that the C_j are allowed to be different between stages. There is no before the fact method of specifying that C_L will exactly equal C_T since the final configuration of test points has not yet been determined. Depending on the constraints placed on L that follow, the range of search of L for the maximum E(V) may be narrowed.

1. Maximize E(V)

   subject to:

$$\sum_{j=1}^{m} \ell_j \leq n-1$$

$$\sum_{j=1}^{m} C_j \ell_j = C_T \text{ (In the sense specified in the preceding paragraph).}$$

$$\ell_j \leq n_j \quad j = 1, \ldots, m$$

For the model, there arises a serious problem of defining the range of ℓ to search. To limit the quantity L to n-1 for large n still presents the range of possibilities from ℓ to n-1. In order that the range of feasible solution be specified more clearly, the following calculations are necessary.

(1) Order costs C_j in increasing (decreasing) value and denote these ordered costs as C_j'. Here C_1 is the ($\underset{j}{\text{minimum}}$ ($\underset{j}{\text{maximum}}$) C_j) and C_m' is the ($\underset{j}{\text{maximum}}$ ($\underset{j}{\text{minimum}}$) C_j).

(2) Reorder the indices of n_j to correspond with C_j' and denote them by n_k (k = 1 .... m).

(3) For the upper bound L_2 of ℓ, start by calculating the quantity $C_\ell = C_{\ell-1} + C_j'$ where:

C_ℓ = total cost of adding ℓ test points to a system

$C_{\ell-1}$ = total cost of adding ℓ-1 test points to the system

C_j' = cost of allocating the test point ℓ to the first or next unit in stage j.

Starting with $C_{\ell-1}$ equal to zero and $C_i = C_1$ of the first stage, accumulate the cost of adding single additional test points in that stage until n_1 is reached. When this is accomplished retain this total cost as C_1 and proceed to the next stage where C_j' now becomes the next higher unit test point cost C_2. Mathematically this may be represented by:

$$C_\ell = \left[\sum_{k=1}^{j'-1} n_k c_k\right] + r\, C_j',$$

where: r = 0 and j' = 1 initially and each time r equals n_j, r is reset to zero.

When $C_\ell \geq C_T$, the value of L_2 may be determined by calculating

$$\left[\sum_{j'=1}^{k-1} n_j\right] + r$$

(4) For the lower bound L_1 of ℓ, start with C_1 as the (maximum C_j) and repeat (3).

(5) For each ℓ in the range $L_1 \leq \ell \quad L_2$, find the Maximum E(V) which satisfies the previous cost constraint on C_L.

(6) Search (5) for the maximum of the maximums of the Maximum E(V)'s and thereby determine the value for L and the configuration of test points.

Depending on the method utilized in making computations there is a necessary condition that may be useful for eliminating configurations once L_1 and L_2 are determined.

Let C_j^* equal the minimum cost associated with adding one test point to a configuration defined by some value L between $L_1 \leq L \leq L_2$. A configuration and its associated L constitute the maximum solution if $C_T - C_L < C^*$ Proof: Suppose $C_T - C_L > C_j^*$ and the configuration associated with L yields a Maximum $E_L(V) = E_L(V)$. Formulate a new configuration by adding the test point associated with C_j^* to the one yielding $E_L(V)$ and thus get some new $E(V) = E_L(V) + \epsilon$ where ϵ is a positive quantity. But this means that $E_L(V) < E_L(V) + \epsilon \leq E_{L+1}(V)$ and therefore there exists a configuration of L + 1 test points which yields a greater E(V) than $E_L(V)$ and satisfies the constraint of $C_{L+1} \leq C_T$. Therefore only those configurations which have a cost in the interval $C_T - C_L \geq C_j^*$ are members of the set of maximum solutions of (5).

2. Maximize E(V)

   subject to:

$$\sum_{j=1}^{m} \ell_j \leq L$$

$$\sum_{j=1}^{m} c_j \ell_j = C_T \qquad \text{(As specified in 1)}$$

This case is, for computation purposes, identical to 1 above, with the exception that the upper bound L_2 is fixed by the constraint. Having found L_1, the problem may be solved in the same manner as before.

3. Maximize E(V)

   subject to:

$$\sum_{j=1}^{m} \ell_j = L$$

$$\sum_{j=1}^{m} c_j \ell_j = C_T \text{ (as specified in 1)}$$

Here the problem is merely specifying the $\binom{n}{L}$ configuration in the manner outlined in Case I-1 and make a search of this set to determine which ever satisfy the cost constraint. Having done this, it is only necessary to search these values of E(V) for a maximum.

Case III $C_L \leq C_T$

The total cost C_L of the allocation of L test points is less than or equal to some fixed limit C_T. There are gross similarities between the following three conditions on L and the previous cases.

1. Maximize E(V)

   subject to: L Variable

$$\sum_{j=1}^{m} c_j \ell_j \leq C_T$$

$$\ell_j \leq n_j \qquad j = 1, \ldots. m$$

With L variable, this constraint allows flexibility to the decision maker similar to that found in Case I-2. However, here the procedure utilized in maximizing E(V) is exactly the same as that found in Case II-1. In this instance, the maximum may be the same as that found in Case II-1 provided that the costs C_T are the same. The decision maker in this instance has the flexibility to look at his entire set of possible decisions for L together with their associated costs.

2. Maximize E(V)

   subject to:

$$\sum_{j=1}^{m} \ell_j \leq L$$

$$\sum_{j=1}^{m} c_j \ell_j \leq C_T$$

$$\ell_j \leq n_j$$

This case is for computational purposes identical to Case II-1, with the exception that the decision maker may request any number of plots from Case II-1 part (6) depending on the size of the increments utilized in specifying the values for C_T.

3. Maximize E(V)

   subject to:

$$\sum_{j=1}^{m} \ell_j = L$$

$$\sum_{j=1}^{m} c_j \ell_j \leq C_T$$

$$\ell_j \leq n_j \qquad j = 1, \ldots. m$$

With this constraint on L, this decision model yields the same solution for Maximum E(V) as that found in Case II-3 provided that the values for C_T are the same. In this case, a plot may be made of increasing cost C_T versus configuration for fixed L and E(V). This may be utilized for purposes of making tradeoffs based on decision makers preferences for specific types of configurations.

Case IV C_T Variable

The total cost C_T is allowed to assume any value necessary to maximize E(V). However, if it is assumed that the decision maker wishes to allow the test designer flexibility in his allocations, it is reasonable to place a limit on C_T equal to the maximum possible C_T of Case III. Having done this, the analysis of all of three types of constraints on L specified in Case III apply to Case IV.

Case V

For this case it is the desire of the decision maker to specify a level of E(V) he wishes to receive from a system. Here the problem is to extract a fixed amount of information for a minimal dollar investment. This case requires that the decision maker have some a priori knowledge of the relationship between the values of L and the E(V) received as a result of allocating L. This is the classical problem of setting requirements for a system test program without having quantitative results of how it might perform in reality. Thus a vicious cycle often arises, and the decision maker is all too often left to set his quantitative requirements based merely on qualitative information. This case is not the entire cure for this problem but merely sheds illumination on how one might attempt to begin to solve it. Returning to the formalization of this case it may be stated as:

$$\text{Minimize } C_{\ell}$$

$$\text{subject to: } E(V) \geq E$$

$$\sum_{j=1}^{m} \ell_j \leq L$$

$$\ell_j \leq n_j \qquad j = 1, \ldots\ldots m$$

where:

L is fixed at n-1.

C_{ℓ} is the cost of allocating ℓ test points to a system ($1 \leq \ell \leq L$)

E is the level fixed by the decision maker.

To get an estimate for E, it is proposed that the decision maker first disregard the cost problem and solve Case I-2 and make the plot specified there. Having done this, he may utilize this curve to specify the level of E that he might judge as being acceptable.

From this point, the minimization of C_1 to meet or better this level E may be handled with the generalized technique developed by Kettele.[1] Applying his techniques to this model the payoff function A is defined as:

$$A = \sum_{j-1}^{m} \sum_{s_j=0}^{\ell_j} \left| \Delta P_{s_j} \right| \binom{\ell_j}{s_j} P_j^*(t_1 - t_0)$$

The problem of generating a dominating sequence by utilizing the algorithm is also alleviated by the fact that the configuration of units is fixed and the problem is placing test points on these units. For discussion of the remainder of the algorithm see the above referenced paper.

2-5 CONCLUSION

A. Requirements

In order that test design be developed in the manner described, the following parameters must be specified.

1. The fixed configuration of units to be analyzed, i. e. the system.
2. n_j = the number of units in each stage j.
3. k_j = the least number of units required for stage success.
4. L = the maximum number of available test points.
5. C_T = the maximum amount of dollars available for test program.
6. Decision criteria.
7. Probability functions $P_j(t)$ for each stage
8. Time of test t_1.

B. Limitations

If one can make the assumption that the operation or failure of every stage is statistically independent of the operation or failure of all subsystems outside the stage, the present technique of analysis may be extended to fairly broad class of systems of predominately serial configuration. These systems may include feedback loops, feedforward loops, diverging branches and converging branches as described in great detail in Nemhauser's book[2].

Regardless of the configuration of the system model, the technique developed here can handle any number of test points that a single unit may require. Here, the implication is that the total number of test points necessary to verify the success or failure of a unit be grouped as a unit test point. Thus no partial allocations to a unit are permitted. If only partial success or failure information can be obtained on a unit then each element of the E(V) expression must be rederived.

Another important aspect pertaining to the test points is that the cost functions associated with adding test points within the stages must be monotomically non-decreasing with the addition of each test point.

Because the magnitude of the computations for practical values of n is large, it is recommended that before solution of any realizable system be attempted, the entire allocation procedure be implemented as a computer program for solution on a large scale digital computer.

BIBLIOGRAPHY

1. Kettele, J.D., Jr., "Least Cost Allocations of Reliability Investment", Operations Research 10, pp. 249-265, 1962.
2. Nemhauser, G. L., Introduction To Dynamic Programming, John Wiley and Sons, Inc., New York, N. Y., To be published in 1966.

SECTION 3
IMPLEMENTATION OF AN ADAPTIVE VOTER

3-1. INTRODUCTION

A. ADAPTIVE VOTER BACKGROUND

One of the most effective practical techniques for introducing redundancy into a digital system is illustrated diagramatically in figure 3-1. The system is divided into a group of identifiable subsystems, which are replicated two or more times and interspersed with redundant voting circuits. A typical voting circuit examines the set of nominally identical signals at its inputs, and, based on this input information provides an estimate of what the correct output signal from the subsystem set should be.

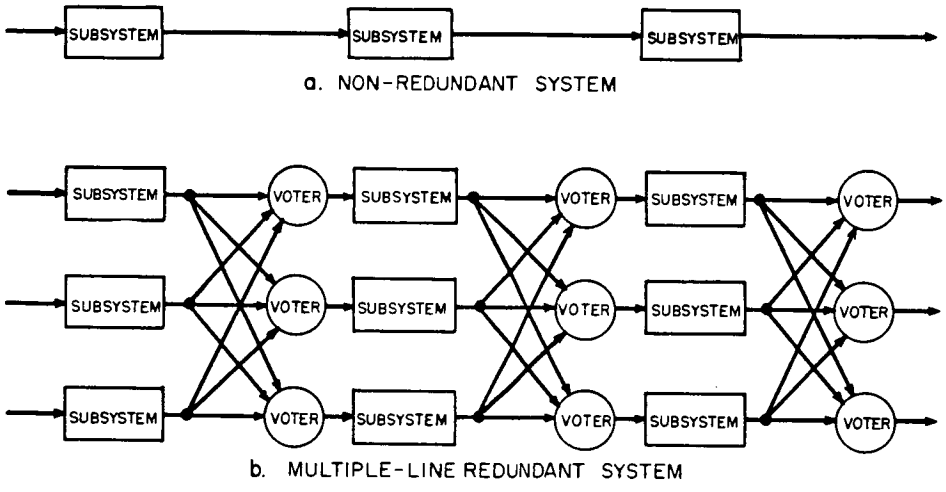


Figure 3-1. Segment of a Typical Redundant System

The most common restoring network is a "majority voter". In order to make a correct estimate of the output for a set of subsystems, the majority voter requires that no $\frac{n+1}{2}$ * of

*n is the number of inputs to the restoring network, i.e., the "order of redundancy".

its inputs be failed to the same state. Although this network is effective when n = 3, it is very inefficient when n> 3. This ineffectiveness exists because the percentage of the redundant subsystems which must be operating correctly to permit a correct vote is undesirably large. The relative inefficiency of the majority voter can be seen by comparing the reliability vs time curves shown in figure 3-2. The lower of the two curves characterizes a 35 input majority voter. The upper curve represents the reliability of a nine input restoring circuit which has the capability to estimate the correct subsystem output as long as any two of its inputs are consistently correct.

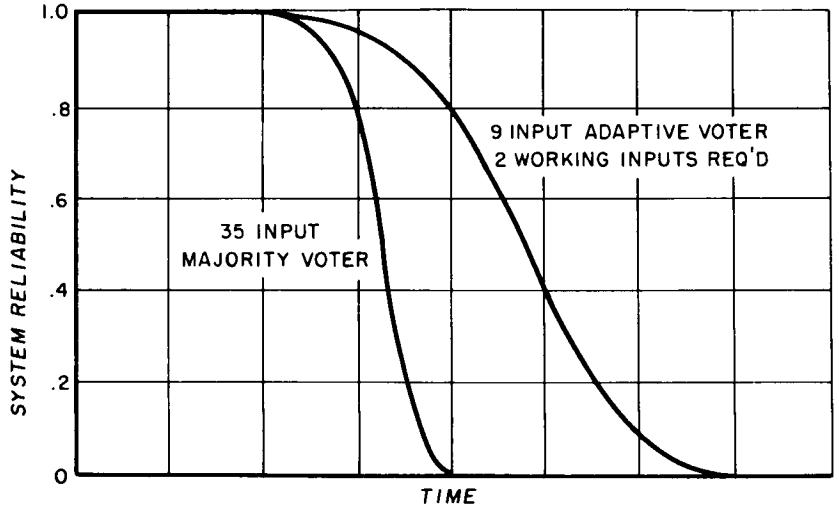


Figure 3-2. Reliability Vs. Time Curves for Two Voters

In order to realize the advantages of voters which can operate correctly with less than a majority of correct inputs, some means of "deweighting" faulty inputs must be provided. In studies of Stanford Research Laboratories and the Westinghouse Research Laboratories, Dr. W. H. Pierce has devised several schemes for optimally weighting inputs as a function of their past history of errors.[1] The general "adaptive voter" configuration which he has proposed is shown in figure 3-3.

As part of this Failure Free Systems study, Westinghouse has been attempting to bridge the gap between Pierce's theoretical studies and the construction and use of a practical adaptive voter. This effort has revealed several important items concerning the adaptive voting techniques which Pierce apparently did not consider in his general study. From his relatively abstract viewpoint many of these items are relatively unimportant, but in relation to certain feasible implementations, these items may be the dominating factors in the final design.

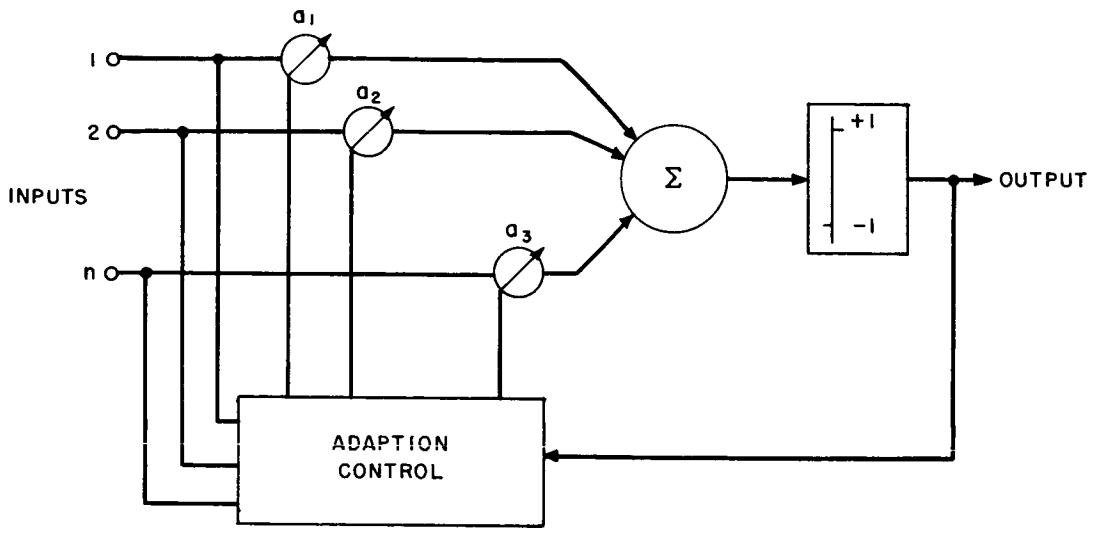


Figure 3-3. Adaptive Voter Configuration

For example, Pierce assumed that the probability of an error occuring in any input channel was symmetrical, i.e., erroneous extra ONES and extra ZEROS were equally likely, regardless of the past history of the channel. In view of the voting schemes which he has proposed, this assumption is not particular significant. However, at least one very simple adaption scheme is known to exist in which the asymmetry of errors is critical to the adaption process. The simplicity of this latter adaption scheme implementation is achievable because the specific characteristics of one of the available weighting devices can be exploited. To implement this scheme, a circuit can be devised such that if a series of asymmetrical errors occurs at an input regardless of the ZERO or ONE orientation, the voter will deweight to its minimum value (i.e., maximum impedance). This scheme is particularly attractive because the circuitry required to monitor the existing value of an input "weight" is not required and no complex weighting function must be computed for each input.

As a second example, Dr. Pierce did not consider the "cost" of making a non-optimal decision. In the above example, the proposed adaptive voter would probably not achieve optimal vote weights, but if the cost of using non-optimal weights is sufficiently low, the simplicity of the implementation will more than offset this cost.

B. VARIABLE WEIGHT COMPONENTS

Inherent in the basic design of an adaptive voter is the requirement for an electrically variable conductance (or weight) device which performs integration and displays relatively permanent memory of the established weight. These special characteristics have stimulated considerable effort toward the development of suitable adaptive components. The devices of this type which have been proposed generally utilize phenomena involving atomic translation or rotation.

During the first phase of this contract, an extensive survey of the more promising of these devices was made. The results of this survey are described in detail in Appendix 3 of reference 2. The devices considered in this survey include three which exploit electrochemical effects and four which utilize magnetic domain phenomena. Briefly, there are:

1. Electrochemical Devices

a. The Memistor

The memistor consists of a sealed plating cell containing an electrolytic bath, a resistive substrate upon which metal is deposited and a metal source electrode. The conductance of the device is changed and stored by plating or stripping between two signal electrodes and a third control electrode.

b. The Solion

The solion is a fluid-state device which functions by controlling and monitoring a reversible electrochemical redox reaction, a chemical reaction in which oxidation and reduction occur simultaneously. By controlling the charge transferred between the two input electrodes, a change in conductivity proportional to the integral of the input current may be obtained between two output electrodes.

c. The Mercury Cell

The mercury cell device consists of a capillary tube filled with two columns of mercury separated by a "gap" or bubble of transparent aqueous electrolyte of metallic salt. A d-c control signal is used to electroplate mercury across the gap, thus causing the bubble of electrolyte to move. Read-out can be accomplished through any of several visual or capacitance detection methods.

2. Magnetic Devices

Various techniques have been suggested for providing variable gain and non-destructive readout with magnetic devices. The phenomena utilized in such devices is based upon the ability of magnetic materials to store a remanent flux which is sensed in a non-destructive manner. Suggested devices provide the capability for a partial switching of

magnetic domain under a volt-second impulse as the basic incrementing source. Suitable magnetic materials include ferrites and tape wound cores which are characterized by a square hysteresis curve. Most of the devices to be described utilize the same basic type of incrementing technique and differ primarily in the manner by which the stored flux is sensed.

a. MAD Integrator

A diagram of a typical multi-aperture device is shown in figure 3-4. Initially the flux around the main aperture is set to cause saturation. A momentary reversal of the magnetizing force driving the main aperture will cause a partial reversal of the flux. The amount of flux reversal is determined by the magnitude and duration of the drive and the value of the hold current. The purpose of the hold winding is to retain a portion of the core saturated in the original direction of magnetization and thereby assure partial switching of the flux. The amount of flux alternately switched around the small aperture is then proportional to the flux which has been switched around the main aperture. The output voltage will consist of a signal whose voltage integral is proportional to the amount of flux trapped in the common area between the two flux paths.

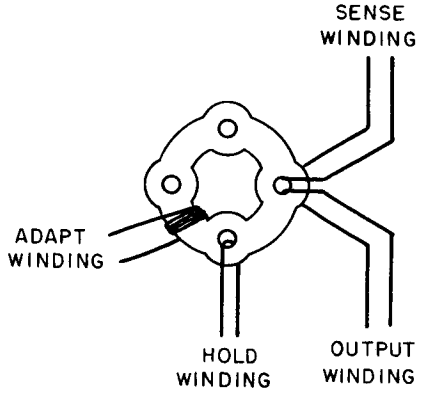


Figure 3-4. Multiple Aperture Device (MAD)

b. Orthogonal Core Integrator

In this device the magnitude and direction of a stored flux is sensed by applying a magnetic field orthogonally to the direction of stored flux. This causes the remanent flux vector to rotate, generating a voltage proportional to its rate of change and hence its magnitude. At the termination of the read drive the flux vector returns back to its original preferred orientation by virtue of domain elasticity. The flux level stored in the core is altered by pulsing input winding. The output signal consists of either positive

or negative pulses depending upon the direction of the stored flux, with an amplitude proportional to the magnitude of the remanent flux.

c. Second Harmonic Integrator

A second-harmonic generator normally consists of a pair of tape wound cores driven from an r-f sinusoidal power source. The output winding is arranged so that the fundamental component of drive voltage cancels out, leaving a second harmonic distortion voltage proportional to the remanent flux in the cores. The output is detected by passing a single sense winding through the cores in the opposite direction.

By passing a direct current through the same sense winding the remanent flux level may be altered. Due to an interaction between the d-c adapt current and the RF drive the rate of change of the remanent flux with respect to the adapt current is constant and reversible.

d. Magnetostrictive Integrator

The direction and magnitude of the net remanent flux in a magnetostrictive core may be sensed if the core is excited mechanically. A simplified scheme for implementing a magnetostrictive storage system employs an ultrasonic delay line to excite several magnetostrictive toroids. Driving source for the sonic delay line is a piezoelectric transducer. Input to each of the toroids is provided by means of narrow width pulses through a separate write coil concentrically with the read coil. If the frequency and rms amplitude of the stress wave is maintained at constant value, the open circuit output of the read coil is approximately proportional to the flux stored in the individual toroids.

C. COMPONENT EVALUATION

The following general observations were made during the survey.

The magnetic devices with their known sensitivity to temperature stress appear to offer the least hope for providing analog memory with long term stability. The requirement for providing carefully controlled incrementing with relatively large drive currents coupled with the small output signals and associated amplification appears to dictate an imposing amount of peripheral circuitry. The degradation in reliability as a result of this complexity represents a liability which makes practical application doubtful for redundant systems.

The electro-chemical devices, especially the memistor and solion in their present state of development, appear to be plagued by a number of stability problems. The memistor with its dependence upon an electroplating process which is not widely understood, chemical impurities and dimensional imperfections will require considerable refinement before application becomes practical. In addition the requirement for sensing the state of the device with an a-c signal makes circuit implementation rather awkward.

Solions appear to be somewhat more practical if size is not an important consideration. It has been reported that the Rome Air Development Center is constructing an adaptive learning machine (CHILD) which uses 1080 solions. With its dependence on the chemical equilibrium of a redox system and the precise construction required to achieve stability the solion presents several challenging design difficulties. The requirement for providing an isolated battery cell between the input and shield electrodes imposes a practical encumberance on a system design which requires a large number of solions.

At the completion of the survey, and at the beginning of this phase of the contract, the mercury cell integrator with photoelectric readout appeared to offer the most attractive approach because of its simplicity, stability in time and general compatibility with conventional circuitry. Because the output is essentially a variable resistance proportional to the integral of the control input current, the device can be easily interfaced with more standard circuitry.

3-2. PROJECT DEFINITION

The objective of this project has been to further investigate the feasibility of constructing adaptive voters of the type proposed by Dr. W. H. Pierce. To accomplish this objective, a program was established to investigate the availability of the various electrically variable weighting devices and to construct and test a model of an adaptive voter.

The accomplishment of the first of the above subgoals showed that, as a result of a development program conducted by the Department of Defense, one version of the mercury cell integrator was available as a manufactured item. The particular model which was available has photoconductive readout. This corresponds to the general type recommended for use in adaptive voter application by the previous survey.

In concurrence with this finding, several of the mercury cell integrators were procured for evaluation. The remaining effort on this task has been concerned with the design and construction of an adaptive voter which employed these devices in an operational model. The specific purpose in designing and constructing this model was to determine the actual usefulness of such devices in an adaptive voter configuration.

3-3. MODEL DESCRIPTION

The breadboard model of an adaptive voter which has been constructed for this purpose, consists of a hybrid combination of analog and digital circuitry and an on-line general purpose computer. The computer generates simulated input data for the voter and performs the feedback adoption control function inherent to the operation of the voter.

In the first portion of this dual role, the computer has been programmed to inject into a random data stream a variety of different error patterns. By selecting the proper error

pattern the investigator has the capability to modify the statistical properties of the voter's input data to fit the requirement of almost any desired test. The use of the computer to perform the feedback control function offers the investigator an additional degree of flexibility. To statistically test any proposed adaption scheme, a relative simple subroutine must be prepared for the computer and inserted into the existing main program. To perform a test, the investigator needs only to supply the computer with the particular adaption subroutine to be considered and the information required to establish the simulated data characteristics.

The portion of the voter which has been implemented as actual circuitry consists of digital control equipment which increments the variable input weighting devices, the analog weighting devices, and output threshold and squaring circuits. Mercury cell integrators with photo-conductor readout were chosen as the input weighting devices. This choice was based on the results of the previous survey of weighting elements, the lack of any more promising new elements, the further development of these devices by the Department of Defense and the availability of the refined elements as G. F. E. from the developers.

The following paragraphs describe the adaption scheme used in the model and the specific nature of the computer programs and circuitry used to implement the model.

A. ADAPTION SCHEMES

1. Weight Values

Any binary decision element is a generalization of the majority organ introduced by von Neumann.[3] The binary generalization of this device is the adaptive voter shown in figure 3-3. The binary numbers +1 and -1 are used as input signal levels both for their conceptual simplicity and for their symmetry. The n input bits, each of which is +1 or -1, are individually weighted by a gain factor, a_i. The resulting weighted signals are summed, and then sent to a threshold element which gives an output of +1 when the sum is positive, and an output of -1 when the sum is negative. In equations,

$$x_i = i^{th} \text{ input digit} \qquad i = 1, 2, \ldots n$$

$$\text{Output of summation} = a_o + \sum_{i=1}^{n} x_i a_i$$

where a_i is called the i^{th} vote weight

$$\text{Device output} = \text{signum} \left[a_o + \sum_{i=1}^{n} x_i a_i \right]$$

The adaptive voter reduces to the majority organ when a_o equals zero and all other a_i equal one. *

*One version of the voter weights each input and then normalizes the sum of the weighted inputs by dividing by the sum of the vote weights.

The primary disadvantage of the majority-vote technique is that a consistently reliable minority could be outvoted by an unreliable majority. This limitation can be overcome by the adaptive vote-taker of figure 3-3 provided the reliable inputs are given heavier vote weights and the unreliable inputs are given lighter vote weights. When errors in the inputs are independent, the vote weights, a_i, can be chosen so that the output of the adaptive voter is actually the binary state which is more likely to be correct. If the error probability of the i^{th} input is denoted by λ_i, then the vote weights which give the optimum (i.e., more probably correct) output are:

$$\text{Output} = \frac{x_o a_o + \sum_{i=1}^{n} x_i a_i}{\sum_{i=0}^{n}}$$

where $x_o a_o$ is the bias term, x_i is the i^{th} input and a_i is the i^{th} vote weight.

$$a_o = \log \frac{\text{a priori probability of } +1}{\text{a priori probability of } -1}$$

$$a_i = \log \frac{1-\lambda_i}{\lambda_i} \qquad i = 1, 2, \ldots n$$

If an input is completely random noise, i.e., $\lambda_i = 1/2$, then the optimum a_i is zero. As λ_i decreases the optimum a_i monotonically increases. Note that λ_i greater than 1/2 require negative inputs. An always wrong input, for instance, would provide always right information if its output were inverted.

2. Schemes for Estimating Input Reliabilities

The following methods of adjusting the vote weights in a decision element have been proposed by Pierce.[1] All use $\hat{\lambda}_i$, the estimated error probability of the i^{th} input, to set the next vote weight to $a_i = \log (1-\hat{\lambda}_i)/\hat{\lambda}_i$.

Adaption Method 1. $\hat{\lambda}_i$ is obtained from an analysis of the circuit which produces x_i. This straightforward open loop adaption requires no data analysis.

Adaption Method 2-A. $\hat{\lambda}_i$ is obtained periodically from M comparisons of x_i with an externally supplied correct answer. This method is useful for the initial adaption

of new circuits or routine preventive-maintenance adaption, occasionally using check problems. The analysis required to pick M is similar to, and simpler than, the analysis which will be given for Method 2-B.

Adaption Method 2-B. $\hat{\lambda}_i$ is obtained periodically from M comparisons of x_i with the output of the decision element, treating the output of the decision element as if it were always correct. The analysis of this method in Pierce's work verifies an important concept: A decision element can maintain its reliability by feeding back its own output in order to judge the reliability of its inputs. This analysis justifies feeding back the output in other adaption methods which are well suited to implementation but poorly suited to mathematical analysis.

Adaption Methods 3-A and 3-B. These methods use Widrow's Adaline[4] to adjust the vote weights. Reference 3 contains a description of the adaption procedure, and shows that at equilibrium the vote weights are proportional to the hyperbolic sine of the optimum vote weights. Reference 4 and 5 give circuit implementations used for pattern recognition. These methods are of conceptual interest because they are based upon surface searching (6) and practical interest because they may offer some reduction of components required for implementation.

Adaption Methods 4-A and 4-B. The only memory required for the i^{th} input is the present value of $(1-\hat{\lambda}_i)/\hat{\lambda}_i$, the log of which is a_i. The next value of $(1-\hat{\lambda}_i)/\hat{\lambda}_i$ is incremented by $f(\hat{\lambda}_i)$ if x_i agrees with the comparison signal, and by $g(\hat{\lambda}_i)$ if x_i disagrees with the comparison signal. The comparison signal is an external answer (Method 4-A), or the output of the decision element (Method 4-B). Suitable functions are $f(\hat{\lambda}_i) = K$, $g(\hat{\lambda}_i) = K(1-\hat{\lambda}_i)/\hat{\lambda}_i$, where K is a positive constant considerably smaller than one. These methods are relatively simple to implement with electrical circuits.

Adaption Methods 5-A and 5-B. A pulse is put into a linear low-pass filter for every disagreement between x_i and the comparison answer, which is an external correct answer for Method 5-A or output of decision element for Method 5-B. If output of filter exceeds θ, $a_i = 0$. Otherwise, $a_i = 1$. Excessive regularity in the statistics of the correct answer may make it desirable to permanently keep $a_i = 0$ if the filter output ever exceeds θ. These methods are simple to implement and very effective against catastrophic failures.

The model of the adaptive voter which has been constructed during this program employs Adaption Method 2-B. This choice was made primarily because Pierce had considered this method in more detailed analysis and experiments could be designed to check the performance of model against his theoretical work. Although this comparison was not a part of this phase of the study, it may be desirable to perform the comparison at a later date. Again it should also be noted that the adaption scheme used in the model can be easily changed by changing the portion of the computer program which performs the feedback control function.

B. THE COMPUTER PROGRAM DETAILS

1. The Simulated Input Signal Generation

To simulate the correct signal which would appear at each voter input if no errors were present, the computer generates a random series of binary ZERO's and ONE's. The average ratio of ONE's to ZERO's in the series is under the control of the input variables. This input control or the signal characteristics allows the investigator to simulate signals having different a priori probabilities of occurrence of either ONE's or ZERO's.

The simulated input signal to any particular voter input is a function both of the correct signal described above and a second random variable which reflects the probability of an error occurring on that input channel. The probability of error is determined by the investigator according to environmental conditions he wishes to simulate. In determining the probability of error, the investigator has the option of setting different error probabilities for any input as a function of the binary state of the correct signal. Thus, a particular input channel may display no errors, only erroneous ONE's, only erroneous ZERO's or a combination of the latter two, depending on the error probabilities associated with that channel.

Again as Pierce has noted[1], one type of error in binary systems is caused by thermal or other noise which occurs randomly in time. Another type of error occurs randomly in space throughout the equipment, persisting from operation to operation; an example is catastrophic failure. Errors, therefore, can occur randomly in time or space. As the program is written, both time and space errors can be simulated with equal ease.

2. The Feedback Adaption Control Function

In addition to the peripheral function of input simulation, the computer also acts as an integral part of the adaptive voter by performing the adaptive control function. In this role the computer continuously monitors each input signal and compares it to the output of the voter. During each operating period, a count is kept of the number of errors (i.e., disagreements with the voter output) observed on each input channel. At the end of M bits of transmission, the ratio of error bits to total transmitted bits is calculated for each input. This ratio is then used as the estimate of the error probability, $\hat{\lambda}_i$, for the associated channel. From this estimate, the weight (a_i) for the channel is computed as described for Adaption Method 2-B.

Once the desired weights (or changes in weights) have been determined, the signals required to initiate any corresponding changes in the actual weighting devices are generated and sent to interface control circuitry. The computer allows a predetermined time in which to make the changes and then reinitiates the entire operation cycle.

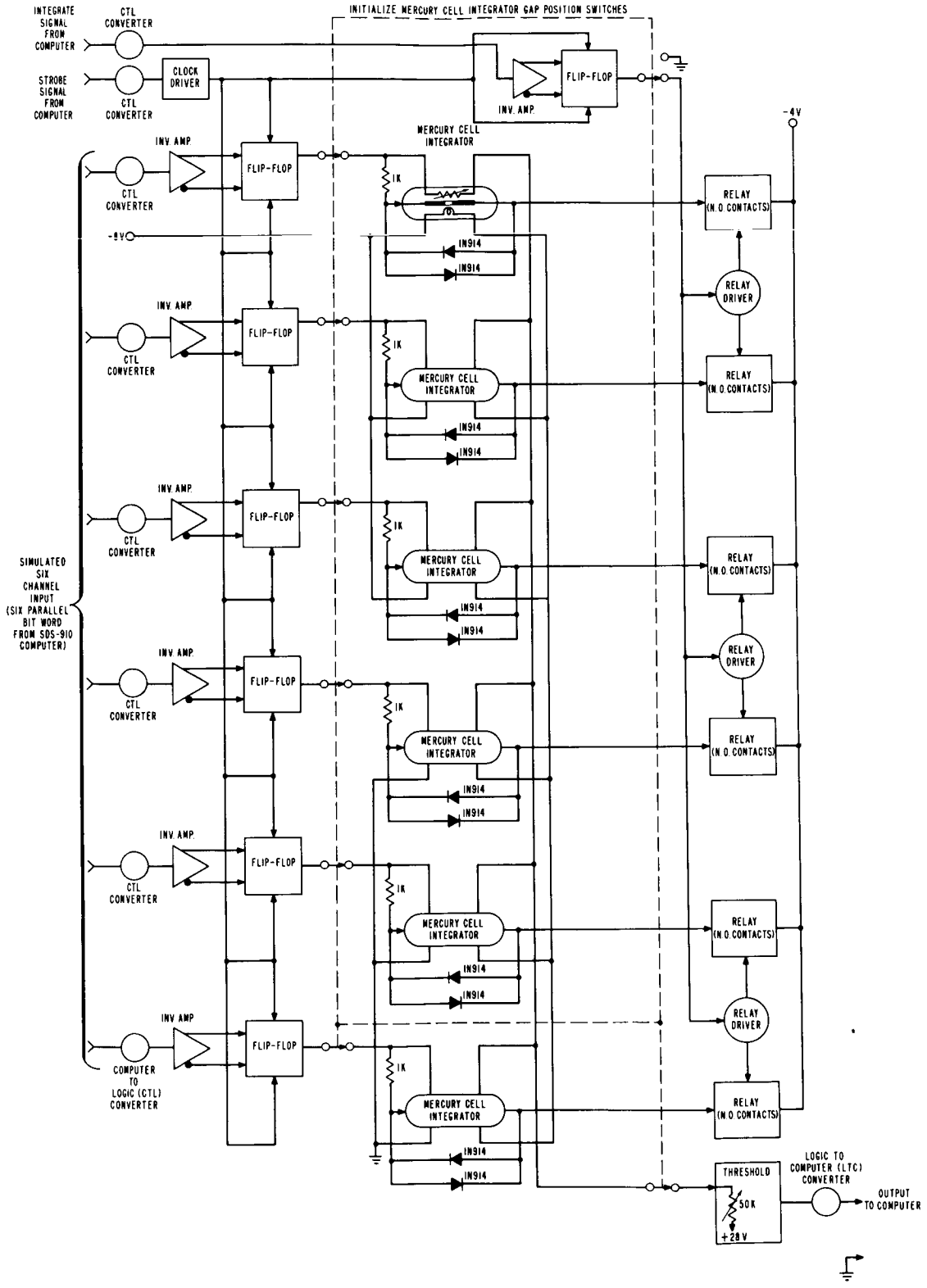


Figure 3-5. Adaptive Voter Breadboard Schematic Diagram

C. CIRCUITRY PORTION OF THE ADAPTIVE VOTER MODEL

The diagram shown in figure 3-5 includes both the actual adaptive voter circuitry and the interface equipment required to connect the SDS 910 computer with the voter circuitry. As the diagram shows, the particular version of the voter which has been constructed assumes an order of redundancy of six, i. e. six nominally identical input channels.

1. The Voter Circuitry

The variable input weights of the voter are Curtis model 251 mercury cell integrators (figure 3-6). As described previously, this device consists of a capillary tube filled with two columns (electrodes) of mercury separated by a gap of aqueous electrolyte. A d-c input signal electroplates mercury across the gap at a rate which is a direct function of the amplitude of the input signal. The movement is in time within certain ranges of photoconductor resistance; it is reversible without hysteresis and it is stable over long time intervals.

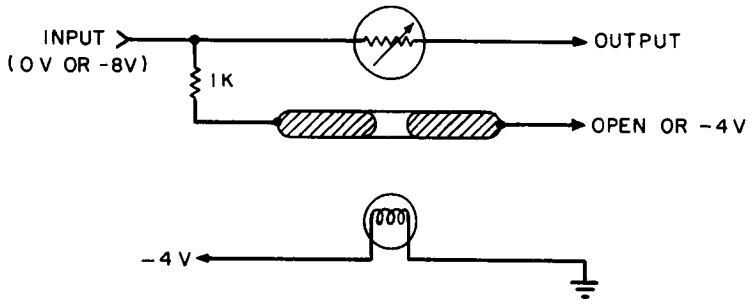


Figure 3-6. Mercury Cell Integrator

The variable weight as a function of the current-time integral is obtained by varying the quantity of light available to excite the photoconductor utilizing the gap as a light shutter. Initial photoconductor resistance is determined by the type of photoconductor, light intensity and initial alignment of the gap between the light source and the photoconductor.

One of the most outstanding characteristics of the model 251 mercury cell integrator is the lack of uniformity between the resistance versus time curves for individual units. The two curves shown in figure 3-7 illustrate the extent of the variation which might be found between two of the integrators. This lack of uniformity between individual units tends to cause a difference in input weights even though no difference should exist. To overcome this problem, separate adaption subprograms for each input could be written for the computer feedback loop, although the actual fabrication of such complex adaption

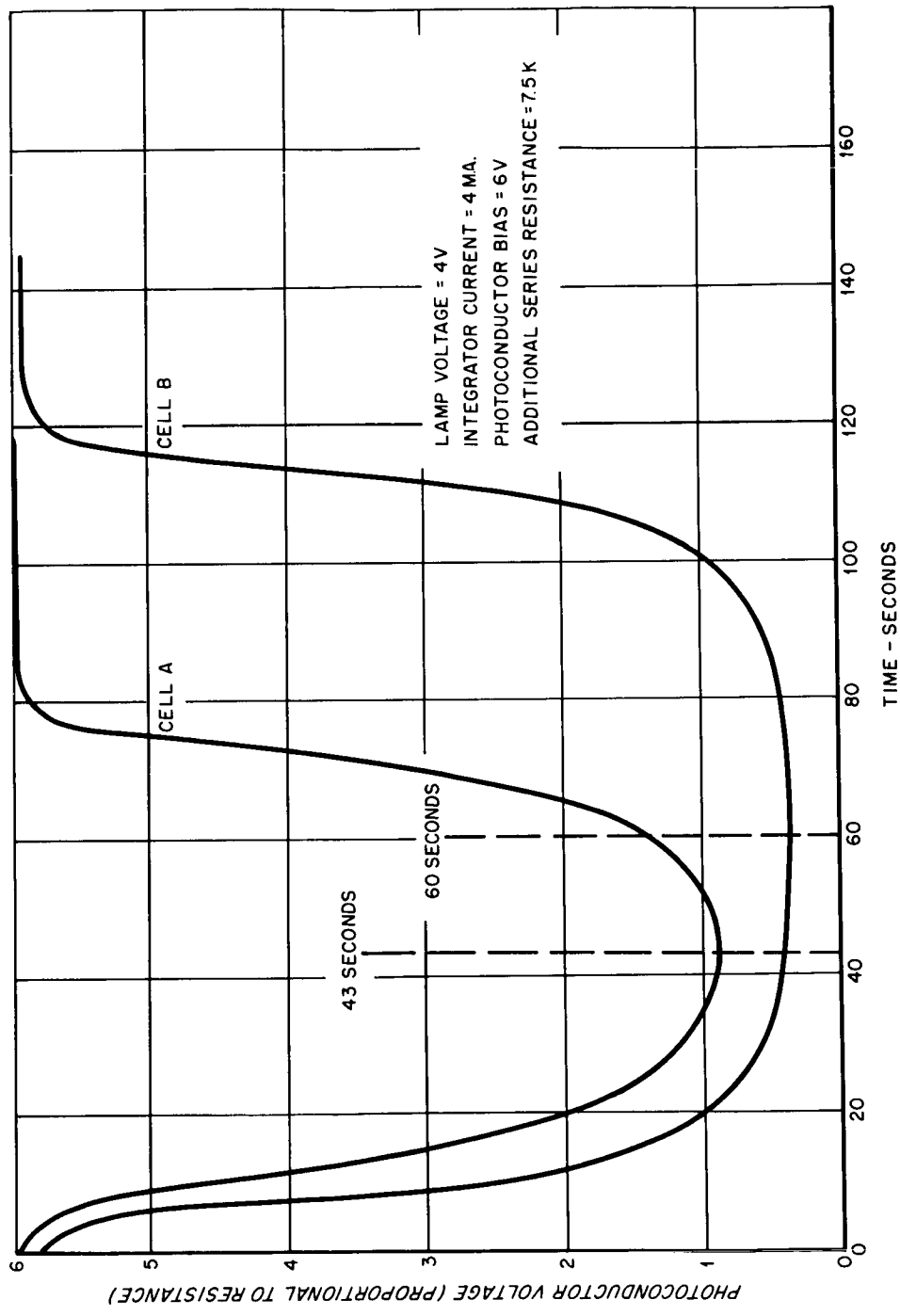


Figure 3-7. Characteristic Curves for Two Mercury Cell Integrators

equipment would be highly unlikely in practical applications. As a result of this, an estimated average characteristic curve for six relatively similar mercury cell integrators was generated and was assumed to describe their characteristics. (See figure 3-8). This curve was found to closely approximate a log function; therefore, the log function was selected to represent the characteristic curve in the computer feedback control program.

In using this particular model of the mercury cell integrators several precautions are necessary for proper operation:

a. The integrator current should not exceed 5 milliamperes
b. The integrator should be protected with parallel back-to-back silicon diodes to prevent the voltage across the integrators from exceeding 0.7 volts
c. The lamp voltage should be less than 5 volts
d. The photoconductor bias should be less than 60 volts
e. A minimum of 5 seconds should be allowed for the resistance to settle after an integration period because the integration current causes a transient change in the electrolyte light refraction.

When the voter is transmitting simulated data, the mercury gaps in all voters remain stationary. As a result, the vote weights remain fixed. Thus, signals appearing at the voter inputs are passed through a resistive summing network to a threshold circuit* (see figure 3-9). The latter emits a binary ONE or ZERO depending upon whether the weighted average of the inputs reaches or fails to reach the preset threshold level.

During the adaption period of operation, an integrate signal from the computer sets the flip-flop* (figure 3-10) in the integrate line. This, in turn, stimulates the relay drivers* (figure 3-11) which energizes the relays. The relays* then pass an integrate current through the mercury cell integrators. The time integral of the current determines the desired change (if any) in gap location; hence, the weight of each integrator. During this adaption period, the state of the flip-flop* in the interface circuitry associated with each input channel determines the direction in which the mercury gap in the integrator of that channel will move.

During the adaption portion of the operation, the photoconductor resistance can increase in some cells and decrease in others until the output of the threshold circuit is correct for a given input. The threshold can be manually set to establish a bias which fixes the number of inputs necessary to produce an output.

*Note: All circuits denoted by an asterisk were supplied by Westinghouse as company owned test equipment at no direct cost to this contract.

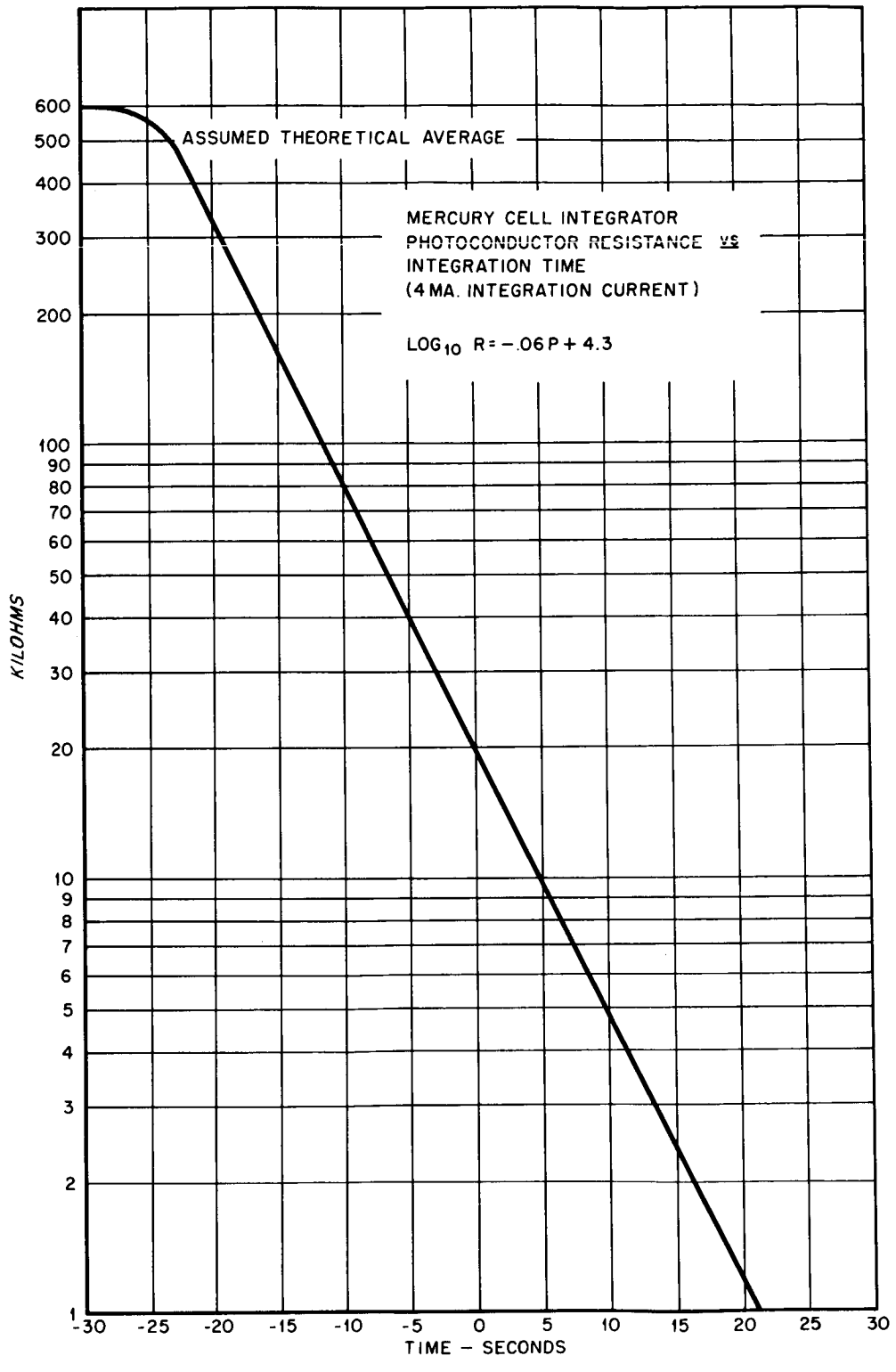


Figure 3-8. Theoretical Average Characteristic Curve

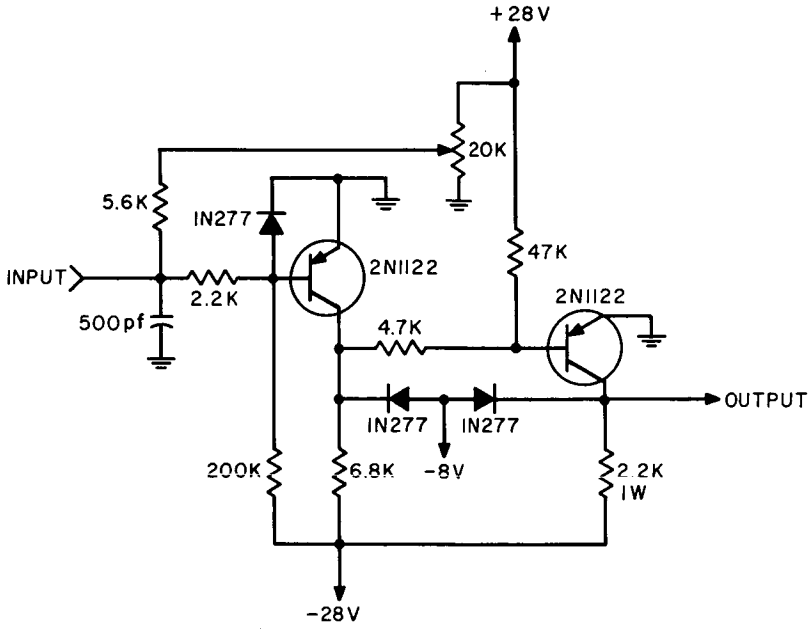


Figure 3-9. Threshold Circuit

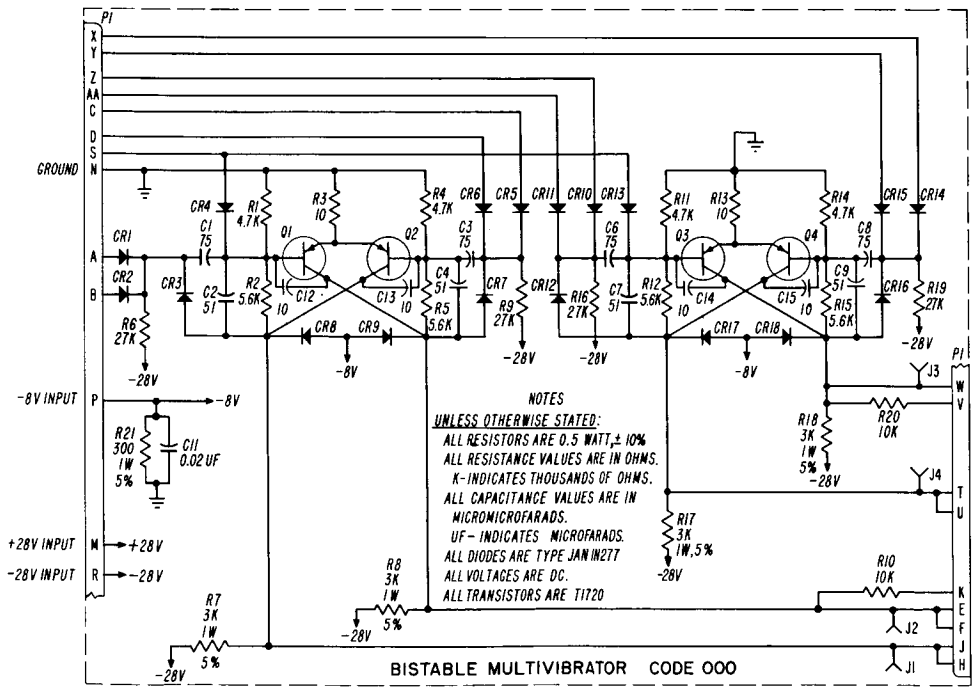


Figure 3-10. Bistable Multivibrator

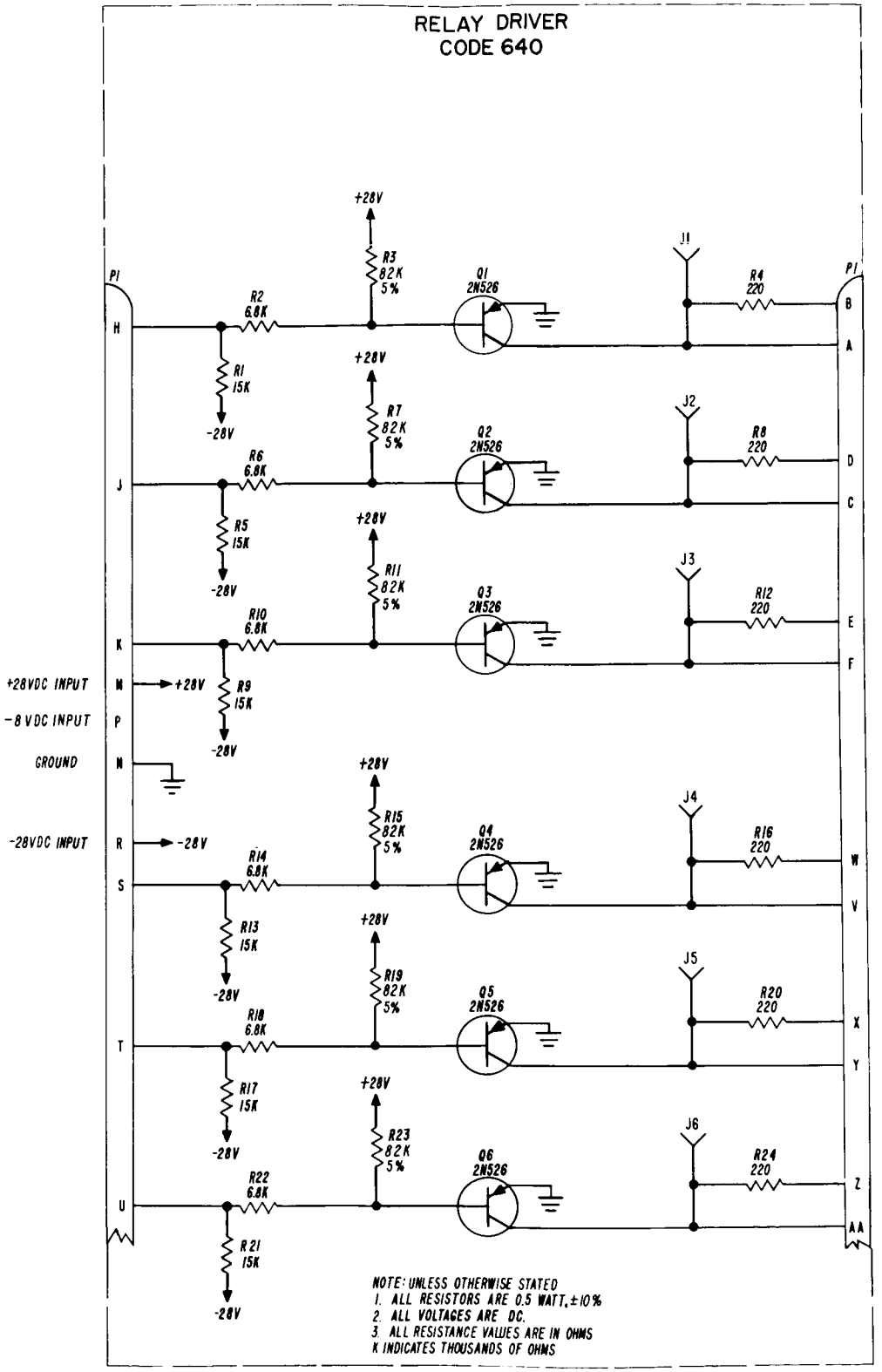


Figure 3-11. Relay Drivers

2. The Interface Circuitry

The normal output channels of the SDS-910 are intended to provide signals to external equipment only at certain intervals. These intervals are determined by an internally generated strobe pulse which appears on a special output channel. Between the occurrences of strobe signals, the integrity of the d-c levels of the other output channels is not maintained. Because the integration control signals for the mercury cell integrators must be maintained for relatively long periods of time (several seconds), special circuits must be used to convert the SDS-910 output signals to d-c type signals. To accomplish this conversion a standard flip-flop circuit (figure 3-10)has been inserted between the voter and the computer output terminal in each channel. For the same reason the flip-flop was also inserted in the integrate control line. Each flip-flop is fed the strobe signal through the clock driver* (figure 3-12). The use of standard inverter amplifiers* (figure 3-13) preceding each S-R flip-flop permits the use of "single-rail" data transfer from the computer. This avoids the necessity for the generation of both the signals and their complements by the computer.

Differences between the logic levels of the SDS-910 computer and the breadboard circuitry also required the use of the Computer-to-Logic (CTL) and the Logic-to-Computer (LTC) converters shown in figure 3-14 and 3-15 respectively. The level converters allows the various interface signals to meet the following requirements.

Computer "ONE" Output -	+6.5v. to +9.5v.
Computer "ZERO" Output -	0.0v. to +0.6v.
Computer "ONE" Input -	+5.0v. to +20.0v.
Computer "ZERO" Input -	-2.0v. to +2.0v.
Voter "ZERO" -	0.0v to -2.0v.
Voter "ONE" -	-6.0v. to -8.0v.

*Note: All circuits denoted by an asterisk were supplied by Westinghouse as company owned test equipment at no direct cost to this contract.

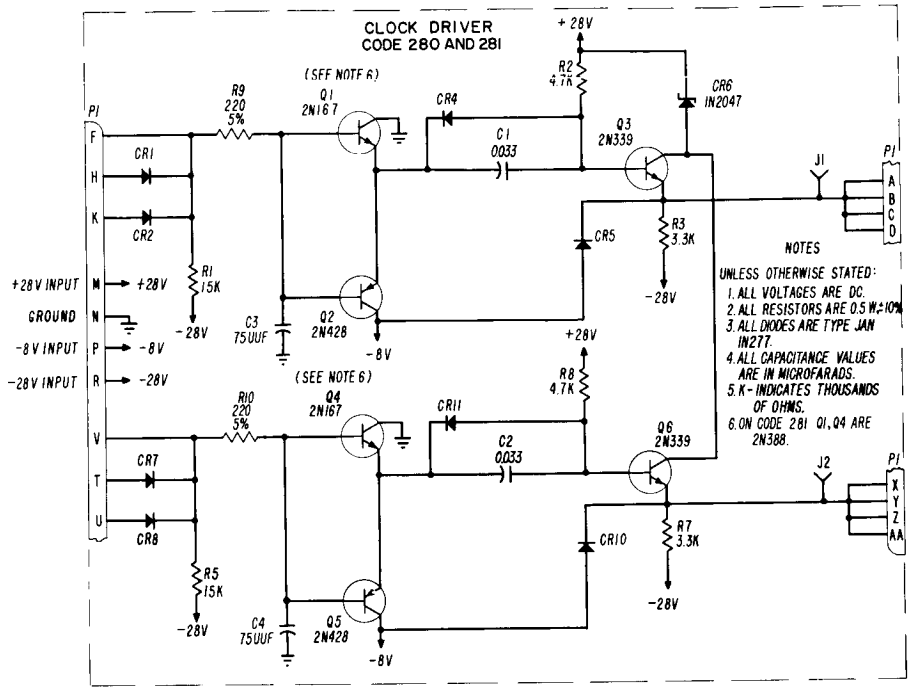


Figure 3-12. Clock Driver

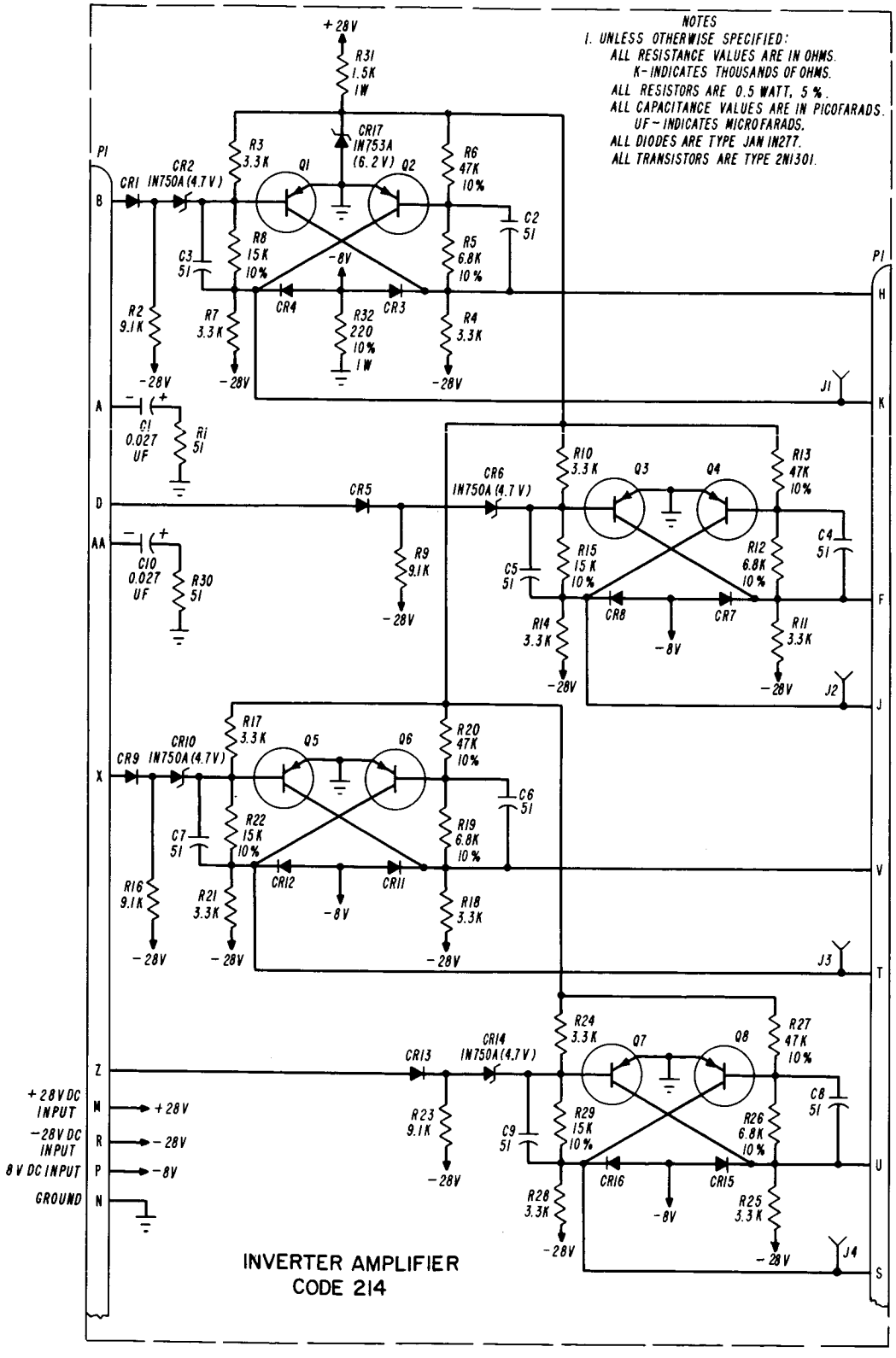
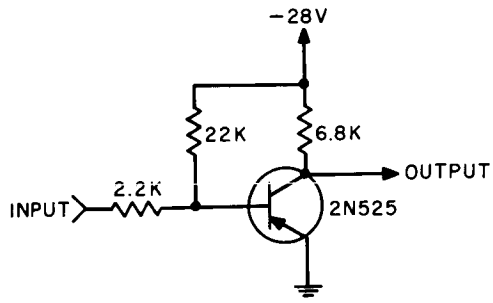
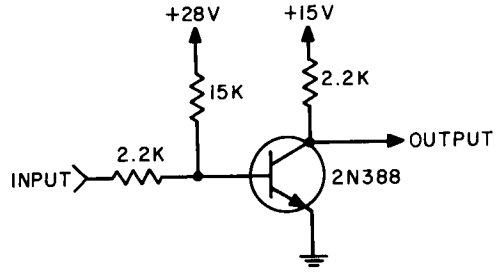


Figure 3-13. Inverter Amplifier



	IN	OUT
"1"	+6.5V ± 9.5V	-28V
"0"	0V TO +0.6V	0V

Figure 3-14. Computer-to-Logic Converter



	IN	OUT
"1"	-8V	+15V
"0"	0V	0V

Figure 3-15. Logic-to-Computer Converter

3-4 RESULTS

A. OPERATION OF THE MODEL

The breadboard model of the adaptive voter, which has been constructed for this project, has been subjected to sufficient testing to verify the operation of the device under a variety of input error conditions. Because of the nature of the particular adaption scheme which was chosen and because the integrate period in each operating cycle of the model was sufficiently long, the vote weight of any input could be adjusted to the desired level without creating the problem of convergence time.

In testing the voter, several weaknesses were noted. The primary difficulty arose because of the differences in the characteristic curves of photoconductor resistance versus time between replicas of the mercury cell integrators. In the particular adaption scheme used here, the existing theoretical value of the individual vote weights were stored in the computer memory. As new theoretical weights were computed, any changes to be made in the vote weights were determined by comparing the stored value of an input with the newly computed value. The computer's version of both the stored vote weights and the newly computed weights are related to the photoconductor resistance in the mercury cell integrators through a single ideal characteristic curve equation. As a result of this approximation, changes in resistance of photoconductors called for in response to computed changes in weights are continuously subject to error. Although the error made in any single adapt cycle may be small, the large number of adapt cycles which would be encountered during the life of the system tend to make the actual vote weights diverge from their intended value. This divergence has the combined effect of creating wrong vote weight ratios between inputs of different reliability, and it changes the ratio of the input channel series impedance to the fixed parallel impedance at the threshold detector input. Since the latter ratio determines the threshold level of the detector, this change can be extremely detrimental.

B. EVALUATION OF THE MERCURY CELL INTEGRATORS

The unique set of characteristics associated with the memory cell integrators definitely provides a feasible means for implementing adaptive voters. Several of the characteristics of these devices have been observed which make the integrators undesirable as circuit components from either the user or the circuit designer's point of view. The remainder of this portion of the report reviews the advantages to be obtained from the use of these elements and describes some of the disadvantages also inherent in their use.

1. Desirable Characteristics

a. It has been found that the output resistance of these devices is very stable in time and is not sensitive to loss and restoration of power. This combined with the fact that they required no power to operate except during the integrate operation periods makes them particularly desirable for use in applications where low power levels are particularly desirable or power may be temporarily lost.

b. The output resistance of the device is electrically adjustable across a continuous range of values. This permits the adaption increments to be as finely grained as desired.

c. The average range of resistance values obtainable from the Curtiss model 251 integrators lies between approximately a megohm and a few ohms. This large range of values facilitates the construction of adaptive voters having the wide range of input weights required for reliable operation of the threshold output circuit.

2. Undesirable Characteristics

a. As figure 3-7 illustrates, the resistance versus time curves of the model 251 integrators have a symmetrical or "two-side" curve about some low resistance region. In normal operation, the resistance of a particular integrator would be varied up and down one side of this curve between the high and low resistance regions. It is obvious, however, from an examination of these curves that an error in the integration timing at some point near the low resistance region could transfer the operation region of an integrator to the opposite side of the curve. This action would then tend to reverse the desired adaption procedure in that a later signal which was intended to lower a vote weight might, in fact, increase the weight. The results are obviously undesirable.

b. Because the output of this model of the mercury cell integrator is reflected only by a change in resistance between the output terminals, the existing value of the output must be determined by one of the following procedures. Either the equipment which determined the desired value of the weight must store that value, or special circuitry must be provided to physically interrogate the integrator. The first option was chosen for this model because the SDS-910 could handle the function easily. This method would be at least as cumbersome as the second method in the type of voter models which would see use in large redundant systems. Both of these methods would require enough circuitry to implement that the net advantage of using a voter of this complexity would be doubtful. This does not necessarily mean that mercury cell implementations of adaptive voters are impractical per se, but it does mean that the adaption may have to be done on an incremental basis not requiring knowledge of the previous value of the individual input weights.

c. The difference between the resistence versus time curves of different nominally identical mercury cell integrators has already been pointed out in section 3-3-c. Although this characteristic obviously reflects a problem in quality control of the manufacturing process, it definitely causes design problems at the present state-of-the-art in the use of these components.

d. Although no controlled shock and vibration tests were conducted, the operation of the devices seems to be relatively sensitive to these stresses. In one instance the investigator intentionally struck one of the integrators with a mild blow of an ordinary wooden pencil and the photo-cell failed into the open circuit mode. As this indicates the sensitivity to shock is apparently very high; therefore, the reliability of the devices in any type of mobile equipment would probably be quite low.

3-5. CONCLUSIONS AND RECOMMENDATIONS

The results of this project have shown that adaptive voters can be constructed using mercury cell integrators as the variable input weighting devices. The results have also shown that the implementation of such voters using voting schemes which require computation of optimal weight values require relatively complex feedback adaption control loops. This, combined with the problems involved in using the presently available models of the mercury cell integrators, leads to the conclusion that adaptive voters of this type are presently an impractical means of improving the reliability of redundant systems.

Despite this rather negative conclusion, the potential improvement in system reliability offered by the use of adaptive voters cannot be ignored. In order to advance the art toward the realization of practical adaptive voting techniques, the following recommendations for future work in this area are made.

1. The search for suitable weighting devices should be actively continued. Certainly, physical changes in the mercury cell integrators which overcome some of the present problems should be considered.

2. A broad range of adaption schemes including those proposed by Pierce should be examined on a comparitive basis. The objective in performing this comparison would be to determine the cost, in terms of lost reliability, of using simple, easy to implement schemes rather than the more sophisticated "optimal" adaption schemes. Although many adaption schemes are not amenable to mathematical analysis, the comparison would be relatively easy to perform through the use of a computer simulation program similar to the one used in the present project. In this case, however, the entire voter would be simulated rather than just the feedback adaption control loop.

3. Using the results of the comparison study recommended above and if suitable weighting devices are available which have the characteristics required by the particular adaption schemes under consideration, complete breadboard models of the more promising schemes should be constructed and tested.

BIBLIOGRAPHY

1. Pierce, W. H , "Adaptive Vote-Takers Improve the use of Redundancy," Westinghouse Scientific Paper 316-K000-P2, Pittsburgh, Pa., 1962. (Also published as "Improvement of the use of Redundancy," Redundancy Techniques for Computing Systems, Spartan Books, Washington, D. C., 1962.

2. "Final Report Research on Failure Free Systems", NASA CR 56686 (X64-14856) (U.S. Gov't. Agencies and Contractors only).

3. Von Neumann, J., "Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components," Automata Studies, Ed. C. E. Shannon and J. McCarthy, Princeton University Press, Princeton, N. J., 1956.

4. Widow, B. and Hoff. M., "Adaptive Switching Circuits", 1960 IRE Wescon Convention Record.

5. Widrow, B., "An Adaptive 'Adaline' Neuron Using Chemical Memistors", Technical Report 1553-2, Stanford Electronics Laboratories, 1960.

6. Widrow, B., "Adaptive Sampled-Data Systems - a Statistical Theory of Adaption", 1959 Wescon Record, Part 4.

SECTION 4

AN IMPLEMENTATION OF A FAILURE RESPONSIVE SYSTEM

4-1 INTRODUCTION

One of the main goals of the current phase of the contract is the development of effective techniques for implementing failure responsive systems. As described in an earlier report[1] failure responsive systems are redundant digital systems which have the capability to partially reorganize themselves in response to the occurrence of detrimental internal failure patterns. In failure responsive systems, the failure of a stage to meet the system's operational criteria will initiate an action to switch a subsystem from another stage to restore the vulnerable stage to operation.

One method of performing this partial system reorganization would be to use a central controller to detect errors or failures, sense the need for reorganization, and perform the switching of the spare subsystems from a "healthy" stage. It should be noted, however, that a failure within such a vital central controller would result in the complete loss of reorganizational capability for the system. Having foreseen this difficulty, the effort on this task has been concentrated exclusively on systems with distributed error detection and switching capability.

The implementation of this distributed function philosophy requires special circuitry to perform both error detection and switching at each subsystem location. The first goal of this part of the study has been the development of techniques for implementing this circuitry in a manner such that failure responsive systems are compatible with modern semiconductor circuits. The second goal has been to design a specific study vehicle which can be used to demonstrate the feasibility of such systems.

4-2 GENERAL CONSIDERATIONS

A. Comments on Switching Strategies

The first problem to be solved in optimizing the design of a failure responsive system is the determination of a strategy for calling to the aid of vulnerable or failed stages the replacement subsystem which can be most easily sacrificed by the remainder of the system.

---

[1] Special Technical Report No. 5, "Analysis and Development of Failure Responsive System Organizations," Appendix.

The objective of the computer simulation test program, reported in the Appendix, was to determine the characteristics of an optimum replacement selection pattern. It was concluded from this survey that all subsystems should have approximately equal relocation potential and that three to five "spares", replacement subsystems, per stage was sufficient for most applications. From a practical standpoint, it would seem that the optimal number of spares per stage seems to be equal to the order of redundancy of the system for non-fractional orders of redundancy*. The choice of this number of spares results in every subsystem having the capability to move to only one location other than its original. This condition greatly simplifies the peripheral circuitry required to electronically position a subsystem in a particular functional location. In addition, this number of spares per stage precisely satisfies the desired characteristic of an equal number of potential locations per spare.

B. Input Control of Subsystem Memory

The basic premise of the failure responsive systems which have been considered in this study is that a properly operating subsystem can perform the function for which it was intended if it is supplied with a set of correct input signals. This premise is not necessarily true if the subsystems contain active memory elements. Even if all circuits are working and the inputs are correct, a correct output cannot be guaranteed unless the memory is first set to the proper initial state. If the failure which has generated the need for repair at a certain functional location also causes ambiguity to exist between the memories of the subsystems originally assigned to that location, a particularly troublesome problem will emerge. Before any subsystems of this type can be handled by failure responsive system techniques, a methods must be found for determining the correct memory state associated with the vulnerable stage and for setting the memory elements of the subsystems effecting the repair to the correct state.

The failure responsive techniques developed in this study provide a solution to at least part of this problem. If the state of the memory elements of the subsystems being considered are controlled entirely by the input signals, the implementation techniques can allow the state of the memory to be set by the input signals before effecting a vote. This technique assures that the failed subsystem will be determined and eliminated from the system.

---

* Fractional ordered systems are those not having the same number of subsystem replicas at each stage.

In addition, in systems containing subsystems whose memory elements are periodically reset to a given state, the subsystem switching and error elimination can be properly timed in relation to this cycle. The error elimination function is simply delayed until the memories of all of the subsystems have been reset.

C. Multiple Inputs and Outputs

In order to switch a subsystem containing only one input and one output, a certain minimum amount of switching circuitry is required. This minimum amount is a function of the total number of locations to which the subsystem may be assigned. At least three logic gates at the subsystem input and three at the output are required for a subsystem containing a single input and a single output, in order to enable the subsystem to perform only one move. If the number of inputs is doubled, the number of gates is likewise doubled. This factor limits the number of inputs and outputs which can be handled by any reorganizational scheme, but this limit is not severely restrictive. The implementation technique demonstrated by the study vehicle design provides a means for simultaneously switching multiple inputs and outputs, without seriously increasing the complexity of the switching circuitry.

D. Vital Elements in the Switching Circuitry

In any failure responsive system there will appear some sections of peripheral switching circuitry which are vital to the operation of the system. The majority of the peripheral circuitry will affect the repair capability of only one stage, or one subsystem; however, there are cases in which a failure of a vital element in the switching circuitry or in the stage output circuitry will cause immediate system failure.

There are several ways to protect the system against such catastrophic failures. The only apparent solution to this problem is to introduce fixed redundancy by making this vital circuitry either functional redundant or component redundant.

In the design of the study vehicle, the objective has been to prove the feasibility of failure responsive system designs, and no effort has been made to protect the system against all of the failure modes of every circuit function. In some cases, however, the more critical switching functions have been protected against the effects of their most serious circuitry failure modes. In any final design of a practical failure responsive system, protection of vital system functions would have to be provided.

E. Subsystem Characteristics

In converting a non-redundant digital system to a fixed redundant configuration, almost no restrictions are placed on the functional requirements of the subsystems into which the system is divided. For example, the actual functional operation of subsystems supplying a set of nominally identical binary signals to a set of majority voters has absolutely no effect on the voter's capability to resolve errors.

In failure responsive system, a similar situation exists during the early life operation before multiple failures have occurred in any stage. During this early life period, the systems operate like multiple-line systems, and the functional requirements could vary from stage to stage without causing any difficulty. The problem arises when "repairs" are attempted. In order for a working subsystem to effectively replace one which has failed, the replacement must be capable of duplicating the function of the failed unit. This requirement means that a subsystem which has the potential capability to operate in more than one location must have the capacity to perform, not only its original function, but also the function required at every location to which it may be moved. If in fulfilling this requirement, a subsystem must perform even two significantly dissimilar functions, the increase in subsystem complexity of the circuitry required to implement both functions and to permit switching between the functions could easily offset the improvement in system reliability sought through the use of failure responsive systems. Unless a simple means can be found for changing subsystem functions, this offsetting effect seems to limit the applicability of failure responsive techniques to systems (or portions of system) which can be divided into functionally similar subsystems.

Even before one attempts to design a realizable failure responsive system, a second practical restraint on the characteristics of individual subsystems can be observed. In order to electronically switch a single input, single output device between two locations, the minimal requirements of one gate per location for both the input and the output, and a memory device to store the desired location must be met. When the circuitry required to generate signals for calling spares and to perform error detection and correction is added to this minimal control circuitry, it becomes apparent that the complexity of an individual subsystem must be fairly large to be even equal to the peripheral circuitry required per subsystem. Neither the complexity nor the homogeneity requirement alone is particularly severe, but the combination of the two requirements restricts the applicability of failure responsive techniques to a relatively small class of systems.

4-3 THE STUDY VEHICLE

A. Desirable Characteristics

A study vehicle design is required in order to verify the practicability of both the organizational strategy theories and the implementation techniques which have been developed. The vehicle must be chosen to tax design capability to the limit of the art, and, more importantly, to demonstrate that the set of failure responsive techniques developed can provide a powerful tool for increasing the useful lifetime of spaceborne digital systems.

Specifically, a study vehicle is needed which contains some type of memory capability, such as flip-flops or shift registers. This is necessary to demonstrate the capacity of the implementation techniques to handle the complications associated with the switching of memory elements. The study vehicle should also demonstrate that the techniques can successfully handle the switching of multiple inputs and outputs. The specific vehicle should contain several identical or at least very similar subsystems which are required to perform in different locations in the system. The circuitry associated with the subsystem should be at least as complex as the switching circuitry needed to implement the failure response capability. It is also desirable that the vehicle operate in some definite cycle, rather than perform only one operation continuously.

After an extensive investigation, it was decided that the type of system which would best encompass all of these characteristics would be a special purpose arithmetic unit. The specific study vehicle selected is the arithmetic section of a beam steering computer from a phased array radar system. This unit receives data, performs a number of arithmetic operations on the data, temporarily stores intermediate results, and then reads out the final results of the computations and is reset. The unit includes all of the above characteristics of a desirable study vehicle.

B. Description of a Non-Redundant Beam Steering Computer

A beam steering computer is required in a phased array radar system, to compute the settings of the phase shifters associated with the phased arrays. The specific system being considered here is so organized that settings are computed for groups of phase shifters, each group being arranged in a separate row or column of the array. This allows only one setting to be computed for each row or column of phase shifters, rather than the separate calculation for each individual phase shifter which might otherwise be required.

The computations for deriving the phase settings of the rows or columns can be reduced to the following form:

$$(1)\ \phi_n = \phi_o + nK + n^2C + S_i$$

where

ϕ_o = the phase setting of the reference row or column. (This will probably be selected in the center of the array and be fixed to a value of zero.)

n = An index which may assume either positive or negative values and which indicates the position of a row or column relative to the reference row or column (i.e., for the row immediately above the reference row n would equal +1 and for that immediately below the reference row n would equal -1; the next rows above and below would have n = +2 and n equal -2 respectively.)

K = A factor which is a function of the fixed array geometry, the transmission wave length, λ, and the beam angle, θ. (θ is measured horizontally for columns and vertically for rows.)

C = A set of factors which is a function of the fixed array geometry and the beam spoilage, R.

S_i = A set of compensation factors for the non-planar wave front which is directed to the array. This factor is distinct for each separate row or column with the exception that there is symmetry about the center of the array; i.e., i equals the absolute value of n.

ϕ_n = The phase setting of the n^{th} row or column.

An efficient method of implementing this computation may be seen by expanding equation (1), using the following relationships:

$$n^2 = (n-1)^2 + 2n-1$$

$$n = n-1 + 1$$

substituting in equation (1):

$$(2)\ \phi_n = \phi_o + (n-1)K + (n-1)^2\ C + K + (2n-1)C + S_i$$

By inspection, we see that the first three terms are equal to ϕ_{n-1}* less the compensation factor. Therefore,

$$(3)\ \phi_n = \phi_{n-1}{}^* - S_{i-1} + K + (2n-1)\ C + S_i$$

*Equal to ϕ_{n+1} for values of negative n.

Thus it will be possible to compute the phase settings by a process involving only simple additions or subtractions relative to the previous phase setting if the settings are sequentially determined in the order of increasing distance from the reference row or column.

A block diagram of the beam steering arithmetic unit used to calculate $\emptyset_n$ is shown in figure 4-1. The circuitry consists of four shift registers and six one-bit serial adders or subtractors. In addition, storage for the compensation factors, S_i, is provided. These factors are fixed by the geometry of the array and are implemented by combinational logic used in conjunction with a shift register to provide a serial output. The necessary timing signals are provided by two counters with associated decoding gates.

In the configuration shown, the three adders in figure 4-1(a) are used to calculate the phase settings of rows or columns offset in a positive direction from the reference row or column, while the adder and two subtractors in figure 4-1(b) are used for those offset in a negative direction. Thus, two phase settings will be computed simultaneously (one above and one below the reference column). Table 4-1 shows the initial contents of the registers and the contents at the beginning of the computation of $\emptyset_n$. The outputs are directed to shift registers which store computed results, and a parallel output from these registers is used to drive the phase shifters.

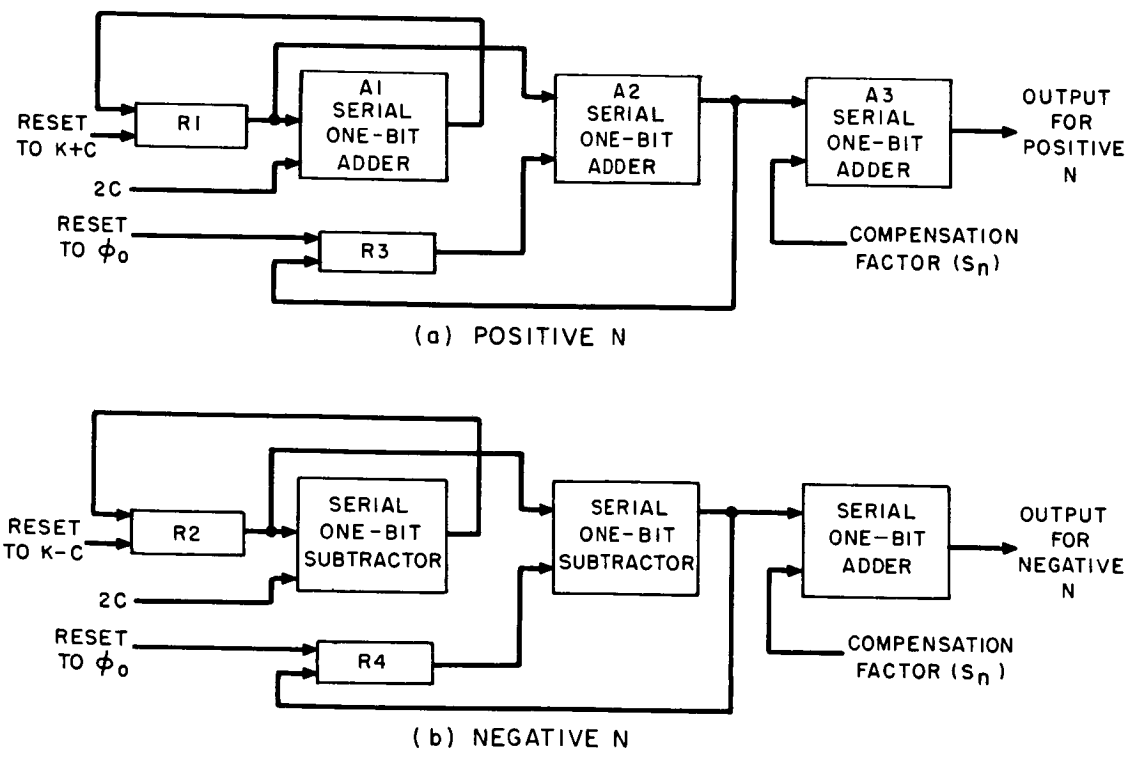


Figure 4-1. Beam Steering Arithmetic Unit

TABLE 1. CONTENTS OF ARITHMETIC UNIT REGISTERS

Register	Contents	
	Reset	Prior to the Computation (assuming positive values of n)
R1	K + C	K + (2n-1)C
R2	K - C	K + (-2n+1)C
R3	ϕ_o	$\phi_{n-1} - S_{i-1}$
R4	ϕ_o	$\phi_{-n+1} - S_{i-1}$

C. Description of the Failure Responsive Beam Steering Computer

The non-redundant beam steering arithmetic unit operates on a three-phase cycle: (1) data input, (2) computation and storage, and (3) results read-out and reset. The failure responsive version of the computer has been designed so that the error detection and correction is properly timed in relation to this cycle. All errors are detected during the computation phase, and error correction is accomplished through system reorganization during the results read-out phase, when the unit is waiting for new data.

One stage of the failure responsive beam steering computer is shown in figure 4-2. The arithmetic unit for positive n, in figure 4-1(a), has been chosen as the subsystem of the failure responsive version of the computer. This subsystem will also perform the computations for negative n if the two adders, A_1 and A_2, are changed to subtractors. This can be done by inverting the input which is to be subtracted, and changing the end-of-computation RESET to a SET. As can be seen in figure 4-2, these two changes are handled by the gating circuitry on the subsystem's adder inputs.

In order to provide simultaneous signals for both the rows and columns of the phased array, displaced in both the positive and negative directions from the reference rows and columns, four arithmetic units are required in the non-redundant computer. Two units perform the computations for positive n, and the remaining two perform those for negative n. Since there are four non-redundant units, the failure-responsive design comprises four stages, with each stage triplicated, making a total of twelve subsystems. A block diagram of the system arrangement is shown in figure 4-3.

1. The Switching Strategy Employed

During the portion of the failure responsive systems study described in the Appendix, a comprehensive set of design rules was established to aid the designer of such systems. Much of the emphasis was placed on the investigation of subsystem switching strategies, with the result that many strategies were found which provide a significant

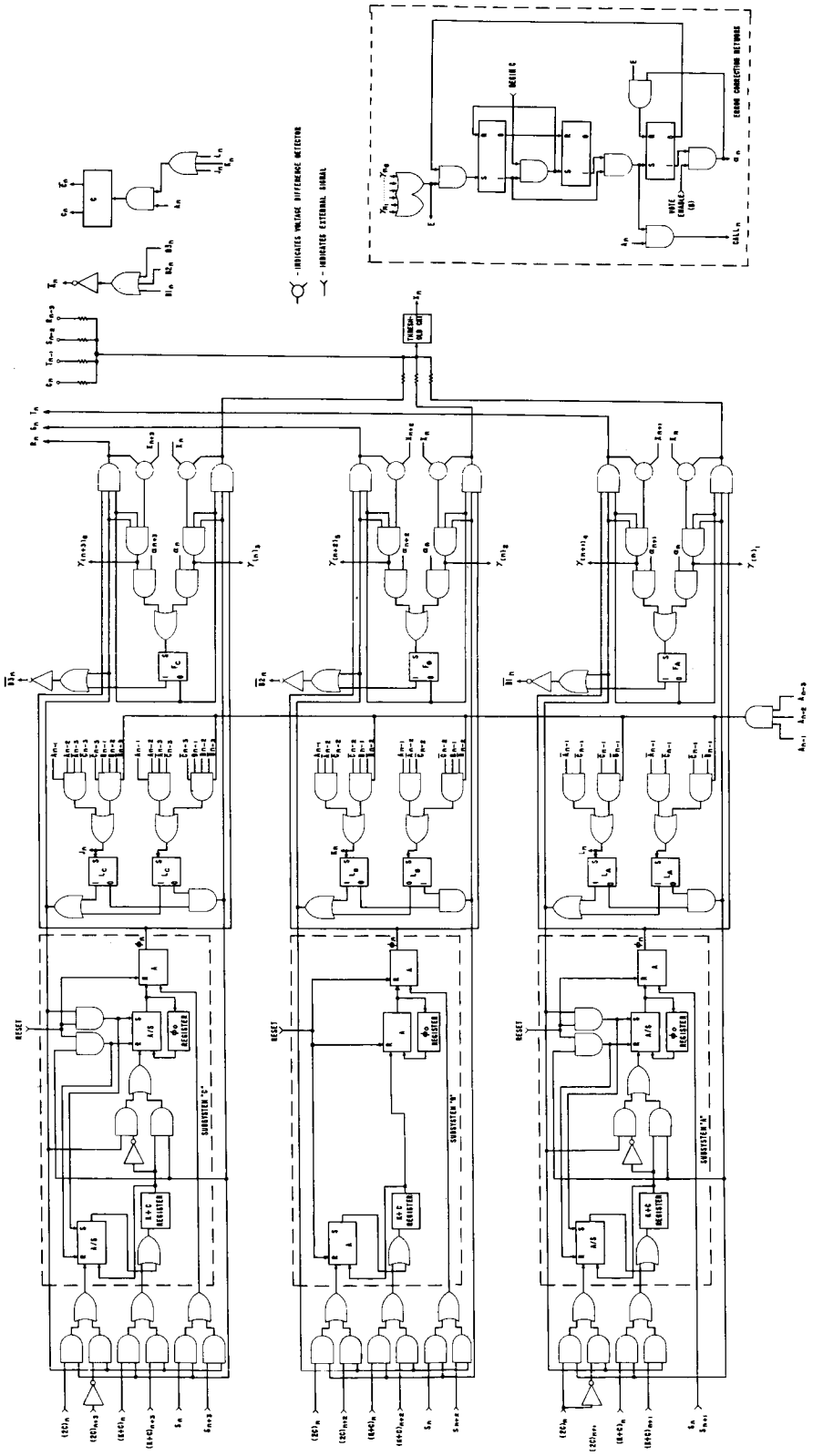


Figure 4-2. Failure Responsive Beam Steering Computer Stage N

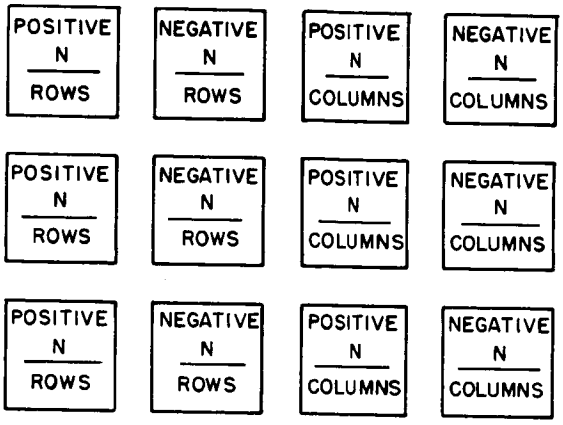


Figure 4-3. Block Diagram of Failure Responsive Arrangement

improvement over the reliability of multiple-line redundant systems. The most effective strategy was found to be the progressively distributed step list pattern, represented diagrammatically in figure 4-4. In the figure, each block represents a subsystem replica and each column of blocks represents a stage of the system. The spare list for subsystem X consists of the numbered subsystem replicas. Subsystem 1 would be the first "spare" to go to the aid of X's stage, subsystem 2 would be the second, and so on. Because of the distributed characteristic of this strategy, it provides a significant reliability improvement only for systems containing a large number of stages, such as the twenty-stage system used in the evaluation. For a relatively small system the distributed step list is not feasible, and the step list strategy shown in figure 4-5, provides the best reliability improvement. This strategy was chosen for the design of the four-stage beam steering computer. The design provides for three spares for each of the four stages, one spare being selected from each of the other three stages. The manner of detecting errors in the output of a stage, physically locating and switching the proper spares, and eliminating errors is discussed in the following sections.

2. The Classes of Circuits

The study vehicle design is composed of three main classes of circuits: (a) input-output channel selection circuitry, (b) error detection and correction circuitry, and (c) location control circuitry.

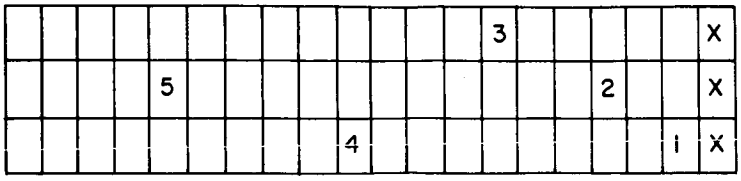


Figure 4-4. Progressively Distributed Step List Pattern

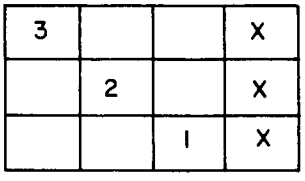


Figure 4-5. Step List Pattern

a. Input-Output Channel Selection

The input selection circuitry consists of the input gating at the left of the subsystems in figure 4-2. The inputs 2C, K+C, and S designate the inputs referred to in the description of the non-redundant beam steering computer. The subscripts indicate the stage of the system to which the inputs belong. The stage shown in the figure is stage n, the one to the right of this stage is stage n+1, the next stage in line is stage n+2, etc.

The inputs which actually energize each subsystem are determined by the location control circuitry. This circuitry energizes one of the two horizontal lines above and below the corresponding subsystem and simultaneously deenergizes the other line. Referring to subsystem A in the figure, there are two stages in which this subsystem could be operating, stage n, or stage n+1. If redundant flip-flops L_A are reset, the subsystem will operate in stage n, and the "n" inputs will be fed into the subsystem. If the flip-flops are set, the n+1 inputs will be fed into the subsystem.

The output selection circuits operate in exactly the same manner. The same two location control lines that energize the proper inputs, also energize the corresponding outputs. The location control lines control the operation of the output selection gates, at the far right extremes of the lines in the figure. The output of the lower of the

output selection gates for subsystem A channels signals to the threshold voter circuit which produces the output X_n, of stage n. In the same manner, the upper output selection gate for subsystem A produces output T_n, which channels information to the stage n+1 threshold voter circuit.

The location control circuitry determines which part of the input-output selection circuitry is operating and as a result, in which stages the subsystems are effectively operating. There must also be provision for determining which of the subsystems are operating properly and which ones should be eliminated from the system. This function is performed by the error detection and correction circuitry.

b. Error Detection and Correction

The error detection and correction circuitry consist of the circuitry in the lower right of figure 4-2, plus the "failure" flip-flops F_A, F_B, and F_C with the associated gating circuits which feed them. Error detection and correction, as mentioned earlier, must be properly timed in relation to the three-phase cycle of the phased array radar system. It also must allow for random noise which can occur in the input signals. To account for both of these factors, error detection is performed only during the computation phase, designated as period B. But one error alone will not enable the spare call function. There must also be a second error in the same stage during the second B period. Consider the section of the error detection circuitry in the lower right of figure 4-2 when all flip-flops are reset. An error signal from one of stage n's subsystems, signal γ_{n_i}, will set the first flip-flop, FFI, during period B. At the beginning of the readout and reset phase (period C) an external signal will set FF2. If the error signal γ, still remains at the next B period, FF3 will be set, the failed subsystem will be eliminated by α_n, and a spare subsystem will be switched into stage n by the location control circuitry.

The ERROR signal, γ, is generated by the gating circuitry which controls flip-flop F_A, F_B, or F_C. Consider the circuitry associated with F_A. If the output of the output-channel-selection gate (subsystem output) does not agree with the stage output, x_n, the difference detector produces a positive error signal, γ_{n_1}. When the error elimination circuit energizes α_n, flip-flop F_A is set, eliminating subsystem A from the system. As can be seen in the figure, this error detection circuit is reproduced for every location of every subsystem, providing distributed error detection for the system.

When a subsystem failure occurs, as described above, a "call" signal is generated and directed to the location control circuitry. The location control circuitry determines which of the "spare" subsystems can be moved to the weak stage.

c. Location Control

When the first failure occurs in a stage, the failed subsystem is eliminated by majority vote, since there are two operating subsystems remaining in the stage. When the second failure occurs, a spare must be called from another stage in order to perform the majority vote. The order in which spares are examined, to determine their availability to be switched, is specified by the step list pattern in figure 4-5. Each spare is examined in turn to see: (1) if it is in its original stage, and (2) if there are three operating subsystems in the stage. If either of these is not true, the next spare in turn is examined. All three spares are examined, if necessary. If none is available to be switched under the above two criteria, the location control circuitry has the capability to "rescan" the spare list with less stringent criteria. A spare will be switched if it is operating properly in its original stage and there is one other operating subsystem in the stage. It is better to leave the spare's stage with only one operating subsystem in order to resolve the ambiguity in the first stage, than to leave the first stage output indeterminate, thereby causing system failure.

During the first scan of the spare list, there are three possible circumstances which would prevent a particular spare from being switched:

A. The previous loss of a subsystem to another stage or a previous failure in the stage has occurred
B. The spare itself has already moved or failed
C. The two other subsystems in the spare's stage have previously moved or failed.

Assume that stage n requires a spare. The first spare to be examined is subsystem A in stage n-1. This spare will be switched if conditions A and C above are false. This will be designated as $\text{Move}_1 = \bar{A}_{n-1}\ \bar{C}_{n-1}$. The second spare, in stage n-2, will move only if conditions A and C are false, and condition A is true for the first spare. Therefore $\text{Move}_2 = A_{n-1}\ \bar{A}_{n-2}\ \bar{C}_{n-2}$. In the same manner, for the third spare, $\text{Move}_3 = A_{n-1}\ A_{n-2}\ \bar{A}_{n-3}\ \bar{C}_{n-3}$.

If no spares are available, the first spare on the list will be examined again. The conditions for a move now are:

$$\text{Move}_1 = A_{n-1}\ A_{n-2}\ A_{n-3}\ \bar{B}_{n-1}\ \bar{C}_{n-1}.$$

The A's refer to the fact that the first scan produced no available spares. B_{n-1} indicates that the spare itself is operating, and C_{n-1} indicates that there are two operating subsystems in stage n-1. In a similar manner, the conditions for the second and third spares moving are:

$$\text{Move}_2 = A_{n-1}\ A_{n-2}\ A_{n-3}\ B_{n-1}\ \bar{B}_{n-2}\ \bar{C}_{n-2}$$

$$\text{Move}_3 = A_{n-1}\ A_{n-2}\ A_{n-3}\ B_{n-1}\ B_{n-2}\ \bar{B}_{n-3}\ \bar{C}_{n-3}$$

If we combine the last three conditions with the conditions for a move during the initial scan, we obtain the total conditions for move for all three spares:

$$\text{Move}_1 = \overline{A}_{n-1}\ \overline{C}_{n-1} + A_{n-1}\ A_{n-2}\ A_{n-3}\ \overline{B}_{n-1}\ \overline{C}_{n-1} \quad (1)$$

$$\text{Move}_2 = A_{n-1}\ \overline{A}_{n-2}\ \overline{C}_{n-2} + A_{n-1}\ A_{n-2}\ A_{n-3}\ B_{n-1}\ \overline{B}_{n-2}\ \overline{C}_{n-2} \quad (2)$$

$$\text{Move}_3 = A_{n-1}\ A_{n-2}\ \overline{A}_{n-3}\ \overline{C}_{n-3} + A_{n-1}\ A_{n-2}\ A_{n-3}\ B_{n-1}\ B_{n-2}\ \overline{B}_{n-3}\ \overline{C}_{n-3} \quad (3)$$

The scan and rescan of the "spare list" described above would actually result in a relatively slow reorganizational capability for the system. The actual location control circuitry, therefore, does not control switching in this manner. The three "move enable" equations (1), (2), and (3) above are implemented with the combination logic associated with location flip-flops L_A, L_B, and L_C in figure 4-2. The output of this logic would be "and" ed with the appropriate "call" signal. The entire scan and rescan is thus accomplished in a single bit-time. The subscripts on the inputs to the location control gates are referenced to the stage which calls the associated subsystem as a possible spare. Functions A_n, B_n, etc., are produced by the peripheral gating circuitry shown in the figure. The subscripts on these functions, however, refer to the stage pictured, n, and would need to be re-numbered to correspond to the inputs of the location control circuitry.

The threshold circuit is essentially a two-out-of-n vote circuit. As the figure shows, one of the inputs is C_n, the signal referred to above which indicates that there is only one remaining subsystem in the system, the second having been switched to another stage. This signal puts an added "ONE" on the voter input, allowing the one remaining subsystem to control the vote, since there are no remaining spares to be called. By allowing the remaining subsystem to control the vote in this manner, system life is further extended.

The location control flip-flops and combinational logic, having a vital system function, are duplicated to increase system reliability. The error elimination network and the output circuitry would also be replicated in the final design of operating systems.

c. The System Reliability

Using values of failure rates proportional to subsystem size and complexity, the system was simulated on the Univac 1107 computer. The computer simulation program used was the same program used for the establishment of design rules during the first phase of the failure responsive systems study. The system proved to have a useful life* more than twice that of a multiple-line redundant version of the same phased array radar system.

*Useful life is defined as the time at which the system becomes 90% reliable.

The phased array radar subsystem is composed of 31 flip-flops and 33 logic gates. The switching circuitry required to implement the failure responsive capability is made up of 10 flip-flops and 93 gates. For the purposes of simulation, two different pessimistic estimates of failure rates were made for the switching circuitry associated with each stage. These estimates were then equally divided between the three subsystems of the stage. This equal division has a very optimistic effect on system reliability estimates. Based on the failure rates of presently available integrated circuits, approximate failure rates were also assigned to the subsystems. The simulations included a comparison of the system with a multiple line version of the same system. In addition to the simulation of the four-stage design, eight stage versions of the same systems were simulated for comparison. The results of these simulations using the two switching circuit failure rate estimates are summarized in table 4-2. In the second case the failure rate of the switching circuitry was assumed to be 50% more than in the first case.

Additional simulations of the failure responsive study vehicle were performed with the error elimination network and voters assumed to be separate subsystems with no repair capability. Failure rates proportional to system complexity were again assigned . A failure in any part of the separate circuitry was assumed to cause system failure. Even with this obviously very pessimistic assumption, the failure responsive design proved to have a useful life comparable to that of the multiple-line system, which was assumed to have perfect majority voting circuits.

TABLE 4-2. COMPARISON OF SYSTEM RELIABILITY

System	Subsystem Failure Rates (Failures/Hr.)	Useful Life (Hours)	
		Four Stages	Eight Stages
NON-REDUNDANT	$.46 \times 10^{-6}$	57,300	28,650
MULTIPLE-LINE	$.46 \times 10^{-6}$	217,610	154,460
FAILURE RESPONSIVE	$.67 \times 10^{-6}$	564,640	377,470
FAILURE RESPONSIVE	1.00×10^{-6}	378,310	252,910

4-4 CONCLUSIONS

The design of a practicable system having failure responsive capability has been accomplished. This design has shown that such systems can be implemented using standard combinational logic circuits to form the various error detection, error correction and "repair" switching functions which are required.

Although the amount of peripheral circuitry required to implement the functions mentioned above does not seem excessive, it may be concluded that the subsystems must be at least as complex as those described for the beam-steering computer considered here. The successful design of this particular study vehicle demonstrates the applicability of failure responsive system techniques to systems containing input controlled memory. In addition, the design has shown that subsystems with multiple inputs can be handled with the reorganizational capability of these systems.

The present design of the study-vehicle contains no specific provisions to protect the system against all failure modes of the peripheral circuitry. In many cases, failures in this circuitry will be treated as subsystem failures. The natural extention of this work would be to continue the design effort to provide protection against all peripheral circuit failure modes and to extend the computer simulation program to permit final design evaluation.

SECTION 5

MEDIUM COMMUNICATION FOR MODULE REORGANIZATION

5-1. INTRODUCTION AND PROBLEM STATEMENT

This study has been one part of the larger program whose objective is to consider new techniques which are expected to increase the reliability of vital electronics. Most of the effort expended in the large program has been oriented toward developing techniques for more effectively employing redundant equipment to extend overall system life. As one result of this effort, it has been found that systems which have the capability to partially reorganize the connection pattern linking their individual subsystems tend to have significantly longer useful life spans than systems with a fixed subsystem configuration. The reorganization capability allows these systems to avoid the use of failed subsystems and to maintain a uniform distribution of the operating redundant subsystems. The two inherent primary difficulties with systems having this capability are (1) the need for relatively complex interconnection switching circuitry and (2) the need for highly homogeneous subsystems.

The system organization which is described in the remainder of this report was originally conceived as a means for implementing the "failure responsive systems" described above. The primary purpose in developing this new organization was to reduce the complexity of the interconnection circuitry. As will be shown, this initial investigation has resulted in the formulation of a system organization which accomplishes the original purpose and provides an extremely flexible technique for reorganizing systems into redundant and non-redundant structures as the applicational requirements vary. The system organization which is proposed has the added advantage that graceful degradation of the system functions are almost inherently achievable.

As the organizational concept was proposed, the system function would be performed by a group of subsystems communicating through a common medium. The channeling of the information would be accomplished either by tagging the data with some time or frequency code, by providing an instruction program within each subsystem or by providing a central controller which would be associated directly with the medium. The objectives of the study reported here has been to consider the feasibility of constructing systems organized in the above manner and to formulate the basic design configuration of a system of this type.

5-2. SYSTEM CHARACTERISTICS

A. MEDIUMS

1. Channel Economy

Any system in which every subsystem has the capability to communicate with every other system is often referred to as a "strongly connected system". Systems may be strongly connected through a system of individual signal channels such as wires or through a common medium such as a gas, a liquid or a block of solid material. Subsystems communicating over individual signal channels may require as many as N^2 undirectional channels or at least $N^2/_2$ bidirectional channels with the associated channel selection circuitry available at each subsystem. If, however, a medium is used as a central node through which all data passes, the characteristics of a strongly connected system are retained, but the number of channels is reduced to 2N unidirectional channels or N bidirectional channels with all or most of the channel selection circuitry confined to the medium.

A typical although rather mundane example of the economy of using a central medium to contain the channel selection circuitry is the telephone system. In this case the switchboard fills the role of a medium which performs all the channel selection functions for the system. This example also illustrates that a large system might profitably be broken into segments organized around individual media each of which communicates with the other media. Although the question is academic at this point, the question of whether the individual media should communicate directly or through a "higher rank" media is one which must be solved before the very large, complex computing systems could be implemented.

2. Media with Memory

The use of a medium as a common signal channel does not necessarily imply that the medium must transmit the signals instantaneously without intentionally introduced delay. By considering the central memory of a digital computer as another type of medium, it is immediately obvious that the information storage properties of an electronic memory may be highly advantageous in many media. In a system employing multiple redundant replicas of each subsystem to achieve higher system reliability, the use of a subsystem interconnection medium with storage properties immediately offers the following desirable possibilities.

a. Individual subsystems may operate asynchronously.

b. If voting speed exceeds computational speed, voting may be done by relatively few voters in common pool voting on the copies of data stored in the medium.

B. BASIC SYSTEM OPERATION

The interconnection of subsystems through a storage medium does not necessarily restrict the choices of memories to any smaller class than would normally be found in a modern digital computer. Because the mechanical scan characteristics of a drum memory seem to be particularly useful in this type application, the specific system implementation which has been considered in most detail during this study assumes the use of a drum.

The computing system consists primarily of a number of semi-autonomous processing units interconnected through the medium of a drum memory. The configuration of the system is illustrated diagrammatically in figure 5-1. In this system each processing unit picks up from the drum a set of words and an associated instruction (or set of instructions) which specify the function to be performed on the data. Having picked up the data and the instructions, the processor inhibits its 'read' head until the processing is finished and the results have been written on another track. The results are accompanied by a tag identifying the subsystem which last processed that block of data. In the redundant configuration this track is known as the "vote" track. In the particular system illustrated there are three vote tracks. The subsystem now reads the next set of data and instruction and continues processing. Each set of data is processed by three different subsystems and the results are placed on separate tracks. The voting circuit does not operate until the triplicated sets of results are produced. A majority vote is performed when the triplicate set of results is available. The output of the voter is sent to a central programmer. If one subsystem has failed and produces incorrect data, the voter will detect this and inhibit this subsystem from performing that particular function again.

The example shown in figure 5-1 will perform ten different functions on the data in any order (with repetitions or iterations possible.) The order is chosen by the central program which sends the result of the voting to a location on the drum associated with the next function to be performed. The subsystems which will subsequently perform the function will depend upon which are free when they scan that location.

The weakness of such a system lies in the central programmer and its associated switching circuitry. This may be made more reliable by having three copies and voting or by other methods. However, a more ideal solution is to abolish the central programmer altogether. This may be accomplished by a variety of methods, leading to the system in figure 5-2. In this version of the system, the voting is a function of some portion of the subsystems. The flow of data through this system is controlled by a type of list processing, where each instruction includes the address of the next instruction, or each set of data may carry an additional 'tag' which instructs the subsystems as to the next function to be performed.

One greatly simplified memory arrangement will be used to illustrate some of the considerations (figure 5-3). Track 1 is a timing track. Each unit picks up the clock pulses from this track and hence its position in storage. The pulses also act to operate instruction sequences within each unit, to trigger shaping pulses, etc.

Track 2 is the program track, which stores the basic program. The program is read in by a special input which prevents any processing units from writing on this track, thus insuring that the fundamental program is protected in spite of all failure modes in the units. This program contains all the instructions and constants necessary for the processing. The inability of the units to modify the basic program does not limit program versatility; it would be possible to transfer parts of the program to another track (where they could be modified,) and temporarily transfer control of the unit to this track. If some failure now occurs, a sub-

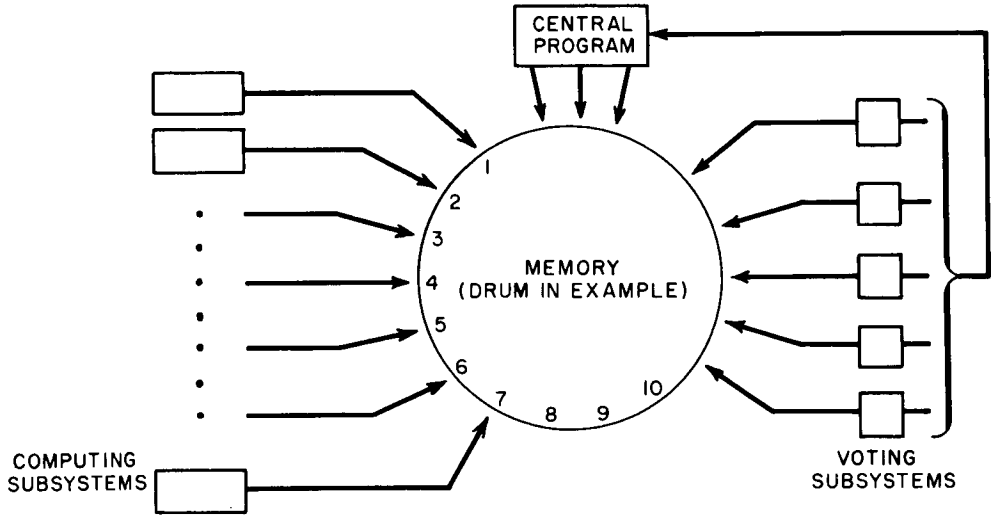


Figure 5-1. System Organization Diagram

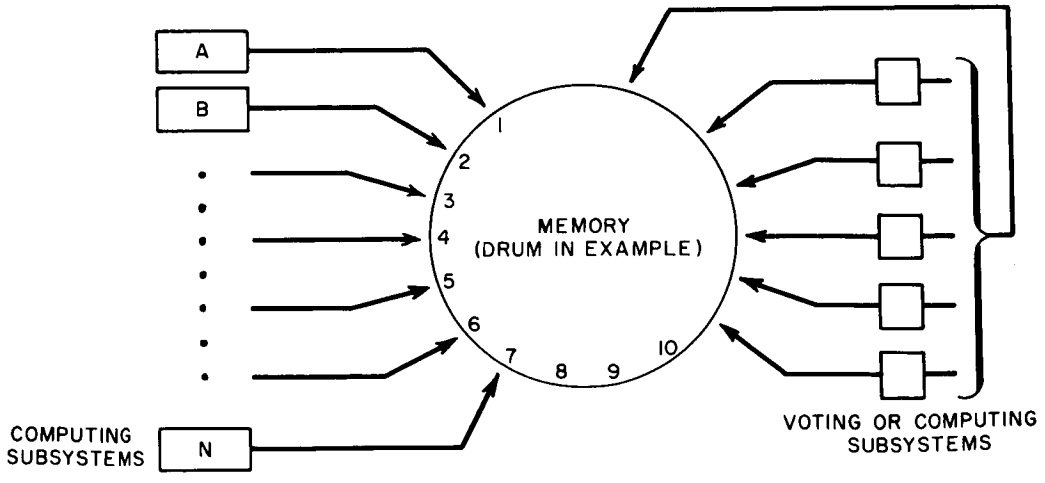


Figure 5-2. Revised System Organization Diagram

system is lost but not the central program. Another desirable feature to build into such a program is the periodic sensing of a track under the control of the operators console. If the computer is on a space mission this sensing could be controlled from a remote console. This permits the operator to have some control over the processing units, although not absolute control, and he cannot destroy the main program by errors in transmission (which have high probability in deep space communication). The control of a unit would temporarily be transfered to the operators track if the program sensed the transfer signal. Transfer back to the main program would occur automatically after a number of instructions. Hence a communication error may result in temporary disability of a unit, but has no effect on the main program, or other units.

On Track 3 the certified data is stored. This is the output of the voters, and the input to the processors. It is always assumed to be correct.

Tracks 4, 5 and 6 contain the uncertified data. This data is the output of the processors, and the input to the voters. A voter will read these tracks simultaneously and transfer the result which is 'voted correct' to the certified data track. Each unit would normally read and write on only one of the vote tracks (4, 5, 6), unless the unit was functioning as a voter. This divides the units into three classes. Some or all of the units would have the facility to change classes, otherwise the system would fail as soon as all units in one class failed with no possibility of transferring a good unit from another class.

As the system is now conceived, the individual subsystem may perform a series of functions in a prescribed sequence or the subsystems may form a complex queue to perform the functions as determined by the availability of certified data. For example, series of functions (F_1, F_2, $F_3 \ldots F_n$) are to be performed on the data words (D_1, D_2, $D_3 \ldots D_n$). Referring to figure 5-2, the first subsystem to become idle (Subsystem A) will compute $F_1(D_1)$. The next two subsystems to become idle will also compute $F_1(D_1)$ and store the results on the vote tracts. During the computing time, a fourth subsystem will probably have started computing $F_1(D_2)$, and subsystem A will have completed $F_1(D_1)$ and moved another function or another data word, e. g. $F_1(D_2)$. As soon as the required number of copies of $F_1(D_1)$ have been computed, this data will be voted on and transferred to the certified data track. Once it is on the certified data track, it is available for subsystems to begin the computation of $F_2(F_1(D_1))$. This general process will continue until all sets of data have been channeled through all the functions.

C. SUBSYSTEM FUNCTIONAL CAPABILITIES

1. Typical Subsystem Operational Capabilities

The exact functional capabilities of the individual subsystems have not yet been defined. It has been established, however, that each subsystem will probably have to have the

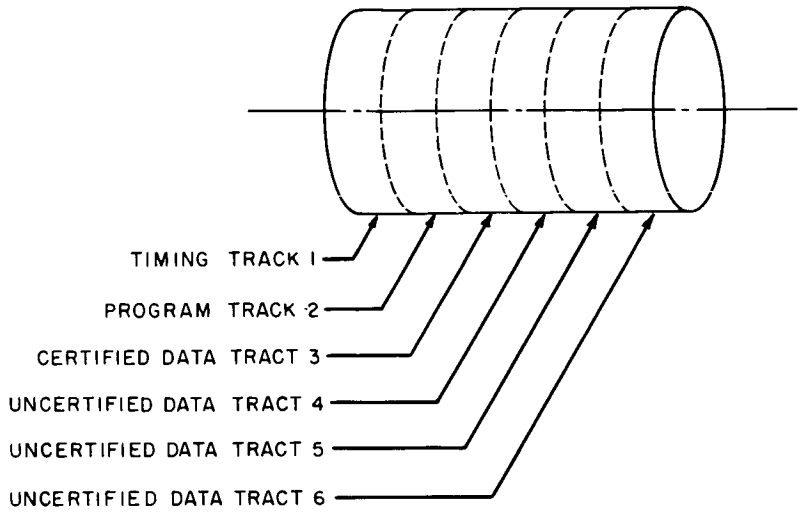


Figure 5-3. Simplified Memory Arrangement

capability to perform three or more of the following types of functions:

a. Read In
b. Read Out
c. Add
d. Subtract
e. Compare
f. Delay (or Transfer)
g. Interpret

The first four instructions are self explanatory. The next instruction, COMPARE, enables the programmer to introduce conditional branching into the program, and could also be designed to implement voting. The DELAY instruction enables the transfer of information -- a choice of addresses.

The INTERPRET instruction tells the processing unit what function to perform, apart from instructions 1 - 6. The function may be in the form of a subroutine on another track, in which case the INTERPRET instruction is a transfer of control. The INTERPRET

instruction may also command functions permanently programmed into each unit, such as "multiply" and "divide".

2. Voting Implementation Alternatives

The operation of a system employing subsystem redundancy requires the use of restoring or voting networks to determine the best estimate of a signal based on the examination of several nominally identical copies of the signal. In most serially operated digital systems, the voting network takes the form of gates whose instantaneous output states conform to the existing states of the majority of their inputs. Because of the serial generation of signals, the voting is performed on "bit-by-bit" basis. In the systems proposed here, the voting will be performed on a "word"-by-"word" basis in any one of the number of ways which are described in Section E below.

The point of interest here is that the voting function may be implemented in either of two ways. A set of special subsystems may be added to the system to perform the function for all of the data generated by the other systems, or performing the vote function may be one of the normal activities of all or a subset of the computational subsystems. At this point in the development, it is not apparent which of these two alternatives is best. It is, in fact, highly probable that the choice of voting implementation will depend on whether the system is always used in the redundant mode of operation, upon the relative speed of the voting process compared to the average computational speed, and upon the complexity of the specific voting function being used.

D. VOTING SCHEME ALTERNATIVES

Many methods of voting are easily implemented using a memory medium. One very economical method is to produce only two copies of each function. These are compared by the voter and, if there is agreement, the result entered in the certified date list. When disagreement occurs a third copy is produced and a normal 'two out of three' vote is taken. Since disagreement is infrequent, most of the functions need only be duplicated, which means that with a given number of subsystems there is about a fifty percent increase in processing capability.

Another type of voting to consider is adaptive. This results in the best 'decision' when there are more than three copies of a result. When working on a bit-by-bit basis the vote in the ideal case is found by associating a weight, $w = \log \frac{1-p}{p}$, with each subsystem (or with each function of each subsystem) where p is the probability of failure of that unit. Hence, w increases if the subsystem is reliable and decreases when the subsystem makes frequent errors. A similar law would apply when comparing 'words' rather than 'bits'.

Although the choice of schemes for comparing data bit-by-bit is fairly limited, the range of techniques for the analysis and comparison of data words is quite broad. The final choice of optimal voting schemes for this type system will compose a study in itself to evaluate the trade-offs in voting speed, equipment required and vote accuracy.

E. OPERATIONAL MODES

In the preceding sections, the discussion of the medium communication type systems has been restricted to those systems operating in a redundant fashion. Although the primary objective of this study did not include the investigation of new implementations of non-redundant system, the technique which has evolved also offers potential advantages to the user of non-redundant systems. This technique also facilitates using a single system alternately in redundant and non-redundant modes or in a combination mode where only certain vital functions are performed in the redundant mode. The desirability of this mode versatility is illustrated by the following example.

Consider a typical set of computer applications on a space mission. Before launch the computer may be used to check out all test points and report any failures. During launch the computer, besides monitoring many test points, may be used for real time control of the rocket motors and guidance etc. Later in the flight path, the computer may control the guidance under direction from a ground base to set it accurately in course. During the major part of the life of the computer, it will probably be used mainly for data monitoring, for processing incoming data (from sensors on the spacecraft) and sending the statistics back to ground. The need for highly reliable operation varies drastically during the course of the mission. Moreover, when the computer becomes incapable of performing the processing for all sensors, it is better for it to continue processing fewer sensors rather than none at all. It is therefore preferable to have a computer system which gradually decreases in capability due to failure of its components, than a computer with constant capability which at some point fails completely.

The control of the operational mode of the computer can probably be made a part of the stored program. As a result, the reliability of the system can actually be controlled by the user of the equipment and only the upper limit on reliability is set by the equipment designer.

5-3. AREAS OF FUTURE STUDY

The objective of the study reported here has been to begin the investigation of the advantages, disadvantages and the feasibility of implementing redundant computing systems whose individual subsystems communicate through a common medium. Although this objective has been achieved, a much more extensive study of the system design alternatives

must be carried out before the net balance of advantages and disadvantages can be evaluated or before such systems could actually be implemented. Some alternatives which should be considered in detail are described below.

1. The Complexity of the subsystems. It is possible to build subsystems which can perform a large variety of complex functions. By merely entering a code word from the memory, the sybsystem may change from performing a function like ($\sin^2 x$) to performing one like ($\sqrt{x}$). With such subsystems, the program on the medium could become very simple. At the other extreme, the subsystems could be very simple and the program on the medium very detailed. The latter arrangement would probably lead to a lower component count, but increases the complexity of the programming and the size of the storage required.

2. The Addressing and Programming of the System

One method of programming is to add a "tag" to each set of data, to identify the next function performed on that data. This procedure may require a search for data bearing corresponding 'tags'. Under those conditions, a form of content addressable memory is required. Hopefully, however, a more conservative type of programming would be possible where the address of the required data would be known. A third alternative type of programming would be 'list processing', where each instruction would contain the position of the next instruction. As a fourth alternative, the subsystems may follow a predetermined sequence of instructions, i. e. all the functions it is to perform under given conditions are determined by the programmer, who must, therefore, foresee all the possibilities.

The individual subsystem may also have its own stored program, in which case the programmer could initially program the functions that the subsystem would perform. In operation the program stored in the subsystem would not normally be changed. It might not be desirable for all the subsystems to be able to perform all the functions, but it would be desirable to be able to redistribute the functions among the subsystems when another problem is programmed.

3. The Memory Hardware

The system which has been described in detail uses a drum memory. However, a similar system could be arranged with other implementations of the memory medium such as a magnetic core or magnetic thin film storage units. In such a case the mechanical scanning (inherent in the rotation of the drum) would be replaced by electronic scanning. In many cases this latter type of memory would be required to eliminate the undesirable effects of using moving parts in a spaceborne system. This move to electronically scanned memories may not be necessary as is shown by the fact that magnetic drum storage has been successfully

used in many computers. Certainly the drum has many advantages including large capacity and medium access time storage using relatively simple equipment.

In the ultrareliable system described here, the reliability of the memory unit would be as important as that of the arithmetic and other logical functions. Also, because the concept of using a 'medium' is based upon having a medium which is simple to address, the implementation by core storage and other means and the concomitant electronic scanning design merits serious study. Alternative solutions also include optical techniques. The optical techniques would provide the electrical isolation between units which is highly desirable and which is provided in the present system by mechanical scanning.

4. Voting

Error correcting methods other than voting should be considered in this system. The system should lend itself well to error correcting codes, the use of special test problems. A combination of voting and coding might be the best alternative. The implementation of the voting is an area of study in itself. Assuming a n-line voter, each process must be repeated n (and only n) times, and each time a different subsystem must do the process. A way of doing this has already been suggested - having three Vote Tracks with each subsystem recording its result on one of the tracks. It will probably be necessary for each subsystem to index its results so that the erroneous result may be associated with the subsystem which produced it and the subsystem treated accordingly.

5. Failed Subsystems

When a failed subsystem has been identified, the question of what to do with it remains. If a system of adaptive voting is being used,then the 'weight' associated with the subsystem would be reduced. It is possible that although the subsystem failed in one of its functions it may still be able to perform the remainder of its functions reliably. Hence, it is desirable to identify not only the subsystem but also the function it performed erroneously and inhibit it from performing that function again. It is this type of arrangement which makes it difficult for the programmer to foresee all the possibilities and indeed, to program routines for all the possibilities were they foreseen. If however, the subsystems could independently decide which functions they are still capable of performing and choose to do those functions when they arise in the main program, then the main program can be comparatively simple.

6. Special Failure Modes

It is possible for the subsystems to fail in many ways. One way which deserves particular consideration is when the write head fails in such a manner as to erase all the information on a track or to write meaningless information. The first approach to solving this

problem would be to try to design a write head which is fail-safe. If the failure occurred right at the head (by an electric short circuit, for example) then the normal switch to turn the head off would probably be ineffective and it may be necessary to cut off the power to the whole unit.

Another approach to solving this problem is to provide each unit with parallel write heads on many tracks. If now one of the heads fails, the particular track it is on is considered a loss (or its weight is decreased - if we associate weights with tracks and carry out a vote), and the unit it belongs to may still be used in conjunction with other tracks.

Using multiple tracks and associating voting weights with the tracks as well as the units also overcomes the problem of a track becoming inferior because of dirt or scratches etc.

5-4. CONCLUSIONS

This task has been an exploratory investigation of possible organizational structures of systems which can easily be reorganized to continue operation with a relatively high percentage of failed components. The investigations have led to the formulation of a general type of computer organization which fulfills the objective of this task. The computer is a multiprocessor in which each subsystem communicates with all other subsystems through a common memory medium. The medium also stores any information which each subsystem would normally contain in any storage which was not controlled by the inputs. Subsystem outputs are stored (and voted on) in the medium.

The many advantages of such an organization provide convincing evidence that the organization merits further study. These advantages may be summarized as:

1. High Subsystem Mobility

a. The prime advantage is that this type system offers one means for realizing the potential benefits of failure-responsive systems. Indeed, in the system where each subsystem can perform all the functions, the maximum "mobility" of the subsystems has been achieved because every subsystem may replace any other subsystem as the failure pattern occurs. In this system two of the main restrictions on mobility have been removed. The first restriction is that all subsystems perform the same function. Subsystems are proposed which have the capability to change function when they change position. As a result the homogeneity difficulties inherent in fixed function subsystems do not arise. The second major obstacle to mobility occurs when a subsystem contains a fixed memory - i. e. a memory not set up within a few cycles by the data stream. Subsystems which are otherwise identical but which contain different information in their memory are not interchangeable.

However, using a memory medium, memory which was formerly a part of the subsystem is now a part of the medium. Any subsystem may now associate itself with the part of the medium containing this memory and perform in this position just as well as any other subsystem.

b. Graceful Degradation

Graceful degradation of the system performance is inherently available in the system. This concept of graceful degradation assumes that the system is not used for only one purpose at a fixed data rate. In that case a definite capability would be required. Having greater capability would be wasteful, and having less constitutes complete failure.

The system proposed has this desirable property. For example, the system may first have twenty subsystems. If each subsystem is identical - can accomplish all the functions, then eighteen subsystems can fail (assuming two out of three voting) and the computer will still be able to do everything that was possible initially, but take ten times as long to do the same task.

c. Optimal Non-Redundant Operation

One of the most common objections to redundant systems is that they use three times the number of components without increasing computer capability. On the other hand, it may be desirable to operate three computers in parallel when failure is very expensive. As this implies, having three individual computers gives one a choice between capability and reliability. This choice is available in an even more useful form with the medium system. By reprogramming the system the subsystems may do each process only once, increasing the power of the computer by a factor of two or three. This option may be very desirable for ground testing equipment before take off, or in any mode where reliability is not as important as speed.

Operation in the non-redundant mode is not alien to the system design and interleaving non-redundant and redundant operation is quite possible. This may be done even in the same computation if certain parts are not as significant as others.

d. Asynchronous Operation of Subsystems

The units in a processing system with memory may operate asynchronously. This relieves the programmer of timing problems and increases the efficiency of computation, since subsystems need not wait for each other. As this implies, the memory serves both as a central medium and as a buffer store for each subsystem.

e. Efficient Use of Time Shared Subsystems

Because the subsystems are not restricted to a single functional location, but rather they perform the next in a sequence of functions as they are needed, the subsystems may be shared between different problems. Priority interrupt is effected by placing the interrupting routine within the main program. The subsystems operate in parallel; hence, each subsystem is used to its full capacity.

A new development program such as this creates many new study areas. One approach to developing the organization in more detail would be to assume some properties for the subsystems and then write the programs to make them function as desired. This configuration could then be simulated on a general purpose digital computer. Such a procedure would insure that realistic solutions are found in each problem area.

APPENDIX A

Previously Published

As

Technical Report No. 5

On

Analysis and Development of Failure-Responsive System Organizations

Contract NASw-572

N65-17605-NASA CR 60897

by

C. C. Masters, Jr.

December 1964

Note: This report is presented in the form of a thesis. As a thesis, the report was submitted by the author to the University of Pittsburgh in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering in December 1964.

The Westinghouse Electric Corporation
Electronics Division
Box 1897, Baltimore 3, Maryland

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

I. INTRODUCTION

A. The Need for High System Reliability

Electronic digital data processing systems have become an integral part of the modern world. These systems are commonly used to perform tasks which were thought unachievable only a decade ago. The great computational capabilities and operating speeds of today's data processors have usually been obtained at the cost of extremely high equipment complexity. This complexity naturally results in low system reliability. This, in turn, limits the usefulness of the equipment to the extent that a paradoxical situation threatens to emerge in which system capability is extremely high but it is almost never available for use.

In addition to the problems caused by loss of operating time, high system complexity and the necessity of frequent complicated repairs aggravate the problems of supplying spare parts and properly trained maintenance personnel. These problems become increasingly troublesome as large systems are put into use at remote locations. The natural environments for most military field and shipboard equipment are sufficiently remote to make the logistics problems dominate over almost all other considerations. The limit in this area is reached by spaceborne equipment where logistics become virtually impossible.

The necessity for high system reliability may also be dictated by the vital nature of the system functions as well as by an interest in maximizing system usefulness or minimizing liaison problems. Quite often control systems, for example, are relatively simply in comparison to large scale data processing systems, but their continuous operation may be of vital necessity for the safety and security of an individual or a nation. The list of applications of this class includes space vehicle "on-board" controls systems,

atomic reactor controls, missile guidance and destruct systems, and secure communications systems.

B. Methods of Increasing System Reliability

1. Conservative Design

One of the first methods that design engineers successfully used to increase system reliability was that of derating electronic components. Using this procedure, circuits are designed with components of much greater power and voltage rating than the specific circuit applications require. In operation, these components are subject to such low thermal and electrical stress that their expected life approaches "shelf-life". This method has proved to be a relatively cheap and effective means for increasing average system life.

2. Hyper-reliable Components

A second method, which has been equally successful, involves the use of special manufacturing procedures to produce more reliable components. This method employs refined fabrication techniques and a supplementary program for individual component testing. The testing program is used to monitor various characteristics of the components during the manufacturing procedure such that any defects can be detected before the product reaches the consumer. This approach to achieving high system reliability has been championed by the Air Force's Minuteman Missile program. Although significant reductions in component failure rates have been realized through the use of this technique, the effort appears to be reaching a point of diminishing returns where each level of improvement is becoming more and more costly to achieve.

3. Coding

An entirely different approach to the problem of achieving high reliability has been found in the use of coded signals. This approach is useful in binary data transmission and storage systems where the primary interest is that of maintaining the accuracy of existing information. In using this technique, the information to be transmitted or stored is broken up into sections called "words". Each of the words is subsequently analyzed to determine one or more of its characteristics. For example, a characteristic which is commonly of interest is the number of ones appearing in the binary word. The results of the analysis are converted to binary data, and this latter data is then combined with the original word to form a complete message unit. Depending on the complexity of the code, single or multiple error detection or correction can be performed when the message unit is decoded following transmission or storage.

In general, this technique is not applicable to systems which perform any function other than data transmission or storage. This limitation exists because any arithmetic or similar function destroys the integrity of the code by altering the message units.

4. Redundant Equipment

Several methods for achieving high system reliability through the use of redundant equipment have also been used. One relatively simple technique has been used for decades in the form of stand-by facilities. Using this method, auxiliary equipment is switched into use in the event of primary equipment failure. Most implementations of this method are extremely costly relative to the failure protection which they provide. For example, one unmaintained primary system and an unmaintained duplicate standby can only absorb one failure in each system before they both become inoperative and the

and the system function is lost. Using more sophisticated techniques, however, it is not unreasonable to expect that equipment which is replicated three or four times might absorb several dozen failures before the system function is lost.

The following section describes the basic types of redundancy techniques which have been developed. The more troublesome disadvantages of these techniques are included to provide a basis for the study reported in the remainder of this thesis.

C. Redundancy Techniques

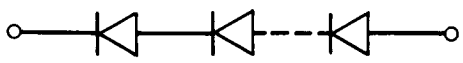
The new techniques which have been developed for systematically introducing redundant equipment into data processing systems can be separated into two general classes: (1) component replication; (2) subsystem replication. It has been shown that the redundant equipment employed in a fixed system configuration is most effective when the system is divided into the smallest divisible units. Because the individual circuit components usually represent such units, this implies that component redundancy is the most efficient technique which can be employed. In attempting to implement redundant systems of this type, however, several problems immediately arise which suggest that this form of redundancy is not always compatible with other system design considerations.

Component redundancy is applied by placing several replicas of an electronic component in a series or a parallel configuration or some combination of the two. Examples of each type configuration is shown in figure 1. These configurations are often much more reliable than a single non-redundant component because more than one component must fail into its detrimental mode (i. e., open or short) before the circuit function of the component is completely lost, and the system fails. For example, if a certain type diode always fails to a short mode, placing two or more of them in series as shown in figure 1a will protect the circuit from failure until all of the diodes in the

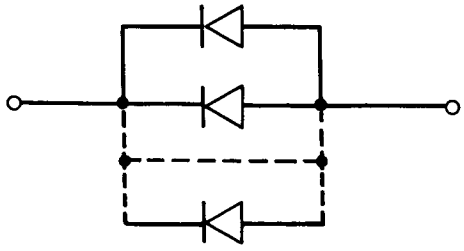
chain fail. A similar protection is provided against open circuits by paralleling components (figure 1b) or against either mode through the use of quads (figure 1c) or larger Hammock Networks (figure 1d).

It is apparent that such a technique for introducing redundancy cannot be applied to components where the actual values of the components are critical to the operation of the circuit. The failure of individual components in these configurations may easily change the impedance of the network by fifty per cent. Although most digital circuits are not particularly critical to impedance changes, many types of circuit applications are sensitive to changes of this magnitude.

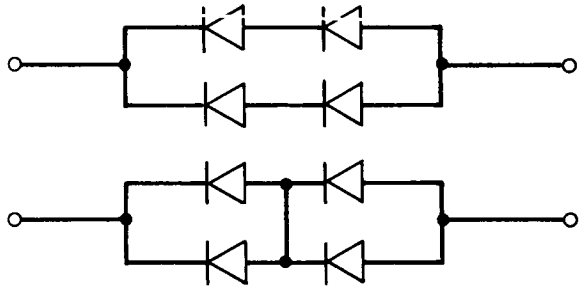
In applying this type of redundancy, the assumption is made that the failure of one component is virtually independent of the operation of any other components. In systems using thin film or molecular-electronic circuits, it has been found that failures of components deposited on the same inactive base or included in the same semiconductor block are highly correlated. This means that in order to achieve even a rough approximation



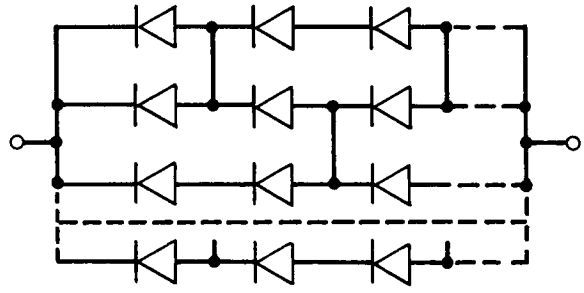
a) SERIES REDUNDANT COMPONENTS



b) PARALLEL REDUNDANT COMPONENTS



c) QUADDED REDUNDANT COMPONENTS



d) HAMMOCK NETWORK OF REDUNDANT COMPONENTS

Figure 1. Redundant Component Configurations

to component independence, components in the same redundant network would have to be deposited on different bases or blocks and connected together with additional wiring. The unreliability of interconnections between these circuits would usually offset the gains sought through redundancy; therefore, a different class of techniques must be used for introducing redundancy into most microminiaturized circuits.

The second class of techniques, subsystem replication, can be subdivided into two significantly different subclasses. In the first of these, the "sense and switch" subclass, two or more nominally identical replicas of a subsystem are monitored and controlled by a monitor and control network. Based on some predetermined operational criteria the network locks the output of the stage[1] to the output of one of the subsystem replicas until a failure in that subsystem is sensed by the monitoring circuitry. At this time the control portion of the network attempts to switch the stage output to a working replica if one is available.

Although this technique is particularly useful in analog systems, it is very difficult to calculate the quality of a digital signal without comparing it to another nominally identical signal. Because of this, the sensing circuits must be very elaborate to capitalize on the advantage of one out of (n) replica operation. This is troublesome because this type operation is the major advantage derived from techniques of this subclass.

The second subclass of techniques for this type of implementation of redundant systems might be called the "voted" techniques. Of the several techniques in this subclass, the "multiple-line" method of implementation appears to be the best. Figure 2b shows basic topological characteristics of a segment of a multiple-line system. A non-redundant version of this equipment would consist of three single input, single output subsystems connected in series as shown in figure 2a. To form the redundant version

---

1 A "stage" consists of all of the subsystem replicas and any associated circuitry required to provide a redundant replacement for a subsystem in a non-redundant system.

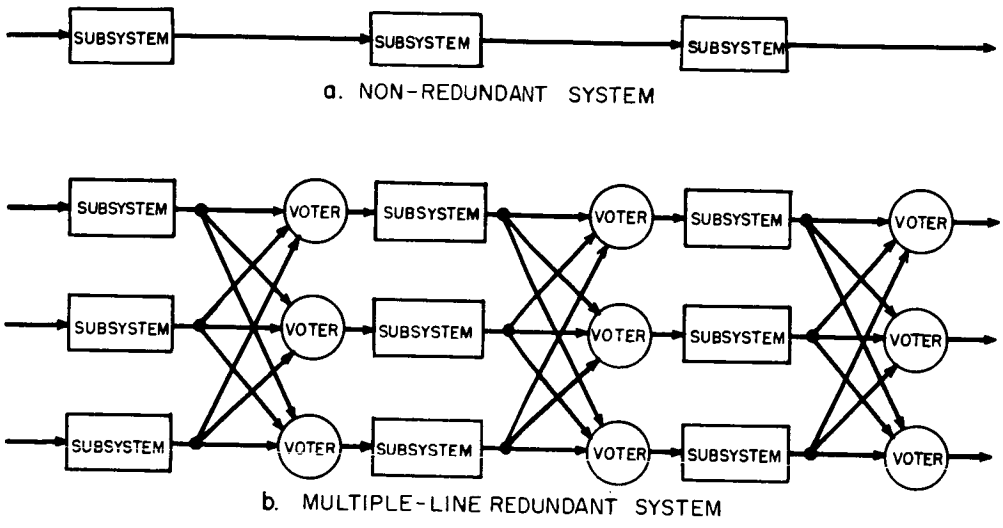


Figure 2. A segment of an Example System

of the equipment, each subsystem has been replicated twice and voting circuits (or voters) have been inserted between the sets of subsystems. The use of three subsystems to replace one from the non-redundant version results in an "order-three" system. Similarly, the use of five to replace one would result in an "order five" system. The voters are usually majority logic gates. The voters may, however, be designed to vote on some alternate threshold level. This would be done if information were available to indicate that the generation of erroneous ones is much more likely than the generation of erroneous zeros or vice versa. The replication of the voters is necessary to prevent system failure because of single failures in the voters themselves.

Several investigation teams [1], [2], [3], [4]* have studied this particular type of redundancy and found it to be applicable to a broad range of digital systems. Under the names of "Multiple-line, Majority-Voted Redundancy" and "Triple-Modular Redundancy"

* Parenthetical references placed superior to the line of text refer to the bibliography.

and possibly others, it is currently being considered by various groups for inclusion in the design of the digital portions of spaceborne equipment associated with several projects including Ranger and Saturn.

The primary disadvantage of systems of this type is that they are vulnerable to certain improbable but destructive failure patterns which may disable the system while most of the redundant equipment is still operational. One of these patterns will occur anytime two of the first few component failures happen to occur in different replicas of the same stage of an order-three system.

II. THE PURPOSE OF THIS THESIS

The techniques described above provide a variety of means for employing redundant equipment to increase the reliability of electronic digital systems. Although these techniques are effective in accomplishing the desired increases, they do not make as efficient use of the redundant equipment as would seem possible.

In this thesis, the author proposes to present the concept of a new technique which the author has developed for more efficiently using redundant equipment to increase the reliability of one class of digital systems. In addition to developing this concept, the author proposes to show that this technique is, in fact, more efficient than the comparable existing technique. The comparison of the new and the old techniques will be made through the use of results obtained from a computer simulation program which the author has developed for this specific purpose.

III. PREVIOUS WORK IN THIS AREA BY OTHER INVESTIGATORS

The use of redundant equipment has interested a relatively large number of investigators in both academic and industrial environments. The publications which have been produced by these investigators are too numerous to list here; however, a bibliography which lists over one hundred of these publications was published by P. A. Jensen [5] in 1962. The majority of this work has been concentrated on the analysis and development of fixed redundancy techniques.

Only a very few investigators seem to have seriously considered systems which are in any way similar to those of interest in this investigation. The most notable work on this latter subject appears to have been done by E. J. Kletsky [6] and S. Seshu, [7] at Syracuse University Research Institute and L. Lofgren [8], [9] at the University of Illinois Electrical Engineering Research Laboratories. Kletsky and Seshu worked as a team under a Navy contract while Lofgren simultaneously conducted an independent study for the Air Force. Both Lofgren and Kletsky were interested in developing mathematical models which would describe the expected life of systems that draw up spares from a common "pool" to perform any necessary subsystem repairs. Although Lofgren's work is generally more abstract than Kletsky's, neither of them was particularly concerned about the problems of implementing such systems. In one paper, however, Lofgren did propose a fluid flow technique for performing the subsystem replacement function. This technique is itself fraught with many problems, but it certainly represents an ingenious contribution to the art. At least one other investigator, R. R. Landers [10] has attempted to extend the fluid flow technique to a more nearly realizable state.

Seshu suggested two possible techniques for implementing systems of the general type that Kletsky was studying. In considering implementation, he immediately recognized the problem associated with detecting errors in systems employing a non-redundant on-line structure supplemented by a pool of spares. He proposed two feasible

implementation techniques. In one technique, he suggested that a central controller be constructed to monitor the remainder of the system and to perform any necessary subsystem replacements. As an alternative, he proposed to have a ring of subsystems with each subsystem monitoring and controlling one of its neighbors.

The system organizations described in this paper have the same general objective, i.e., long system life, as the self-repairing systems which were considered by Kletsky, Seshu, and Lofgren. The organizational structure of the systems described here, however, are much more closely related to presently practicable digital systems than are those of the limiting cases considered by the above authors. Because of this difference between the organizational structures, this new work does not appear to be an extension of any of the other author's work.

IV. FAILURE RESPONSIVE SYSTEM ORGANIZATIONS

A. The General Concept

A "failure responsive system" is a redundant system which has the capability to partially reorganize itself to combat the detrimental effects of internal subsystem failures. Before any subsystems have failed, failure responsive systems closely resemble the multiple-line redundant systems which have been previously described. Within these systems each subsystem is also replicated several times, and each replica in each stage is supplied with a set of the inputs associated with the stage. The outputs are fed into a switching network and used to determine the best estimate of the correct stage output in a manner similar to the voting circuits of the multiple-line systems. These systems resemble the multiple-line systems until one of the stages experiences multiple subsystem failures. When this condition occurs, the switching network for that stage signals for a partial system reorganization. This reorganization consists of the elimination of the failed subsystems, and the functional movement of other subsystems through the switching of their input and output connections. The result is the restoration of the system to an operational state. This process is continued as long as enough subsystems remain operational so that the reorganization action can effect the necessary restoration. It should be noted that the reorganization should not change the functional operation of the system. It only changes the distribution of the redundant subsystem replicas. As this statement implies, the subsystems which take part in the reorganizations are functionally identical so that any one can be substituted for any other one.

As an example of a typical series of operations, the reorganization actions of one system as it would respond to one particular failure pattern, will be considered. The system which will be considered is presented by the pattern of blocks shown in figure 3. This pattern of blocks represents a seven stage, order three, failure responsive system. The non-redundant version of this seven stage system would be similar to the three stage

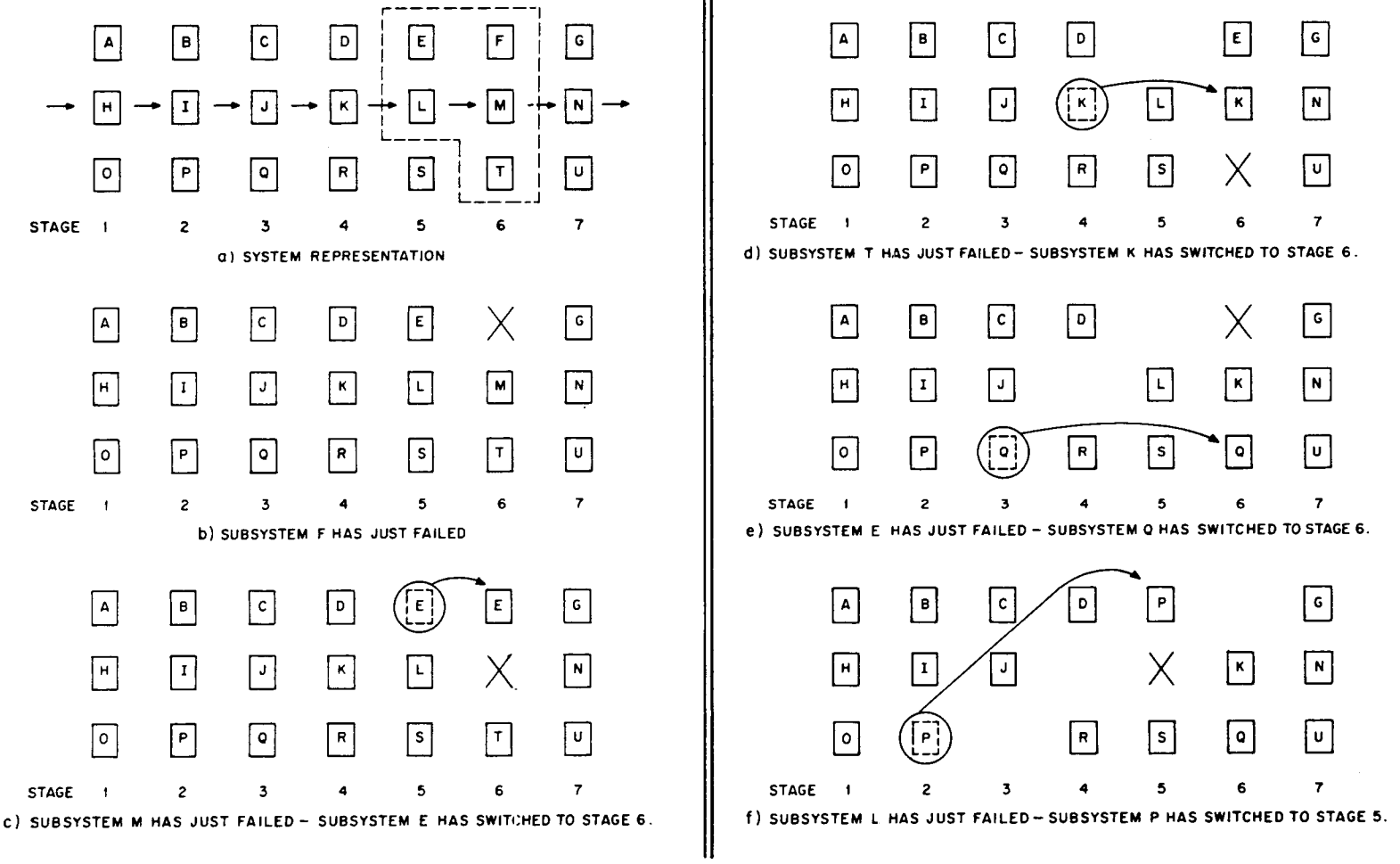


Figure 3. Example Failure Pattern

system illustrated in figure 2a. In this case and in the figures which follow, the blocks in the diagram represent individual subsystems. The physical position of the blocks represent the relative functional positions of subsystems within an electronic system. It should also be noted that the peripheral switching circuits required to implement the various systems have not been shown.

Referring to the letter code shown in figure 3a, the following series of subsystem failures are assumed to have occurred: F, M, T, E, L. Note that this series includes all of the subsystems enclosed by the dashed lines in figure 3a. Using a preprogrammed response strategy, the system would react to this pattern of failures in the following manner:

1. When F failed, its output would be permanently turned off. No other action would be taken. (See figure 3b.)
2. When M failed, the ambiguity caused by the failure of one of two nominally identical subsystems, M and T, will cause one of the working subsystems from another stage to be switched to stage 6. In this case, subsystem E will be moved up one stage. With E and T now properly performing the function of stage 6, the ambiguity existing between M and T is resolved and M is turned off. (See figure 3c.)
3. When subsystem T fails, an identical procedure will be used to call K to stage 6. Again the ambiguity existing between working subsystem E and failed subsystem T will be resolved, and T will also be turned off. (See figure 3d.)
4. When E fails, processor Q will be moved to stage 6 and again the system will be restored to operation. (See figure 3e.)
5. The subsequent failure of L, will result in subsystem P being moved to stage 5. This will restore the stage and the system to operation. (See figure 3f.)

This example was specifically chosen to illustrate the conceivable range of variation in response strategies as well as the potential power of the failure responsive technique.

The first of these two is evident. For example, although a definite response strategy was employed in the above example, it is not necessarily obvious to the reader what the strategy was, even after observing the effect of several failures in the same general location. As for the second item, it is obvious that the system would have failed after the third failure and quite probably after the second failure if the multiple-line majority voted technique were still being employed. With the failure responsive reorganization capability, however, the system has withstood five consecutive failures in a tightly grouped pattern without suffering system failure.

B. The Specific Organizational Objectives

One of the primary objectives of this study has been to develop a set of design rules for failure responsive systems. These rules are intended to serve as guidelines for facilitating system designs which will make very effective use of the redundant equipment, subject to switching network unreliability and various instantaneous failure masking requirements[2]. To establish a meaningful set of rules, a wide variety of feasible response strategy characteristics had to be considered to determine which characteristics were necessary, which were only desirable and which were undesirable. These characteristics include the following:

1. The number of replacements which should be available to any one stage. (The assumption is made that the addition of replacements results in an addition to the peripheral switching circuitry.)
2. The pattern for specifying which subsystems should be used as the replacements for any particular stage and the order in which they should be called.
3. The use of fractional order of redundancy, i.e. not every stage being the same order in the initial state.

[2] "Instantaneous failure masking" means that a subsystem failure in any stage is completely masked by that stage so that no errors propagate through the system during the time the system is reorganizing itself to eliminate the failed subsystem.

4. The use of minimum order of redundancy to be maintained at a stage from which a failed stage would like to take a replacement.

5. The capability of a vulnerable stage to override the minimum of number (4) in the event no replacement is otherwise available.

6. The capability of a single subsystem to make more than one change of location.

These and other response strategy characteristics have been considered during this study. The relative importance and desirability of all of them are reflected in the conclusions presented in section IX.

V. ANALYSIS METHODS

To evaluate failure responsive systems and compare the effectiveness of various response strategies, a method had to be found for determining the reliability of these systems. In the case of multiple-line redundancy in which the functional locations are static, various analytical techniques have been used to express reliability. The problems presented in the following paragraphs indicate that the techniques used for analyzing fixed redundant systems are not generally amenable to failure responsive systems.

Before proceeding with the description of the problems involved in applying analytical techniques to failure responsive systems, it should be noted that all of the systems considered will be limited to those of simple unilateral signal flow with single inputs and single outputs at each subsystem. It is also assumed that all stages are identical; therefore, all stage reliabilities are equal. Although such systems are obviously idealistically simple, any more realistic modifications in the models would only serve to complicate the existing problem or increase the overall number of problems.

A. The "Brute Force" Method

As stated above, the assumption has been made that all stages are identical. This statement implies that the system reliability, R_S, can be expressed as

$$R_S = (R_{ST})^N \tag{1}$$

where R_{ST} = the stage reliability,

N = the number of stages in the system.

Because N is always known, the only significant problem is the determination of R_{ST}. For a system employing fixed redundancy, this problem is easily solved by enumerating the number of failure patterns which can exist within the stage and still permit stage

operation. The computation is completed by summing the probabilities that each of these patterns will exist. For example, the reliability of a stage in an order-three, majority-voted multiple line system is given by:

$$R = (e^{-\lambda t})^3 + 3(e^{-\lambda t})^2 (1 - e^{-\lambda t}) \quad (2)^3$$

This problem is not so easily solved in the case of failure responsive systems. The mobility of the subsystems in these systems suggests that the enumeration of operating states must be performed on a complete system basis rather than be restricted to an individual stage. This approach is complicated by the fact that many response strategies are sensitive to the order in which failures occur as well as the particular locations of the failures. The number of possible operating states and the permutations of failure orders combine to make the overall reliability computation process too lengthy for practical use.

B. The Markov Chain Method

The changes in system operating states caused by subsystem failures may be regarded as transitions between states in a Markov chain. The formulation of this reliability analysis problem as a Markov chain automatically provides a group of solution methods which are not otherwise available.

Before proceeding with the analysis, however, it would seem wise to consider the size of the Markov transition matrix which would be required for the systems of interest. A typical system might have as many as a hundred or more stages in it, but to be conservative a ten-stage system will be used as an example. The number of possible operational states of a ten stage order three system is 2^{30} or 1,073,741,824. This assumes that each of the 30 subsystems is either working correctly or catastrophically failed.

---

[3] The assumption is made that cancelling errors do not occur and the voting circuitry is perfectly reliable.

The number of entries in the transitional matrix for this system would be $(2^{30})^2$. It is obvious at this point, that even if special techniques could be found to eliminate 95% of these entries from consideration, the matrix would still be too big to handle conveniently, even using a large, high speed computer to perform the computations.

C. The Minimal Cuts Techniques

A technique for determining the lower bound on the reliability of redundant systems has been developed by Esary and Proschan [11]. This technique depends on the existence of "coherent" systems and definable sets of "minimal cuts". These terms have been precisely defined by Esary and Proschan in the following manner: A system is "coherent" when it fulfills the following conditions:

(1) A system which has failed because of a pattern of component failures existing within the system would not begin working again upon the occurrence of any additional failures.

(2) A system which is working in the presence of a set of component failures should not stop working if any of the failed components is repaired or replaced.

(3) A system should work when all of its components are working.

(4) A system should fail when all of its components are failed.

A "cut" is a set of components whose simultaneous failures are sufficient to cause system failure regardless of the operational state of the other system components. (A system will usually contain a relatively large number of cuts with many components appearing in more than one cut.) A "minimal cut" is defined as any cut in which there exists no subset of components whose combined failures would cause system failure.

Failure responsive systems meet all of the conditions required of coherent systems. They do not, however, always meet the condition of definable minimal cuts. The sensitivity of many of the response strategies to the order in which failures occur destroys

the concept of a minimal cut. Figures 4a and b shows the system which illustrates this point. In the example the response strategy allows only the subsystems on the top row to change location. Any of these may move forward[4] one or two stages if required by the existing failure pattern. If failures occur in the order indicated by the small circled numbers in figure 4a, the system will remain operational with the moveable subsystem from stage C having shifted to stage D. If however the failures occur in the order shown in figure 4b, the system fails because an unresolvable ambiguity exists in stage C. It is apparent from this example, that cuts can not always be identified by the pattern of failures existing at any particular time. This difficulty, combined with the complex problem of enumerating all the minimal cuts which can be identified, virtually prohibits the use of this analytical technique for estimating the reliability of failure responsive systems.

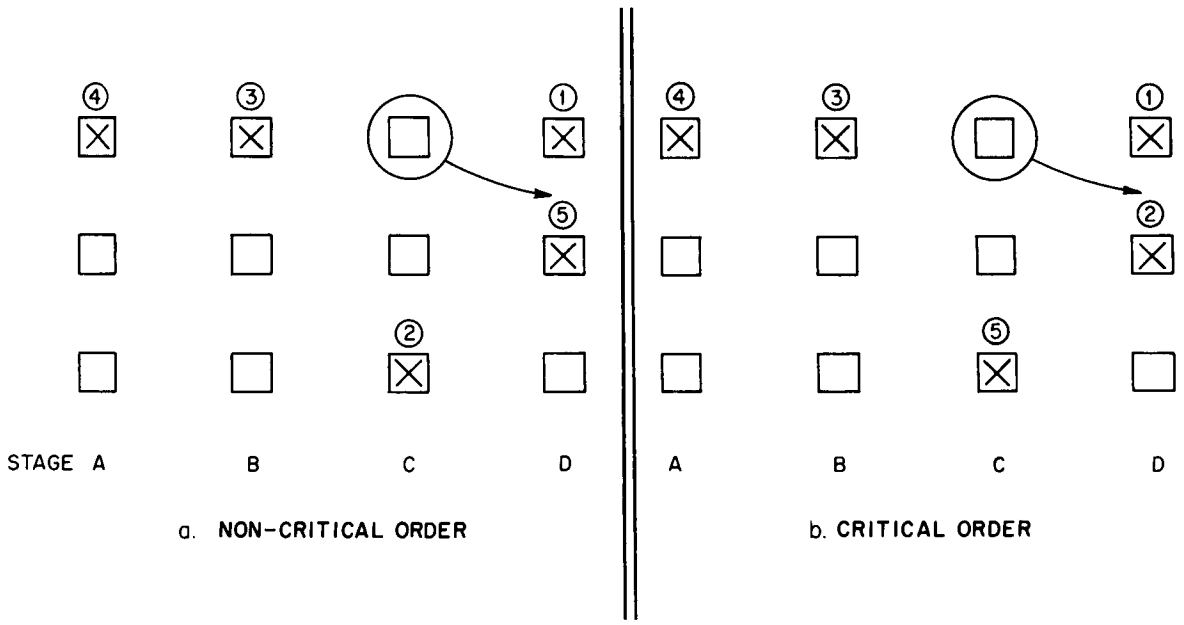


Figure 4. Critical and Non-Critical Order of Failures

[4] Stages A and D are assumed to be adjacent so that the moveable subsystem in stage D, for example, can be moved to stages A or B.

D. The Computer Simulation Method

The concept of physically modeling a large system and testing the response of the model to gain knowledge of the true situation is a form of simulation that has been used for centuries. Although in computer simulation no physical model is built, a functional representation of a system to be tested is formed by a sequence of program statements. These statements are used to specify all of the individual deterministic actions of the system. Inputs and outputs to this model are presented to the computer in the form of data rather than physical quantities. The response of the true system to various perturbations in the input data is estimated by observing the response of the computer representation just as if a physical model had been built.

Mathematicians almost always object to the use of either physical or computer simulation because no rigorous proofs of the results can be given, and the system response cannot be described by a group of neat, closed-form expressions. Because these are valid objections, simulation analysis is usually used only for treating very large complex systems where the number of variables in the problem prohibits the use of more standard mathematical modeling techniques, or where the cost of exercising the real system is too high. In the case of failure responsive systems, the variety of characteristics inherent in the response strategies are difficult to model accurately in a mathematical expression. However, such systems can be easily handled by a computer simulation program.

The inputs to this particular program are in the form of response strategy constants, subsystem failure rates and random numbers. The random numbers are correlated with individual subsystems to represent random failures. After the simulation of several hundred input failure patterns, the program output is used to estimate system reliability.

VI. THE COMPUTER SIMULATION PROGRAM

A. The Operational Principles of the Program

The topography of a system is modeled in the computer simulation program by an array of stored data. These data can be roughly divided into two sets. The first set contains information which specifies the operating state or the characteristics of individual subsystems. The second set contains information which determines the characteristics of the overall system operation. Different system response strategies and other operational requirements are simulated by establishing, within the computer memory, the appropriate initial values of each of the stored data words. In some cases these values are read directly into the computer from an external source, while in other cases, the data is generated by the computer operating under the command of special input control constraints.

B. Individual Subsystem Information

1. Failure Location Intervals

The operation of the program is based on the assumption that subsystem failures can be simulated by the computer in such a manner that they represent the way in which actual failures would occur in operating systems. The main problem is to determine which subsystem should be designated as failed when a subsystem failure is assumed to have occurred at a particular time. In order to accurately represent the occurrence of a failure in an operating system, the conditional probability of a subsystem's just having failed, given that exactly one subsystem failure has just occurred, must be equal to the

same conditional probability that would apply to the subsystems in a comparable operating system. It is shown in the Appendix that this conditional probability is given by the simple expression:

$$P\ (i/1) = \frac{\lambda_i}{\sum_{i=1}^{L} \lambda_i} \qquad (3)$$

where i refers to the i^{th} subsystem; λ_i is the failure rate of the i^{th} subsystem and L is the total number of subsystems which were operational before the occurrence of the present failure.

Randomly located subsystem failures are generated by the simulation program, subject to the above conditional probability, in the following manner. The conditional probability of failure associated with each subsystem is computed according to equation (3). The interval of numbers between zero (0) and one (1) is then divided into L subintervals with the length of each subinterval being directly proportional to conditional probability of failure of one subsystem. The assignment of one subinterval to each subsystem results in the unique association of every number in the zero (0) to one (1) interval with exactly one subsystem. To locate a simulated failure, the computer draws a random number from a population which is uniformly distributed over this same zero (0) to one (1) range. The random number thus selected must fall into one of the subintervals associated with one of the subsystems. The computer locates this subsystem and designates it as failed.

In performing this operation, the computer first reads in the failure rates of the subsystems. It then uses the failure rates to determine the conditional probabilities of failure to be associated with the subsystems, and corresponding intervals of numbers. The upper and lower bounds on the intervals then become a part of the stored data.

2. Space Lists

One of the major differences between the response strategies is the sequence in which subsystems are called to aid the failed or, in some cases, vulnerable stages. One of the outstanding features of this computer program is the simple manner in which an almost unlimited variety of sequences can be set up.

As part of the initialization procedure, an identification number is assigned to each subsystem. The sequence of subsystems to be called to aid any particular stage is established by simply reading into the computer memory a list of identification numbers. The order of the numbers combines with their actual value to precisely specify the desired sequence.[5] The list of identification numbers is referred to as a "spare list". (This technique also permits the testing of random response strategies by the insertion of random number spare lists.)

3. Other Stored Data

In addition to the information concerning random number interval bounds and subsystem spare lists, a variety of other information is stored in the computer memory. This information is used to specify the general characteristics of the response strategy being tested and to control many of the peripheral program operations. Figures 5a and 5b illustrate a typical example of the general strategy characteristics which are specified in this manner. In both cases, stage three has experienced one failure and stage four has experienced two failures. At this point, stage four requires aid. In both cases, the first

---

5 It should be noted that the program is equipped with a pattern duplicating option that permits a sample spare list to be read in for one stage and the pattern reproduced for all other stages with all "spare" subsystems coming from the same relative location.

choice of a replacement is subsystem A; the second choice is B. The stored information specifying the operation of the system in figure 5a permits A to shift to stage four, leaving stage three in a non-redundant state. In contrast, the operation of the system in figure 5b restrains the movement of A because of the previous failure in the same column and forces B to aid stage four.

An example of the peripheral program operations controlled by the remaining variables is the output format. The information which is printed out by the simulation program can be manipulated so that details of the individual simulated failure patterns are available for inspection. Conversely, the output may be restricted to a brief summary of the combined statistical results of many runs.

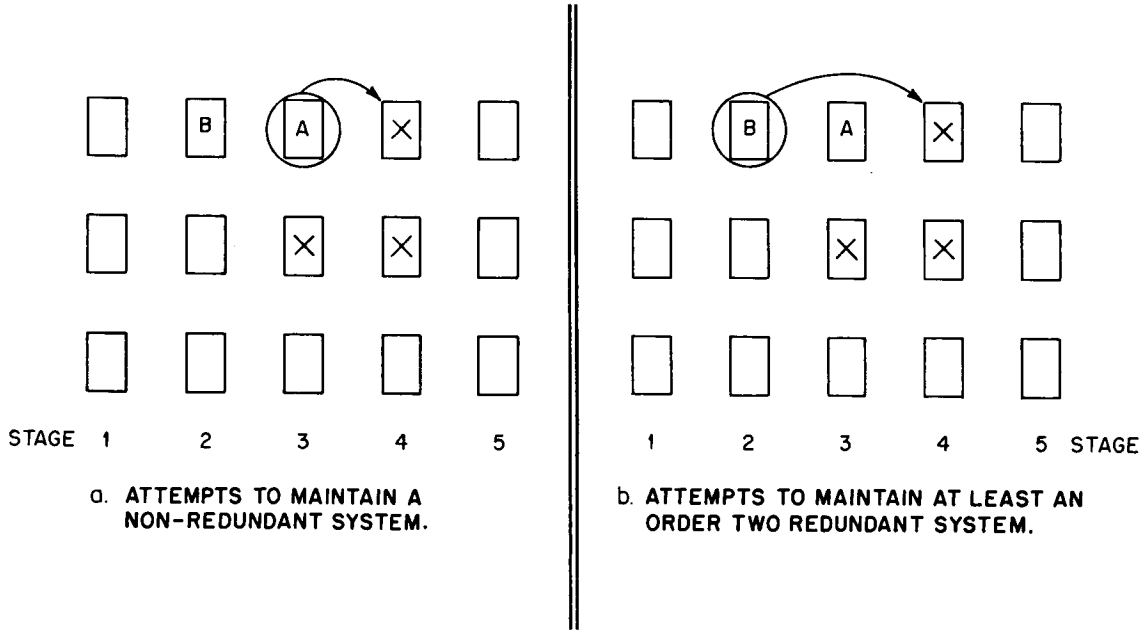


Figure 5. Two Response Strategies

C. The Detailed Operation of the Program

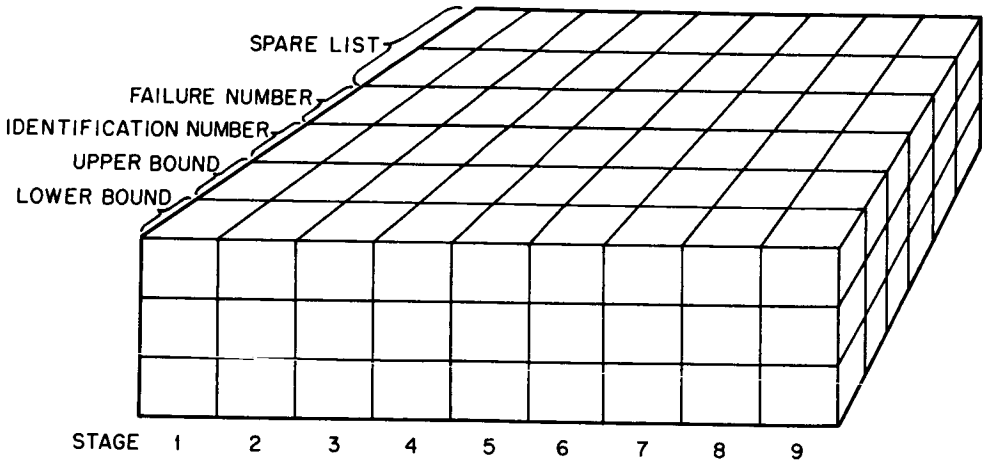
1. Data Storage

The portion of the data which concerns individual subsystem operation is organized into the format of a three dimensional matrix. This matrix closely resembles the actual form of the system being simulated because two of the dimensions correspond to the number of stages and the order of redundancy of the base system. The third dimension contains data words about the subsystems represented by the first two dimensions. Figure 6a shows one such matrix which represents the typical system shown in figure 6b. As shown in figure 6a, the first two words at each location specify the random number interval bounds associated with that subsystem location. The third word specifies the identification number of that location. The fourth word is non-zero only if the simulated subsystem initially found at that location has moved or failed. If this word is non-zero, it equals the number of moves or failures which have occurred in that column at the time the particular subsystem moved or failed. The remaining data words in each matrix location are members of the spare list, where the fifth word represents the first entry on the list, the sixth word the second entry, and so forth.

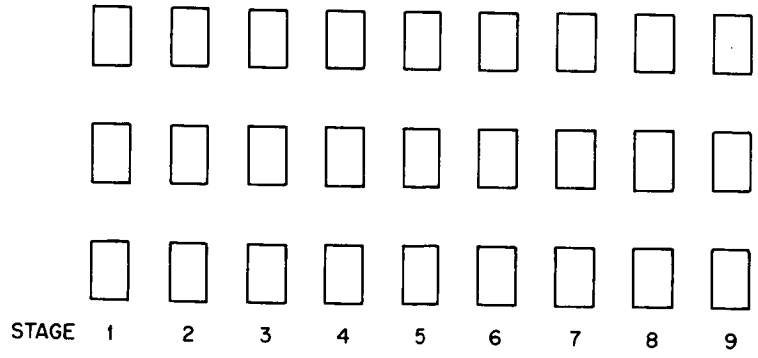
The data which is stored outside this matrix applies to the overall system or program operation. This data is simply stored as individual variable values and does not form any sort of integrated data block.

2. The Simulation Procedure

After all the initial data concerning the system operation has been inserted into the computer memory, the actual simulation phase of the program begins. Although this part



a. MATRIX REPRESENTATION



b. A TYPICAL SYSTEM

Figure 6. A Typical System and Its Matrix Representation

of the program is complicated in terms of computer instructions, it is relatively simple in principle. A series of random numbers is chosen from a population uniformly distributed between zero (0) and one (1). As each number is chosen, it is associated with one of the simulated subsystems by locating the subsystem whose random number interval contains the chosen number. The failure of the subsystem is noted by adding one (1) to the previous number of failures observed in the stage to which this subsystem belongs and storing the new number in the fourth position in the matrix at that subsystem location. In addition, the random number interval bounds are set to zero (0), thus prohibiting multiple failures of any one subsystem.

After the subsystem failure has been recognized, the computer checks to see if the stage which experienced the failure subsequently requires the aid of a replacement subsystem. If the stage still meets all of the requirements imposed by all of the related criteria, no further action is taken, and the next in the series of random numbers is selected. If the stage requires aid, the program begins searching through the subsystems whose identification numbers appear on the spare list of the previously failed or moved block in the vulnerable stage.[6]

The search is conducted by interrogating the possible spares in the order in which their identification numbers appear on this spare list and determining their availability. This continues until the "repair" is made or it is determined that the repair cannot be made. If the repair can be made, the data describing the subsystem to be moved is shifted from its initial location to the location of the previous failure in the vulnerable stage. Depending on the strategy being tested, the subsystem in its new location may lose all of its remaining repair capability; it may retain its old capability, or it may assume the

---

6 The only case in which aid may be required by a stage which has not previously experienced a failure or the loss of a subsystem to another stage is in systems having unequal stage redundancy. The program then considers the low order stages as having lost some subsystems.

capability of the subsystem which it replaced. If the repair cannot be made, a check is made to see if the number of operating subsystems remaining in the vulnerable stage is two or greater. If the answer is yes, the simulation continues with the drawing of another random number. If the answer is no, it is assumed that the most recent failure has resulted in the occurrence of an unresolvable ambiguity in the vulnerable stage; therefore, the system has failed.

The procedure is continued until the system reaches the failed state. At this point, the total number of subsystem failures in the system is recorded, the matrix is reset to the original state, and the entire procedure begins again. The repetition of this procedure several hundred times produces statistical information which can be used to construct estimates of the reliability versus time curves of systems using the response strategy being tested. The entire simulation procedure is summarized by the flow chart in figure 7.

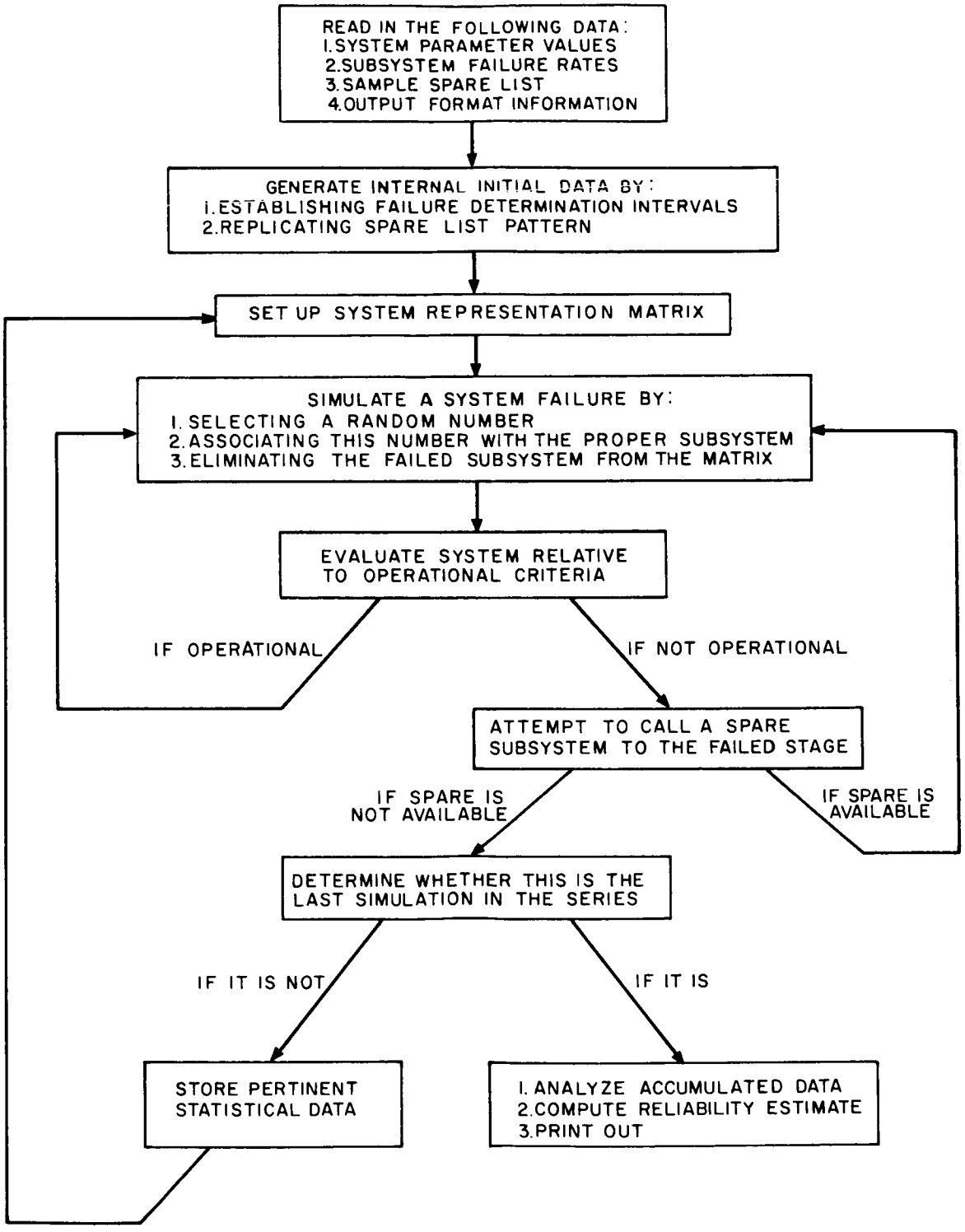


Figure 7. Summary Flow Chart of Computer Program

VII. SYSTEM EVALUATION

A. Methods For Estimating System Reliability Versus Time Curve

1. The Conditional Probability Method

The information obtained from the simulation procedure can be used to construct a histogram which describes the relative observed frequency of system failures for any given number of subsystem failures. Figure 8 shows an example of such a histogram. The height of the lines f(x) in this histogram are determined by counting the number of systems which were observed to fail with exactly x subsystem failures in the system and dividing this number by the total number of system failures which were simulated. Thus, the magnitude of these lines represent a statistical estimate of the probability that a particular system will fail at the occurrence of exactly the xth subsystem failure.

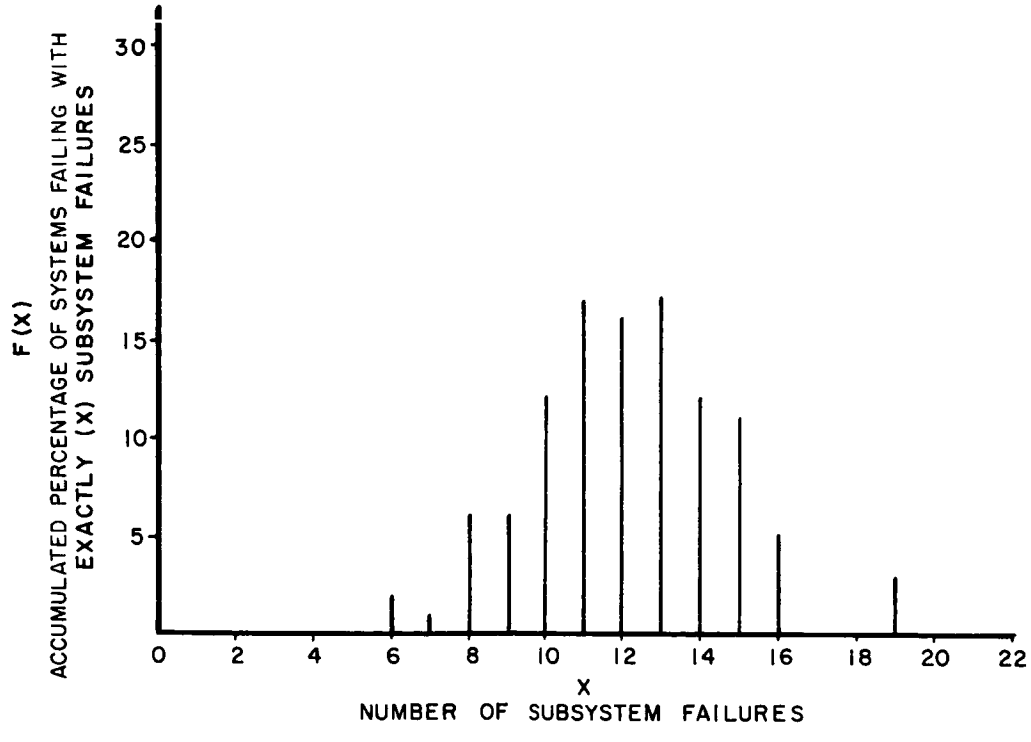


Figure 8. Histogram of Observed System Failures

Figure 9 shows the cumulative curve which is formed by adding segments of the above histogram according to the relationship

$$F(x) = \sum_{i=0}^{x} f(x) \tag{4}$$

The magnitude of F(x) is an estimate of the conditional probability that a system has failed, given that exactly x failures exist within the system. It is this probability that is needed to calculate the system reliability.

If the assumption is made that the failure rates of all the subsystems are equal, the probability of exactly x failures occurring in a system containing N subsystems can be calculated from the expression

$$P(x,t) = \binom{N}{X} (1-e^{-\lambda t})^{x} (e^{-\lambda t})^{N-x}. \tag{5}$$

where,

$\binom{N}{X}$ is the symbol for x combinations of N items.

This probability can be combined with the estimated conditional probability of system failure to produce an estimate of the overall system reliability. This can be done using the relationship

$$R(t) = \sum_{X=0}^{N} F(x) \quad P(x,t). \tag{6}$$

To apply this technique to non-homogeneous systems having more than one subsystem failure rate, two alternative possibilities have been considered. By recording the distribution of failures among the different types of subsystems, the individual lines

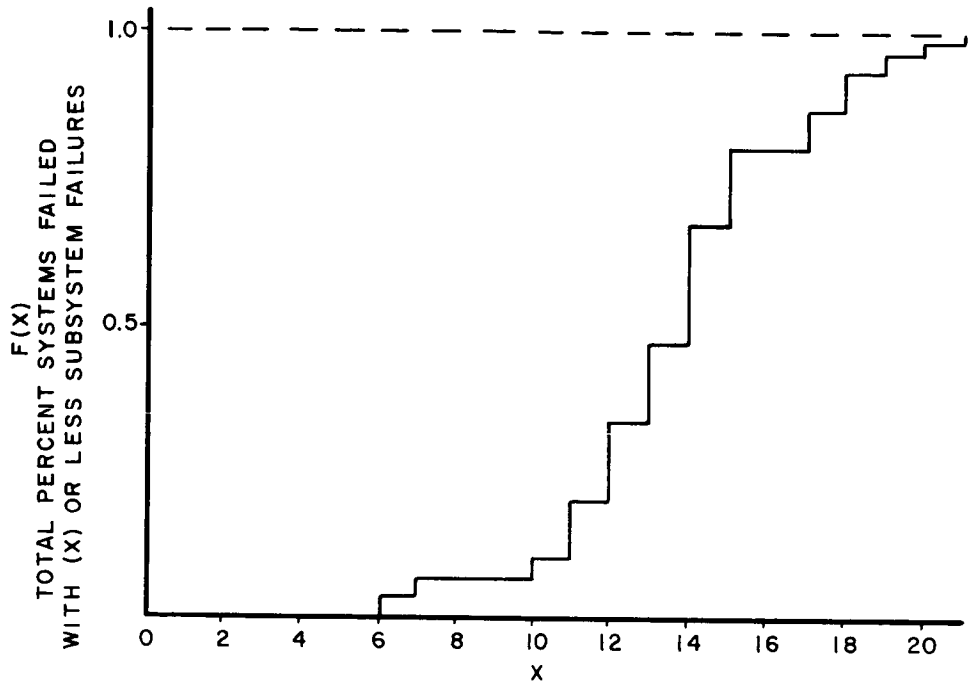


Figure 9. Cumulative History of Observed System Failures

shown on the histogram could be subdivided so that their magnitudes represented the conditional probability that system had failed, given that the system has absorbed x failures of one type subsystem, y failures of another types subsystem, z failures of another and so forth. To obtain meaningful estimates of each of the conditional probabilities which can be defined in this manner, an unreasonably large total number of system failures would have to be simulated.

A much simpler method, which is equally accurate for a limited number of samples[7], has been used in this program. In this second method a weighted average[8] of the various subsystem failure rates is computed, and this number is substituted for the single failure

---

7 i.e., 500 to 1000

8 $\sum_{i=1}^{N} \frac{m_i \lambda_i}{N}$ where m_i is the number of subsystem subject to the failure rate λ_i.

rate used in the equation given above for computing the reliability of homogeneous systems. It has been found experimentally that the random error introduced by the generation of random failures usually masks out completely any error introduced by the use of the weighted average.

2. The Random Time Generation Method

In addition to the simulation of random failure patterns, the computer program can be used to locate randomly in time the occurrence of each failure in a pattern. It has been previously stated that each subsystem is subject to a constant failure rate. This implies that the probability of continuous operation of all (N) subsystems in any system from the time t = 0 is given by the expression

$$R(t) = e^{-\sum_{i=1}^{N} \lambda_i t} \tag{7}$$

Conversely, the probability that the first subsystem failure will occur in the interval of time zero to t is given by the expression

$$P(1^{st}) = 1 - R(t) = 1 - e^{-\sum_{i=1}^{N} \lambda_i t} \tag{8}$$

Using a relationship described by A. M. Mood,[12] a set of random numbers drawn from a population uniformly distributed between zero and one can be transformed to a similar set of random numbers belonging to any other distribution. For the case of the exponential distribution of interest, this is accomplished by letting

$$f(y) = 1 \qquad \text{for } 0 \leq y \leq 1 \tag{9}$$

$$f(y) = 0 \qquad \text{elsewhere} \tag{10}$$

and $y = G(t) = 1 - e^{-\lambda_s t}$ where $\lambda_s = \sum_{i=1}^{N} \lambda_i$ (11)

Figure 10 shows this last relationship graphically.

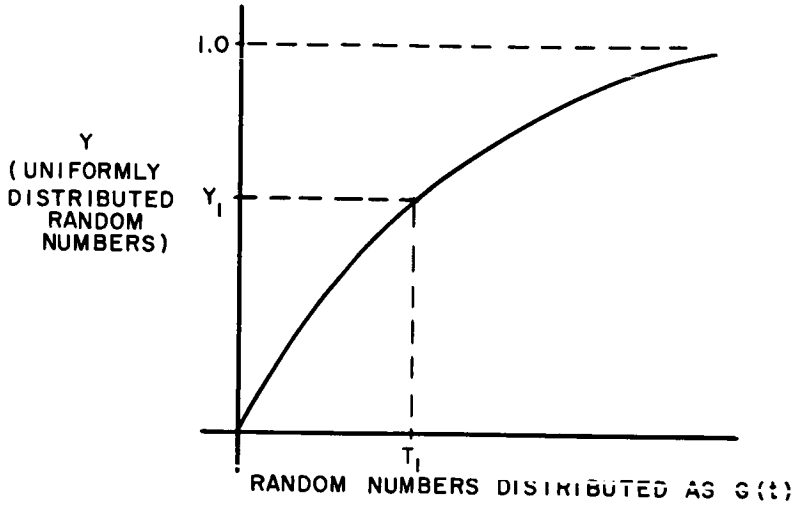


Figure 10. Uniform to G(y) Distribution Transformation

Using this relationship, a random number taken from a set of uniformly distributed numbers is used to generate the time to the first subsystem failure with the correct probability of picking a time from any increment along the time axis. By simply subtracting the failure rate of the first failed subsystem from the total failure rate λ_s and setting the time scale reference at the point of the first failure, a time between the first and second failure can be determined in the same fashion. The sum of these two times simulates the total system operating time up to that point.

This process is repeated until the system withstands so many failures that it fails to meet the system operational criteria. The occurrence of this event stops the procedure, and the various system state change characteristics and the total operating time are recorded.

The record of total operating times can be directly used to estimate system reliability[9] versus time. This is done by ordering the individual operating times so that the percentage of systems operating prior to any given time can be calculated. This percentage is exactly the observed system reliability and may be used as an estimate of the true system reliability. It should be noted that the observed system reliability is always constant between observed system failure times, therefore, a discontinuous curve such as that labeled "A" in figure 11 results from the unmodified use of this estimation procedure. A much smoother curve can be obtained by interpolating intermediate values in the area between the points.

---

9 Reliability, is defined here as the probability of continuous system operation over a time interval zero (0) to (t) when it is known that the system was operating at time zero (0).

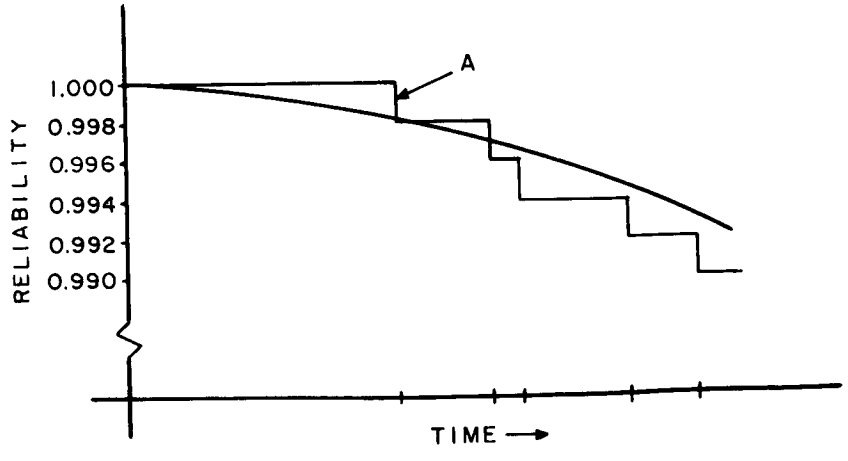


Figure 11. Comparison of Reliability Estimation Curves

3. Comparison of the Two Estimation Techniques

The reliability curves produced by both of these estimation techniques tend to be more accurate in the central region of the curves where most system failures occur than they are at either of the upper or lower extremeties. This situation exists because the extreme regions are dominated by the few system failures which occur either with very few subsystem failures or unusually many subsystem failures having been withstood. No rigorous method for evaluating the two estimation techniques has been devised. For the purpose of this study, the equation method of estimation has been chosen rather than the time generation method. This choice was based on fact that the equation method required no sophisticated method of interpolating between observed points to provide meaningful estimates of the shape of the reliability curve in the high reliability region. The curves shown in figure 11 may help clarify this point. In contrast to the five events which control the shape of the step curve which naturally results from a sample of 500 events using the time generation technique, all the 500 events contribute in some amount to the continuous curve produced by the equation technique.

B. Single-Valued Measures of Performance

The techniques which have been described above provide an estimate of system reliability as a function of time. Because the comparison of the reliability of various systems at every point in time is not practical or particularly meaningful, a method of using the functional reliability estimate to generate a single-valued measure of performance had to be found. The several possibilities which have been considered are described below.

1. Mean Time Before Failures

The most popular reliability measure applied to non-redundant systems is the "mean time before failures" or "MTBF". The MTBF is a quite useful reliability measure for non-redundant systems of this type because the associated reliability curves are all of the exponential form, having time constants which are inversely proportional to the MTBF (see figure 12). This measure is not as meaningful for failure responsive systems whose reliability curves vary in form. Figure 13 shows two curves which have approximately the same MTBF's, but they are obviously not equivalent systems. It would seem, therefore, that a more useful measure should be found.

2. System Reliability at a Selected Time

The reliability of systems at one point in time is an alternate measure that deserves consideration. This is by far the easiest measure to compute, but it does have some inherent disadvantages. This measure may simply show that one system is more reliable than another system at one particular point in time. If a situation such as the one illustrated in figure 13 exists, the system which is more reliable at t_1 may not be the more desirable if the mission is completed at t_2. Similarly, two systems may appear to be nearly equivalent at the evaluation time when they differ greatly before the end of the mission time. Figure 14 shows examples of the reliability curves of two such systems. Again, it would seem that a still better measure should be found.

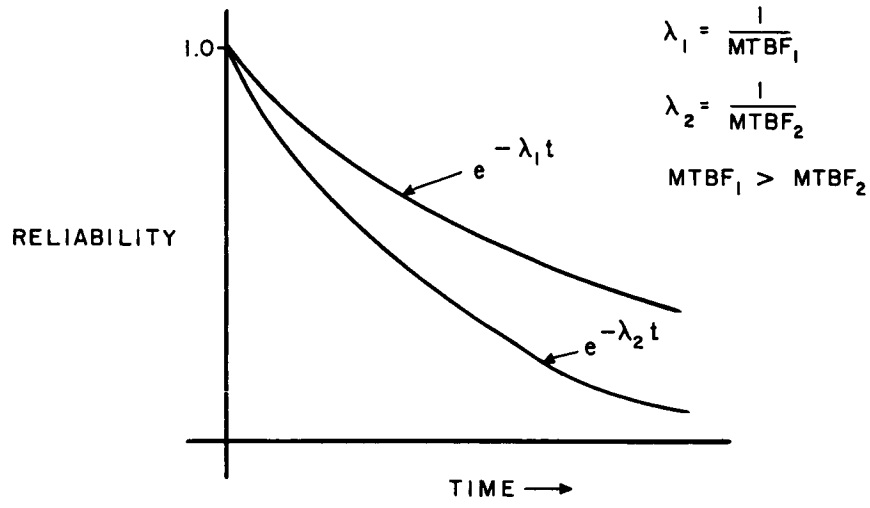


Figure 12. Non-Redundant System Reliability Curves

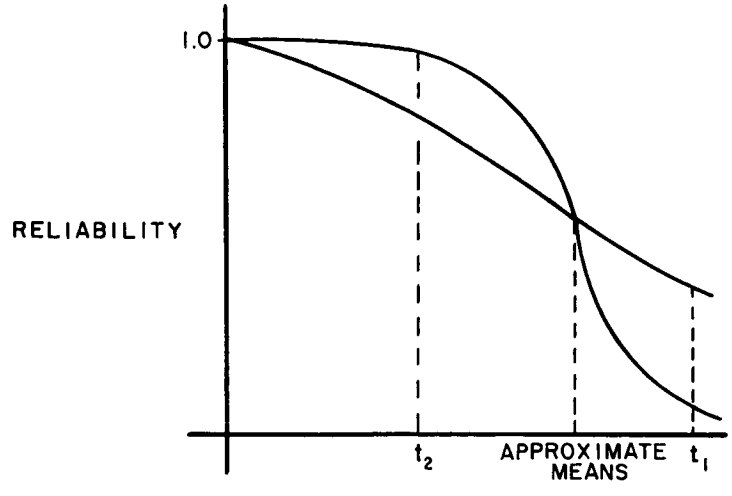


Figure 13. Different Reliability Curves with Similar Means

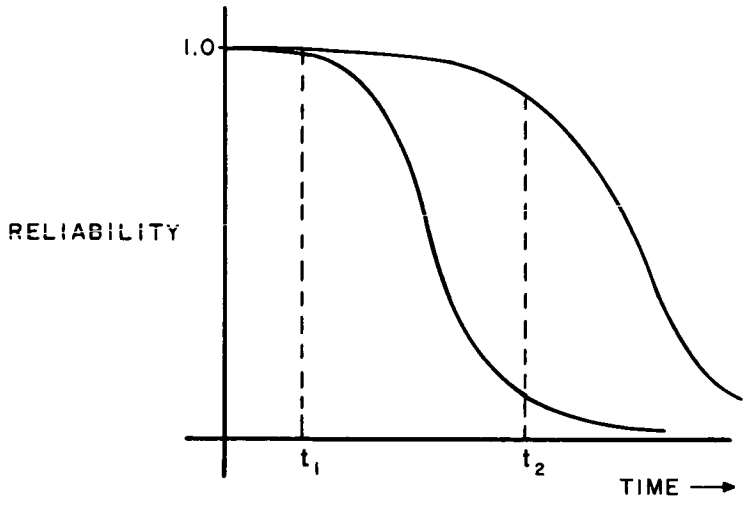


Figure 14. Different System Reliability Curves with Similar Short Life Reliabilities

3. Quantile Occurrence

The third available evaluator has not been extensively used in the past, but it seems to overcome some of the disadvantages of the first two possibilities. This method uses the time at which the system reliability falls below a pre-determined quantile as the measure of evaluation. This measure is defined here as the "useful life". Figure 15 illustrates the method for the 0.90 quantile. In this case, the system characterized by the 0.90 quantile occurring at t_2 is more desirable than the system with the quantile occurring at t_1. This evaluator tends to overcome the problem inherent in the MTBF evaluator because only the region of the curve which is of interest enters into the evaluation. The problem of performing the evaluation only at a single point in time, which is associated with the second evaluator is also solved because this third evaluator is more sensitive to differences in system reliabilities in the high reliability region.

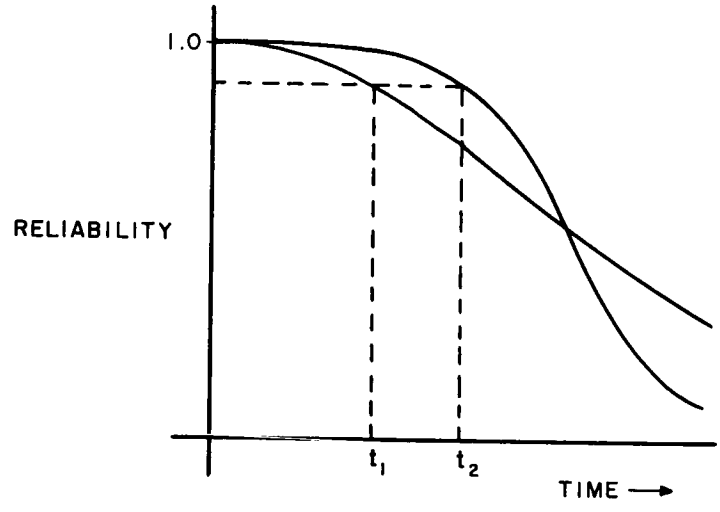


Figure 15. The "Useful Life" Measure

The problem of accepting the wrong system because of curve crossover may be virtually eliminated by confining the quantile selected as the criteria to the high reliability region. This is not a particularly significant restraint because the nature of the applications, which require the use of the sophisticated systems being considered here, will require operation strictly in the high reliability region.

VIII. SIMULATION RESULTS

The simulation study of response strategies has been conducted in two phases. In the first phase the assumption was made that all of the peripheral error detection and switching circuitry required to implement the systems was perfectly reliable. In the second phase, this assumption was dropped and a failure rate was associated with the peripheral circuitry as it is with the functional subsystems.

Although the first phase effort may appear to be completely superfluous when compared to the second phase, this is not the case in practice. The first phase results indicate which response strategies are optimal if it is given that certain numbers of subsystems appear on the individual spare lists. This optimal strategy information is independent of the failure rate of the switching circuitry. The second phase results merely show what the length of the spare list should be, given the failure rate of the subsystems, the minimum peripheral circuitry failure rate, and the additional failure rate which must be added to the minimum to account for each addition to a spare list.

In the pages which follow, the results which have been obtained during both phases of the study are described. To obtain each point estimate of the reliability of systems using any of the response strategies, the simulated systems have been subjected to five hundred sets of failure patterns of sequentially generated subsystem failures. The patterns contain the minimum number of subsystem failures required to cause system failure when the subsystem failures occur in the order generated.

The curves shown below were constructed by plotting the time of occurrence of the 0.90 quantile on the estimated reliability curves. All the curves represent systems of twenty stages, with the subsystem failure rate constant for all stages in all systems. The original order of redundancy of the systems tested is noted in the subsection title.

A. Phase I Simulations

1. Order-Three Systems

a. Experiment I. In the first set of response strategies to be tested, the capability of a subsystem to serve as a spare (or a replacement) was restricted to one subsystem in each stage. The difference between the strategies stemmed from the pattern and the order in which the subsystems having the spare capability appeared on the spare lists of the individual stages. Three subsets of strategies were tested in the course of this experiment. Figure 16 shows a sample spare list for one stage of each subset. The spare list pattern is replicated for each stage, with the first and last stages assumed to be adjacent, thus forming a closed "loop". The members of each subset all employ the same spare list pattern. The individual members of a subset may be distinguished from each other by the number of subsystems composing their associated spare lists.

The object of this experiment was two fold. The first objective was to attempt to verify the null hypothesis that the individual strategies were pair-wise equivalent, i. e. that only the length of the spare lists was significant, and not the selection pattern. The second objective was to determine the effect of allowing systems to have a "rescan" capability. A system with rescan capability is one which first scans a spare list attempting to find and call up a replacement subsystem only from a stage which has experienced no failures. If no replacements are found, it will "rescan" the list, searching for a subsystem from any stage which has more than one operating subsystem.

Figures 17 and 18 shows the results of this experiment graphically by the curves. It is apparent from these curves that the difference between spare list patterns (i. e. , sequential, uniformly distributed, or alternating consecutive spare lists) is insignificant, but that the rescan capability does have a significant effect.

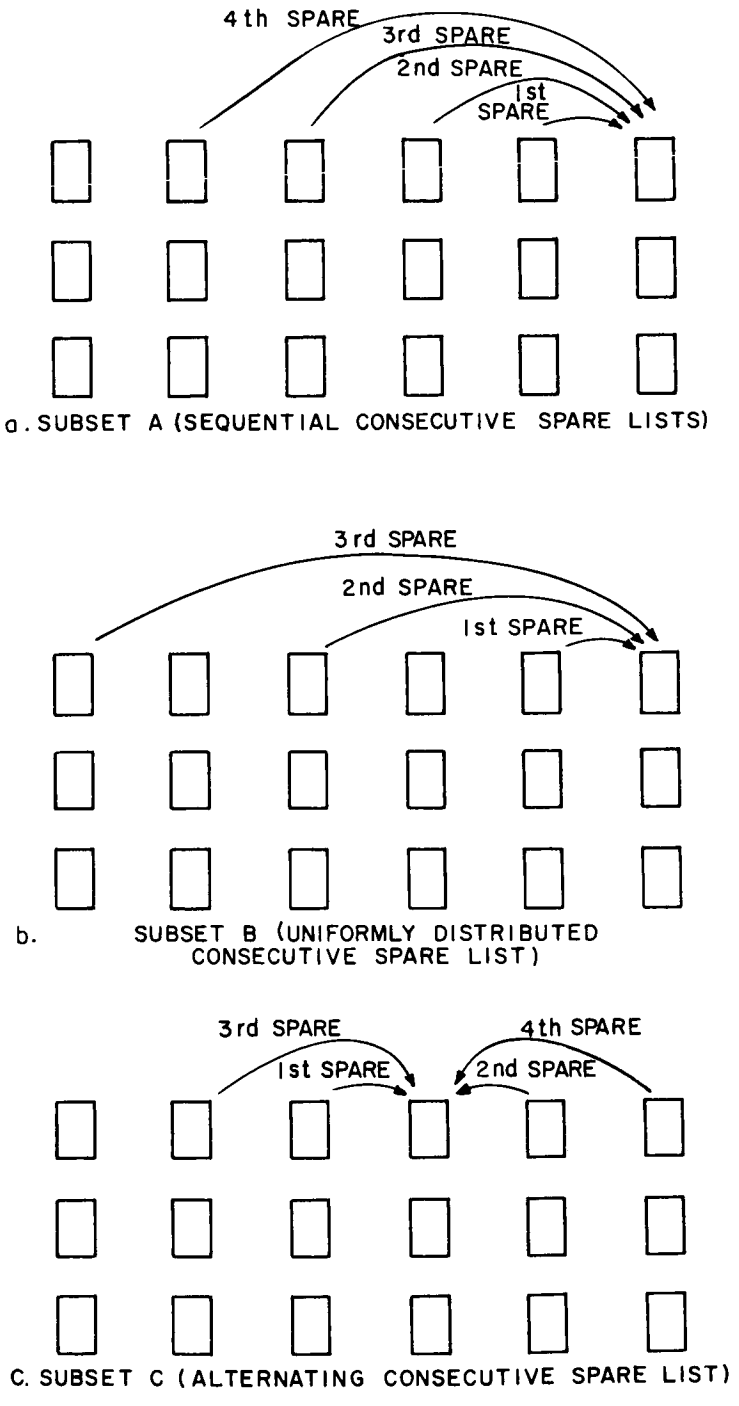


Figure 16. Sample Strategies for Consecutive Lists

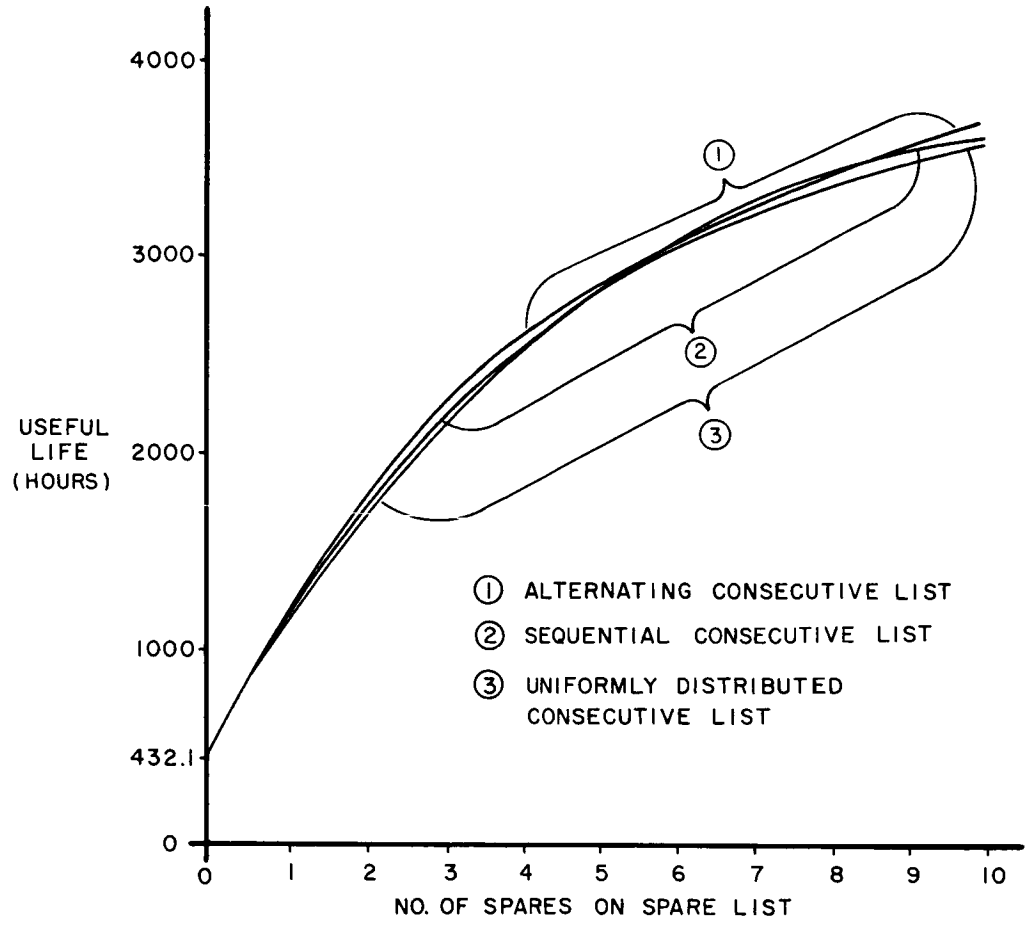


Figure 17. Comparison of Alternating and Sequential Consecutive Lists

b. Experiment II. In the second set of strategies to be tested, the single spare per stage restriction was released and any subsystem in a stage was allowed to perform as a spare if the spare list lengths required. Figure 19 shows the "normal step" pattern which was the basic pattern used for all the strategies in this class. The only difference in the strategies was the length of the spare lists. The object of this experiment was to determine the effect produced by spreading the spare capability among more subsystems with less movement capability.

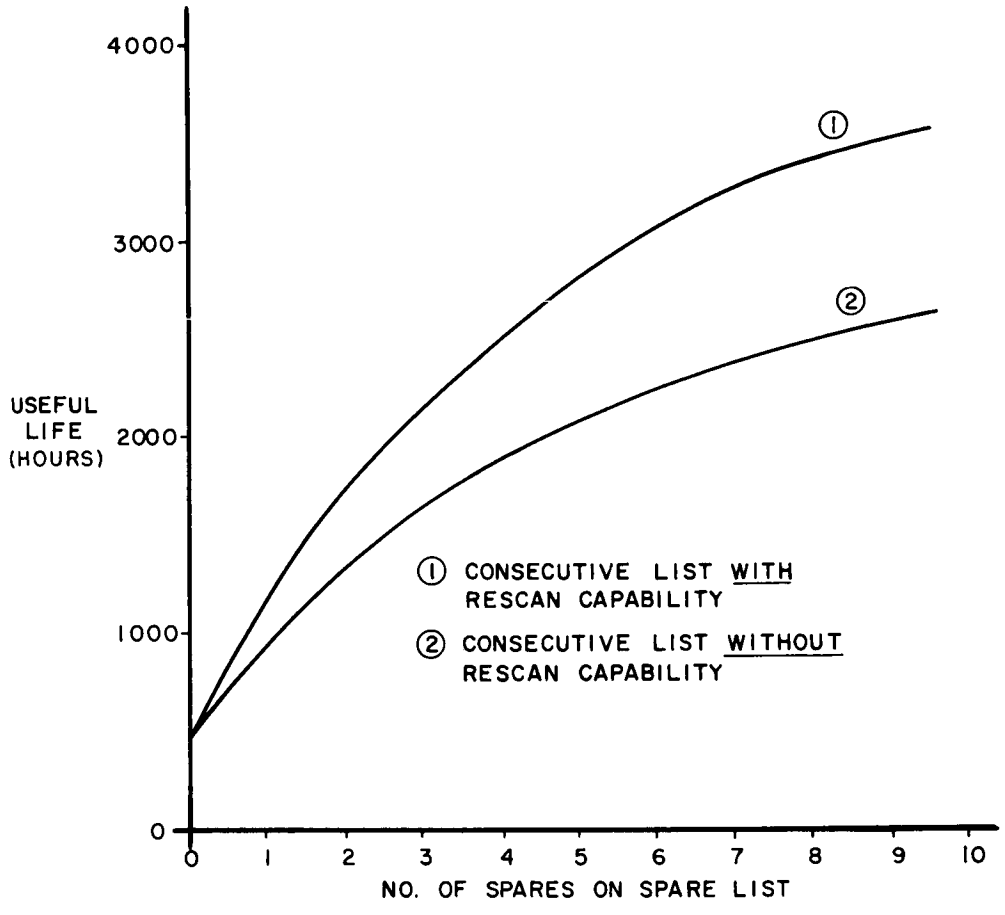


Figure 18. Comparison of Response Strategies with and without Rescan Capability

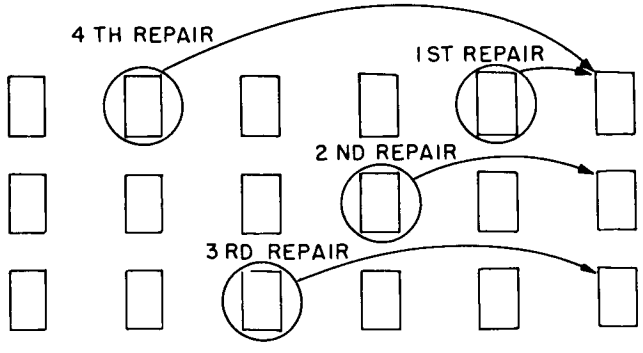


Figure 19. Sample Strategy for a Normal Step List

Curve 1 in figure 20 shows the results of this experiment relative to curve 2, the curve for the consecutive lists from Experiment I. It can be seen from these curves that the use of the step list results in a pronounced improvement over the consecutive list system.

c. Experiment III. The next set of response strategies to be tested can be described as modifications of the step list strategies tested in Experiment II. Figure 21 shows an example spare list pattern used by these strategies. The close resemblance to the step list pattern is immediately apparent. The primary difference between the two sets of strategies is the distribution of the stages from which the spares are drawn. The strategies tested in this experiment tend to reduce the mutual dependence of any two stages on replacement

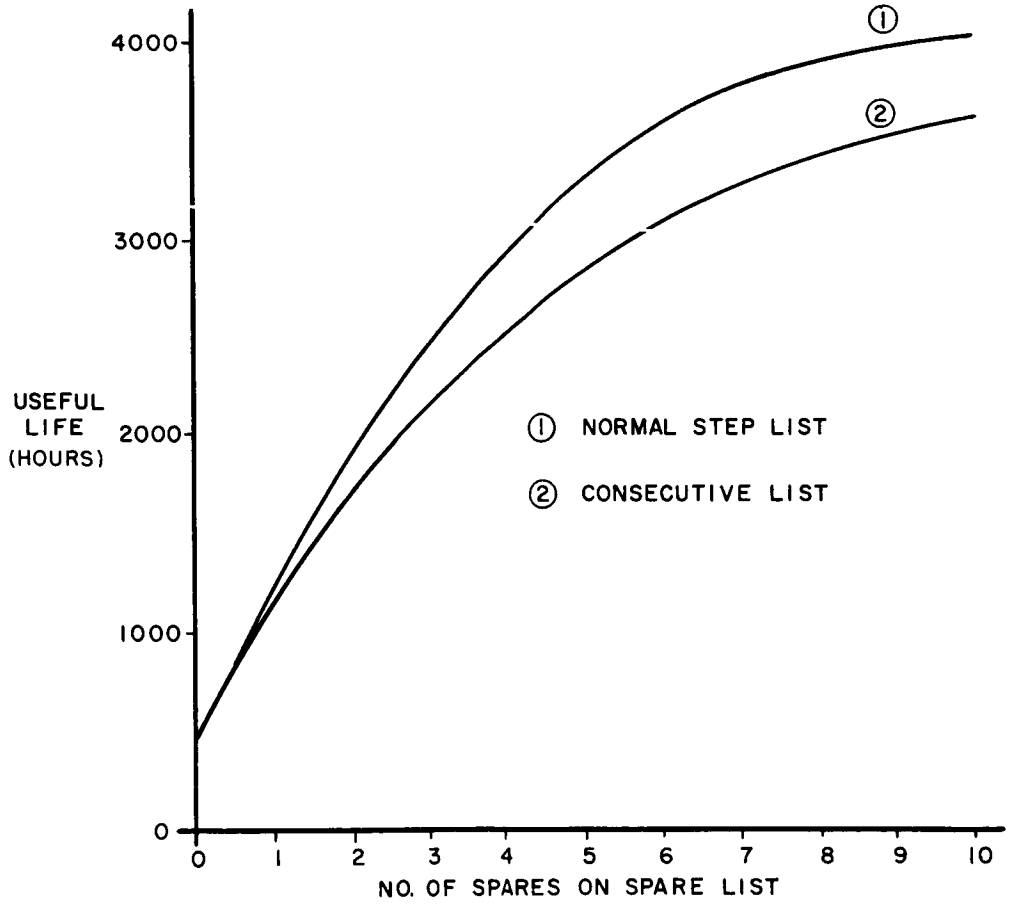


Figure 20. Comparison of Normal Step and Consecutive Lists

subsystems from the same stages. The object of this experiment was to test the effect of this reduction in the mutual dependence.

Figure 22 shows the curves which indicate the effect achieved by progressively distributing the spares. The relatively minor gains which are made by this simple modification are significant, however, because they can be achieved without increasing the amount of peripheral circuitry, regardless of the type circuitry which is used.

d. Experiment IV. All of the strategies considered in the first three experiments have restricted spare subsystems to making only one of its possible moves. Thus, if a subsystem moved to a new location and made a repair, every spare list on which that subsystem originally appeared was effectively shortened by one entry. The set of strategies which were tested in this experiment employed spare lists which were identical to those of the consecutive and distributed step list used previously. The only difference was that subsystems were allowed to move to the aid of vulnerable stages without regard to whether they had moved previously. The object of this experiment was to determine if this "multiple-move" capability would be significant in improving system reliability.

Figure 23 shows the results of the simulation graphically. Again, one of the consecutive list curves developed in the earlier experiments is included in figure 23 to provide a reference for the degree of improvement. It can be seen from this figure that a slight improvement is obtained through the addition of the multiple-move capability, but it is not nearly as pronounced as some of the other effects have been.

This same experiment was conducted using the progressively distributed step list. In this case, the curves were precisely the same for systems having less than four spares on spare lists of the individual stages. For systems having four or more spares, the curves were so nearly the same that the difference could not be observed from plots made to the same scale as the rest of the curves presented in this paper. In retrospect, this

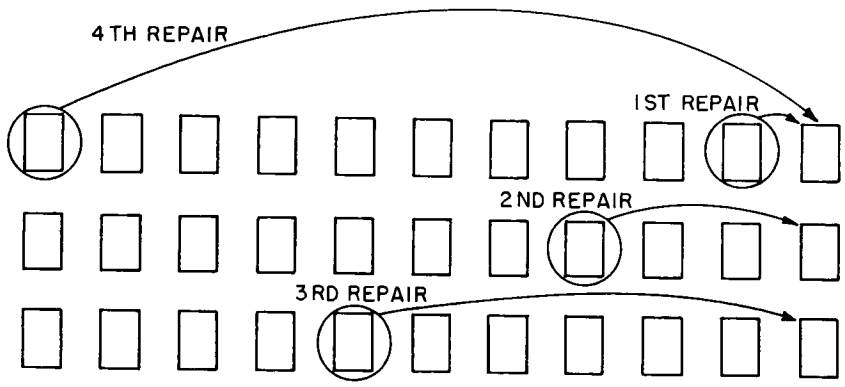


Figure 21. Sample Strategy for Progressively Distributed Step Lists

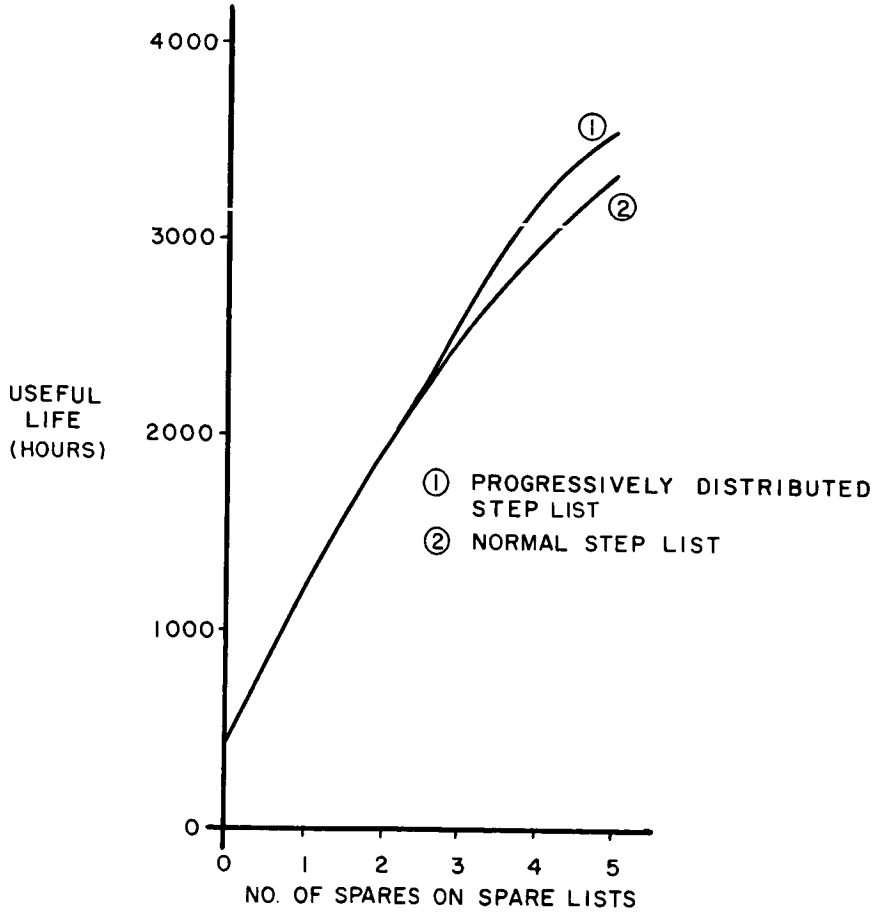


Figure 22. Comparison of Progressively Distributed and Normal Step Lists

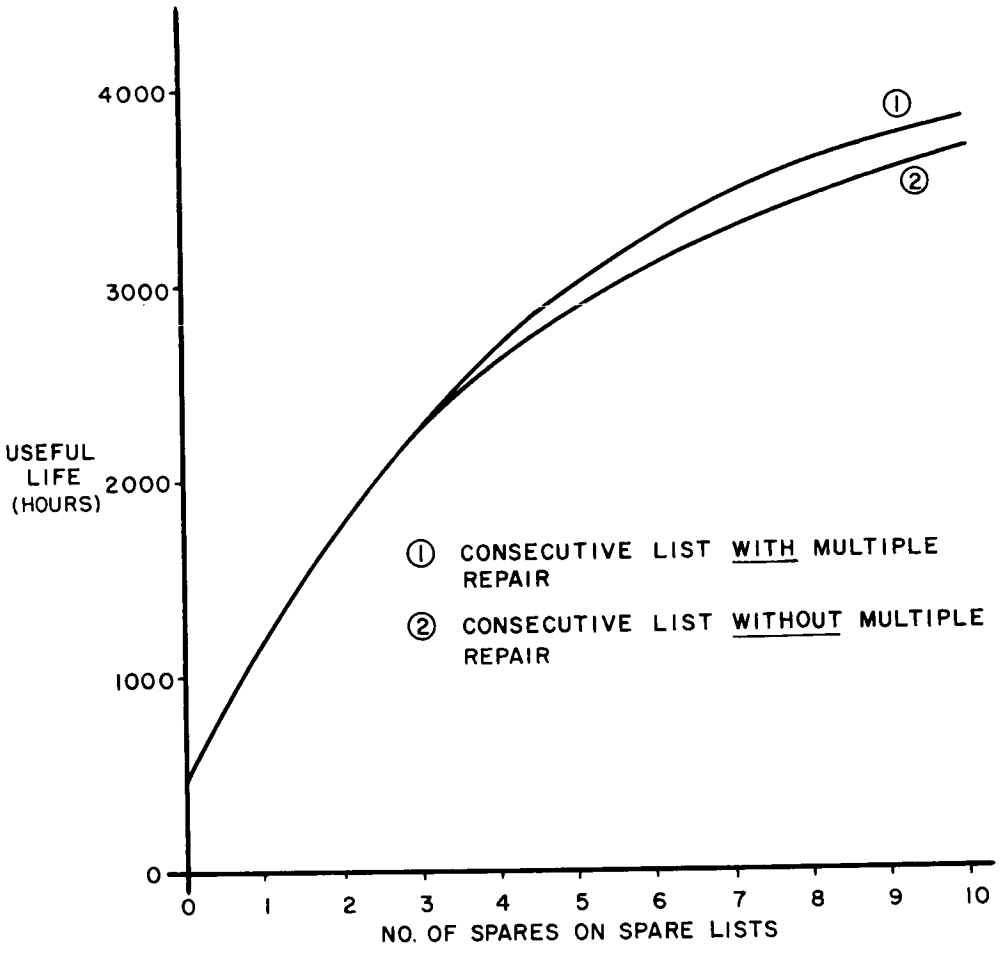


Figure 23. Comparison of Consecutive Lists With and Without Multiple Repairs Per Subsystem Capability

result could have been at least partially anticipated because the subsystems in systems having less than four spares per stage have only one movement possibility; therefore, multiple moves are inherently impossible. For systems having four or more spares per stage, the chance of requiring multiple moves is apparently very low.

e. Experiment V. Each of the first four experiments was designed to test the effect of some particular characteristic of systems using well-ordered response strategies. The response strategies which were simulated in this experiment do not belong to this well-ordered class. The spare lists for this set of strategies were, in fact, completely random.

Random patterns for each stage were generated by forming the spare lists from a set of randomly selected identification numbers. With the exception of those subsystems originally located in the stage for which the spare list was being generated, all of the I. D. numbers in the system were available each time a selection was made.

The primary object of the experiment was to test the relative effectiveness of the well-ordered strategies by determining the reliability of a system using different randomly selected spare lists for each stage. Figure 24 shows the results obtained from this experiment. It can be seen from the comparison of curve (1) with curve (2) in figure 24, that the random strategy is certainly not as bad as might be suspected. This is true because the mutual dependence of any two stages on spares from any other stage is relatively low, and the spare capability is spread among all the subsystems in the system. As it was shown in Experiment III, these two factors are very effective in improving system reliability. Furthermore, it should be noted that the results shown in figure 24 correspond to a random system which was found to be the best of several such systems tested.

As a matter of interest to the investigator, another set of random strategies was simulated. This set was permitted to have a different spare list for each individual subsystem. The only restriction which was imposed was that no subsystem spare list could include the identification numbers of subsystems located in the same stage as the subsystem for which the list was being prepared.

The object of this portion of the experiment was to determine if systems using individual subsystem spare lists were potentially more reliable than those which are restricted to one list per stage. Figure 25 shows the results of the simulation. Although the results shown here do indicate that such systems offer a slight advantage in the lower region of the curve, the investigator judged the implementation problems of this type response strategy to be too formidable to merit further study at this time.

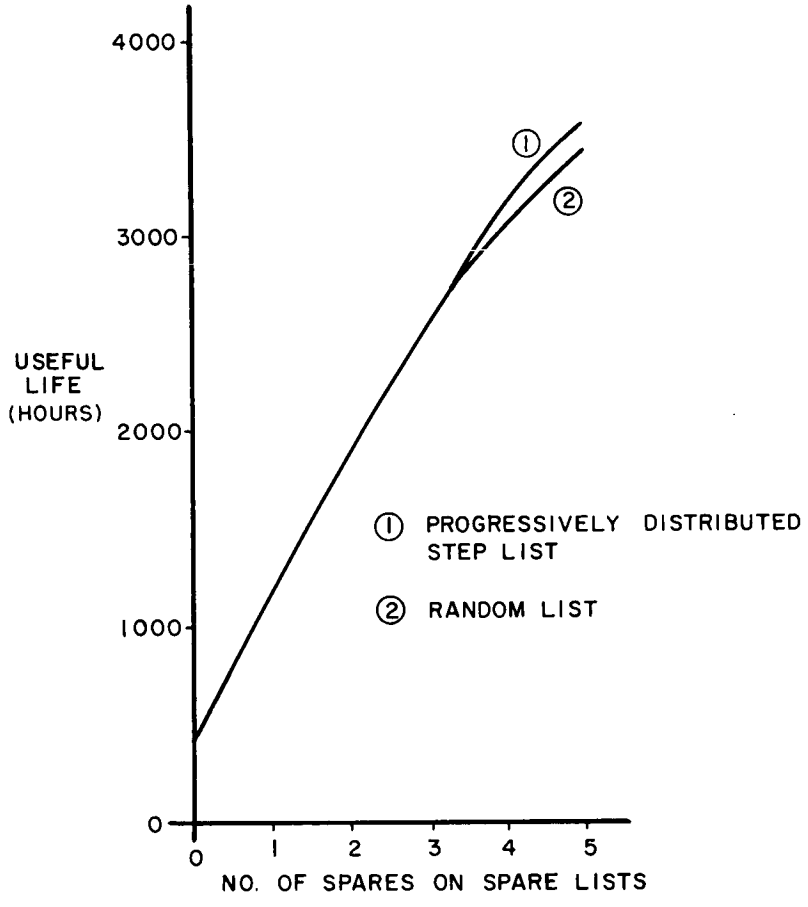


Figure 24. Comparison of Progressively Distributed Step and Random Spare Lists

2. Order-Four Systems (Experiment VI)

Higher order redundant systems may be used to reach either of two objectives. One of these objectives is the achievement of longer system life through the provision of additional failure absorption capability. The second is achievement of a high degree of instantaneous failure masking capability. There is a relationship between these two objectives which inherently results in the partial realization of both effects whenever one is sought. There is, however, a definite difference between the system structure required to maximize either effect. In the long life case, the systems are organized so

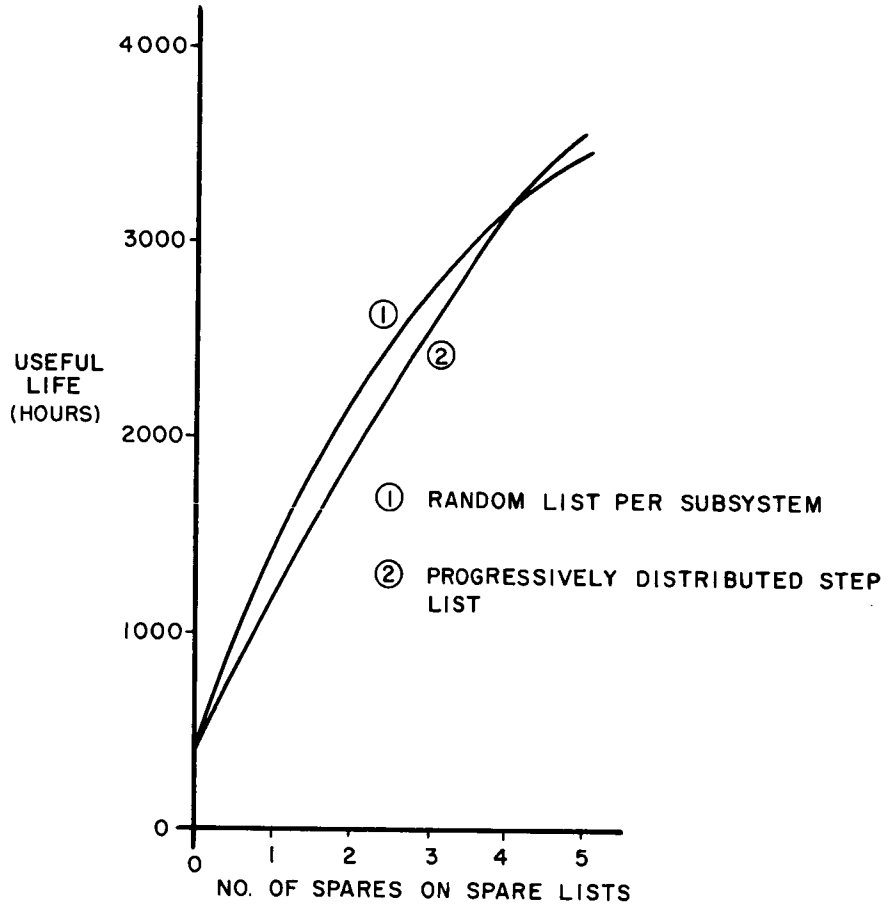


Figure 25. Comparison of Random List (Per Subsystem) and Progressively Distributed Step Lists

that no repairs are effected until a stage has experienced a subsystem failure which causes an unresolvable ambiguity to exist in that stage. This is the same switching criteria used for the order-three systems. In the high failure masking case, repairs are performed whenever a subsystem failure results in less than order-three redundancy being maintained at any stage. It should be noted that the assumption has been made in both cases that any subsystem may move only one time, i.e., may make only one repair.

Based on the preceding test results, only the progressively distributed step list response strategy was considered. The simulation tests for the order four systems were used to determine the relative potential difference between systems subject to different

failure masking restraints. Figure 26 shows the results of the test. As might be expected, the early use of spares to provide instantaneous failure masking capability precludes their later use for greatly extending the life of a system after it has experienced a relatively large number of failures.

3. Fractional Order Systems

a. Experiment VII. The serious consideration of less than order three redundancy for systems using the multiple-line configuration is virtually impossible. Certainly no consideration would be given to making any stages second order because these stages would be twice as vulnerable to failure as their non-redundant counterparts. Systems of this type are, however, quite practicable when the systems have some failure responsive capability.

Figure 27 shows two,"two-and-one-half" order system. As the figure illustrates, these systems have third-order redundancy at half their stages and second order at the other half. The use of fractional order system introduces some interesting new problems. For example, if consecutive lists are to be considered, the problem of where to put the "empty spots" in the system immediately arises. Figures 27a and 27b illustrate the two must divergent possibilities. Figure 27a schematically shows a system having the "empty spots" in the row from which spares are taken. Figure 27b shows a similar system having the "empty spots" in a different row. Figure 28 shows the curves which compare these two possibilities and the progressively distributed step list. The most significant item to be found in figure 28 is the potential improvement in useful system life over the order-three multiple-line configuration by failure responsive systems having less than order-three redundancy.

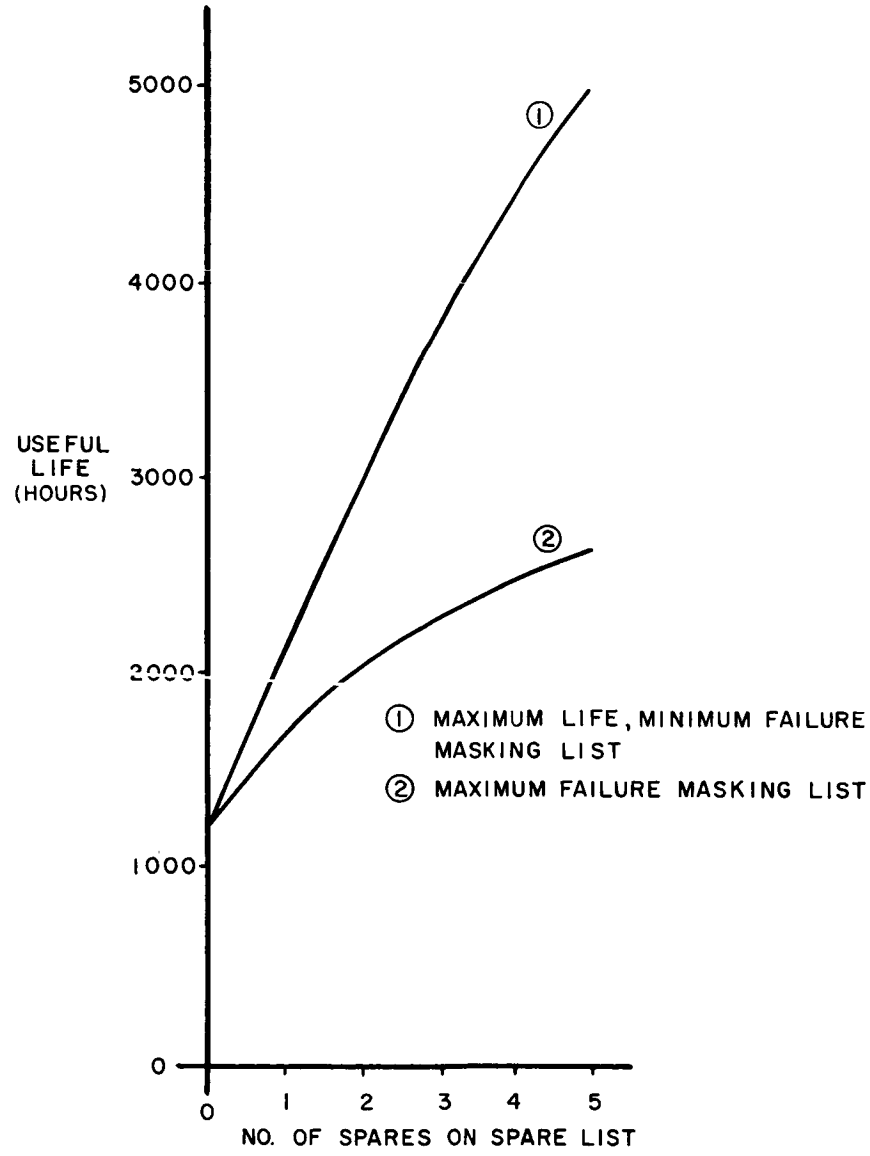


Figure 26. Comparison of Minimum and Maximum Failure Masking Lists (Order-Four Redundancy)

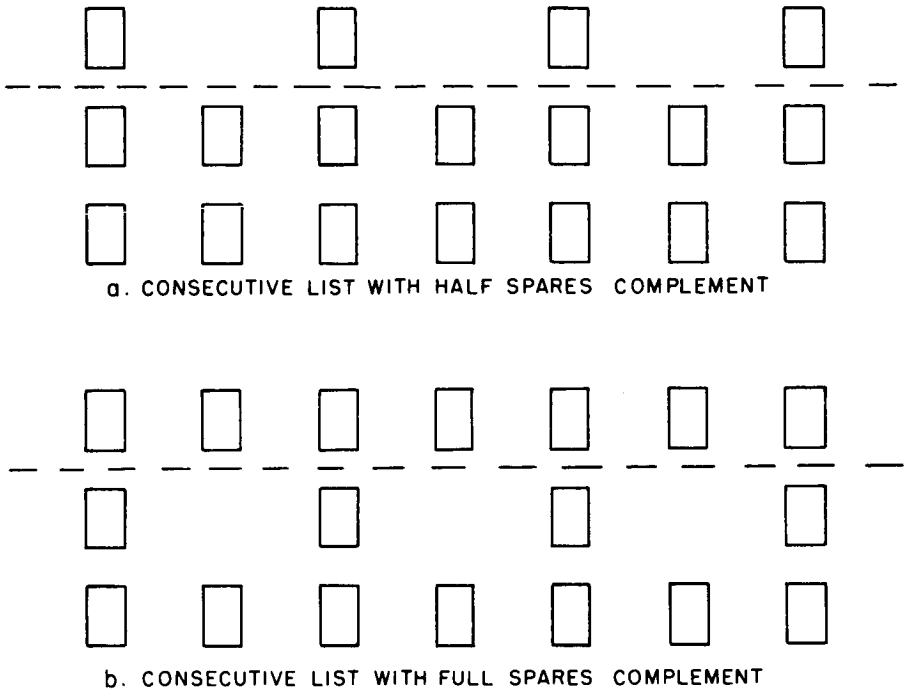


Figure 27. Sample Strategies for Order-Two-and-One-Half Redundancy Systems

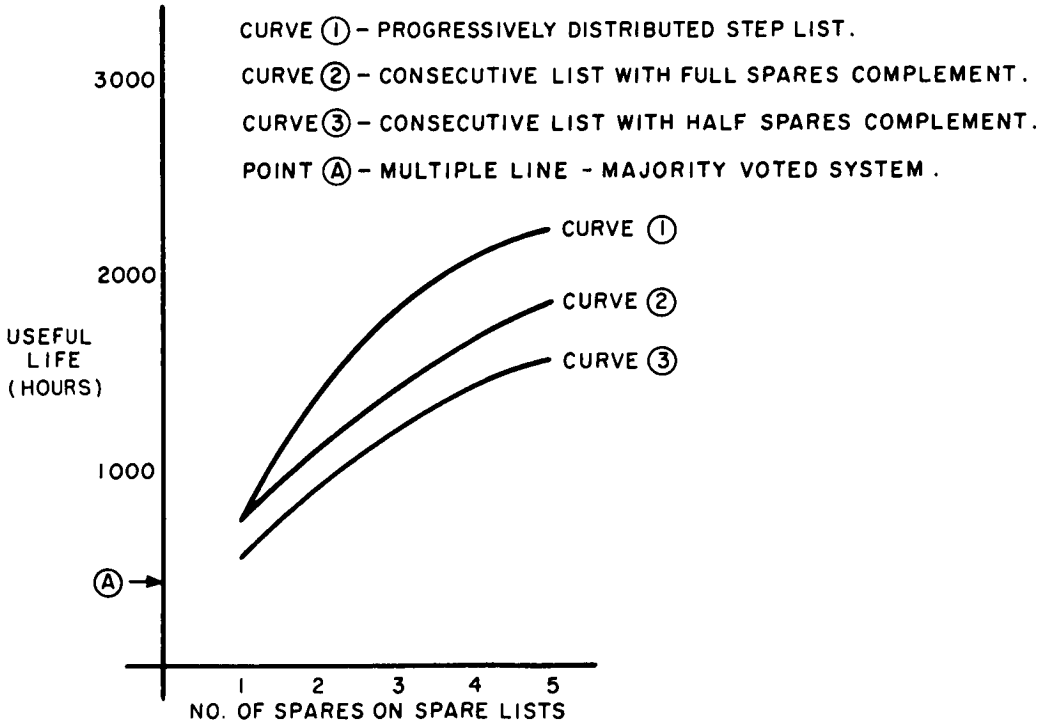


Figure 28. Comparison of Three, Order-Two-and-One-Half Failure Responsive Systems With a Third-Order Redundancy Multiple-Line System

b. Experiment VIII. In the same manner that systems can be designed using two-and-one-half-order redundancy, they can be designed using three and one half order redundancy. The primary reaons for employing this greater order of redundancy are identical to those associated with the order-four systems, i. e., longer life or higher instantaneous failure masking capability. As in the case of the order-four systems, the achievement of high instantaneous failure masking results in a shorter overall "useful" life. It is important to note, however, that even under the high degree of failure masking restraint, these systems have potentially much longer lives than either order three systems or fixed redundant (i.e., no spares) order-three-and-one-half systems. Figure 29 shows the curves illustrating all of these effects.

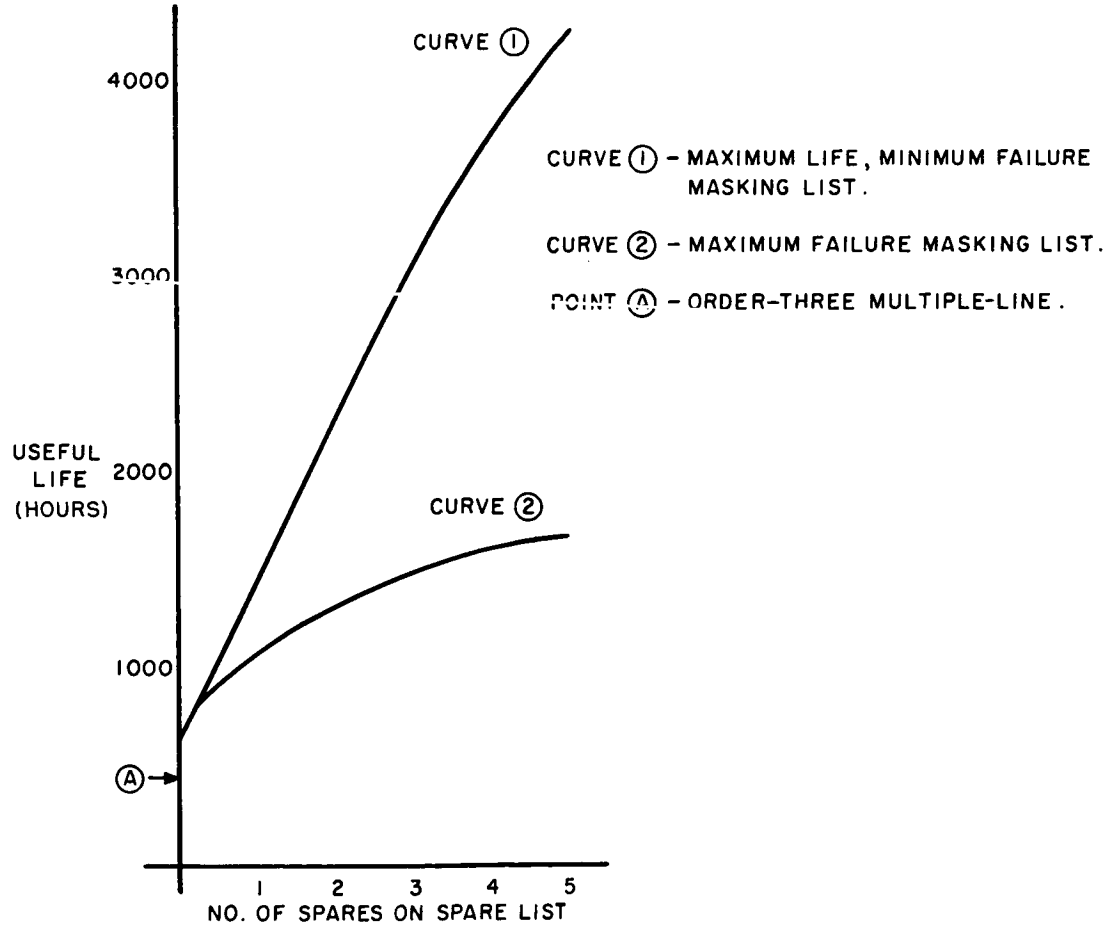


Figure 29. Comparison of Minimum and Maximum Failure Masking Lists (Order-Three-and-One-Half Redundancy)

B. Phase II Simulations

In all of the experiments which were conducted during Phase I of this study, the assumption was made that failures could not occur in the peripheral switching circuitry required to implement the response strategies. For the experiments of Phase II, this assumption has been dropped and a much less restrictive, more realistic set of three assumptions has been substituted. These assumptions may be stated as follows:

1. All the peripheral error detection switching circuitry may be divided into sections which can be uniquely associated with a single subsystem.

2. The failure of a detection and switching circuit will have the same effect as the failure of the associated subsystem.

3. The error detection and switching circuitry associated with any particular subsystem may be subdivided into a fixed portion (FSC) and a variable portion (VSC). The fixed portion represents the minimum amount of circuitry required by the subsystem to operate in its original location. The variable portion is the amount of added circuitry required by the subsystem to move to each new location.

The relative failure rates of the subsystems, the FSC and the VSC are represented in the following discussion and figures by the designations:

$$\text{Subsystem Failure Rate} = \lambda_{SS} \tag{12}$$

$$\text{FSC Failure Rate} = \lambda_{FSC} \tag{13}$$

$$\text{VSC Failure Rate} = \lambda_{VSC} \tag{14}$$

In this study, only the relative failure rates were of interest; therefore, these rates are expressed in "units", rather than in parts per hour or any other specific units.

An example of how these relative failure rates are used to compute the total relative failure rates of individual subsystems is given below. For this example, the following assumptions are made:

1. The relative failure rates are:

$$\lambda_{SS} = 1.0 \text{ Units} \quad (15)$$

$$\lambda_{FSC} = 0.2 \text{ Units} \quad (16)$$

$$\lambda_{VSC} = 0.5 \text{ Units} \quad (17)$$

2. An order-three redundancy system with four spares per stage and a progressively distributed step list is being considered. (This assumption means that two thirds of the subsystems in the system will have the capability to move to one new location and the remaining third can move to two locations.)

The total relative failure rate of the subsystems which can move to one new location is:

$$\lambda_{SS} = 1.0 \quad (18)$$

$$\lambda_{FSC} = 0.2 \quad (19)$$

$$\lambda_{VSC} = 0.5 \quad (20)$$

$$\lambda_{TOT} = 1.7 \text{ Units} \quad (21)$$

The total relative failure rate of the subsystems which can move to two new locations is:

$$\lambda_{SS} = 1.0 \quad (22)$$

$$\lambda_{FSC} = 0.2 \quad (23)$$

$$\lambda_{VSC} = 2 \times 0.5 = 1.0 \quad (24)$$

$$TOT = 2.2 \text{ Units} \quad (25)$$

These total failure rates may be interpreted to mean that the switching circuitry associated with a particular subsystem is approximately 0.70 or 1.20 times as "complex" as the

subsystem, respectively. The results of the experiments conducted during Phase I of this program indicate that the progressively distributed step list response strategy is generally the most effective of the strategies considered. For this reason, the experiments of Phase II have been limited to systems using the progressively distributed step list response strategy.

The objective of these experiments was to show that the addition of failure responsive capability would be highly beneficial to redundant system life even if the error detection and switching circuitry were relatively unreliable. To accomplish this, the relative failure rates used in the above example were applied to the order two and one half, order three, order three and one half and order four systems. Figures 30, 31, 32, and 33 show these results.

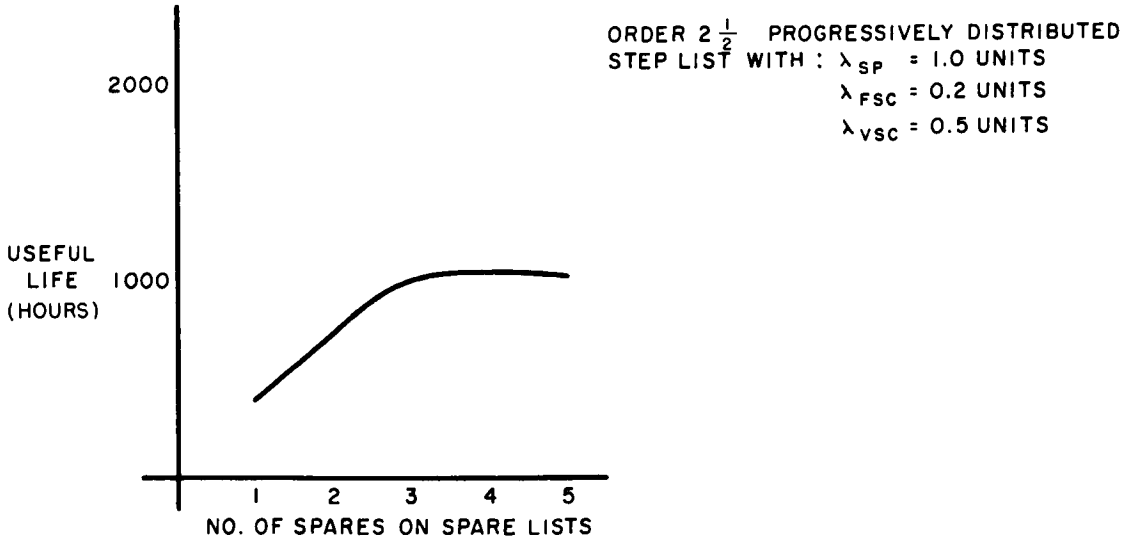


Figure 30. Order-Two-and-One-Half Progressively Distributed Step List

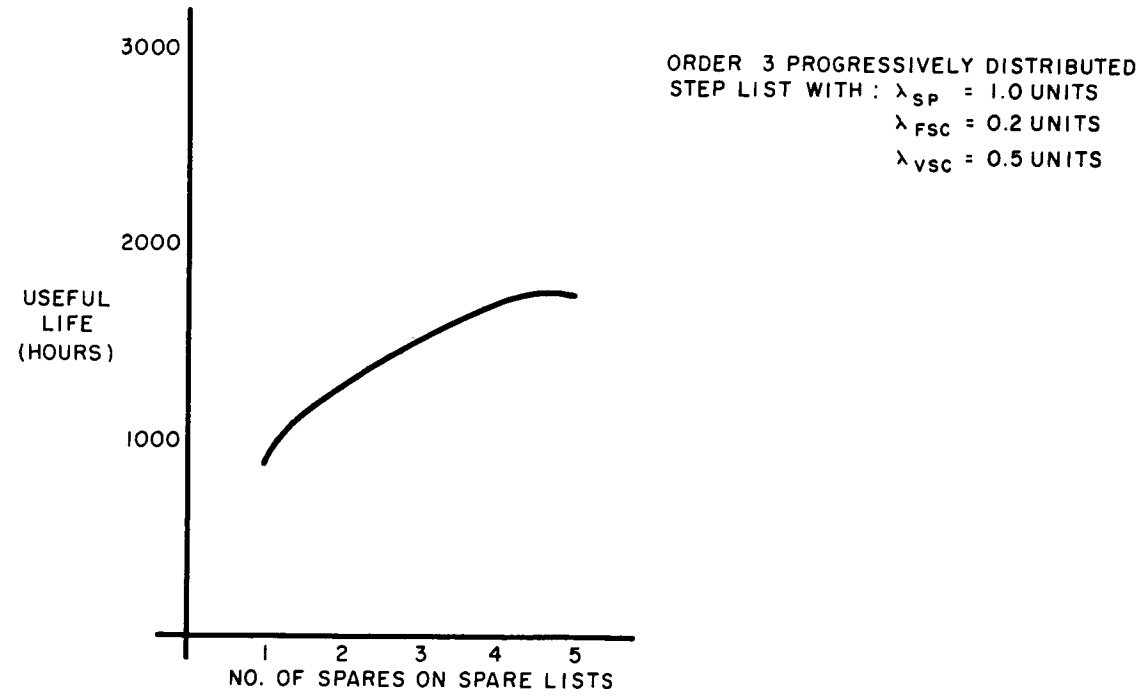


Figure 31. Order-Three Progressively Distributed Step List

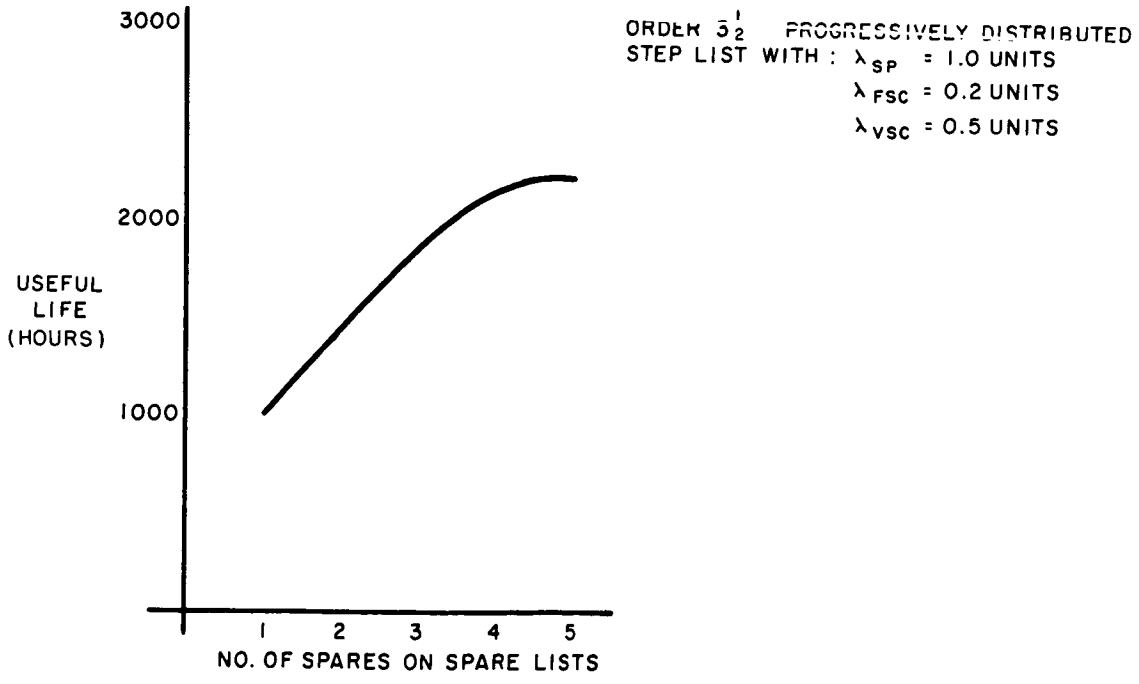


Figure 32. Order-Three-and-One-Half Progressively Distributed Step List

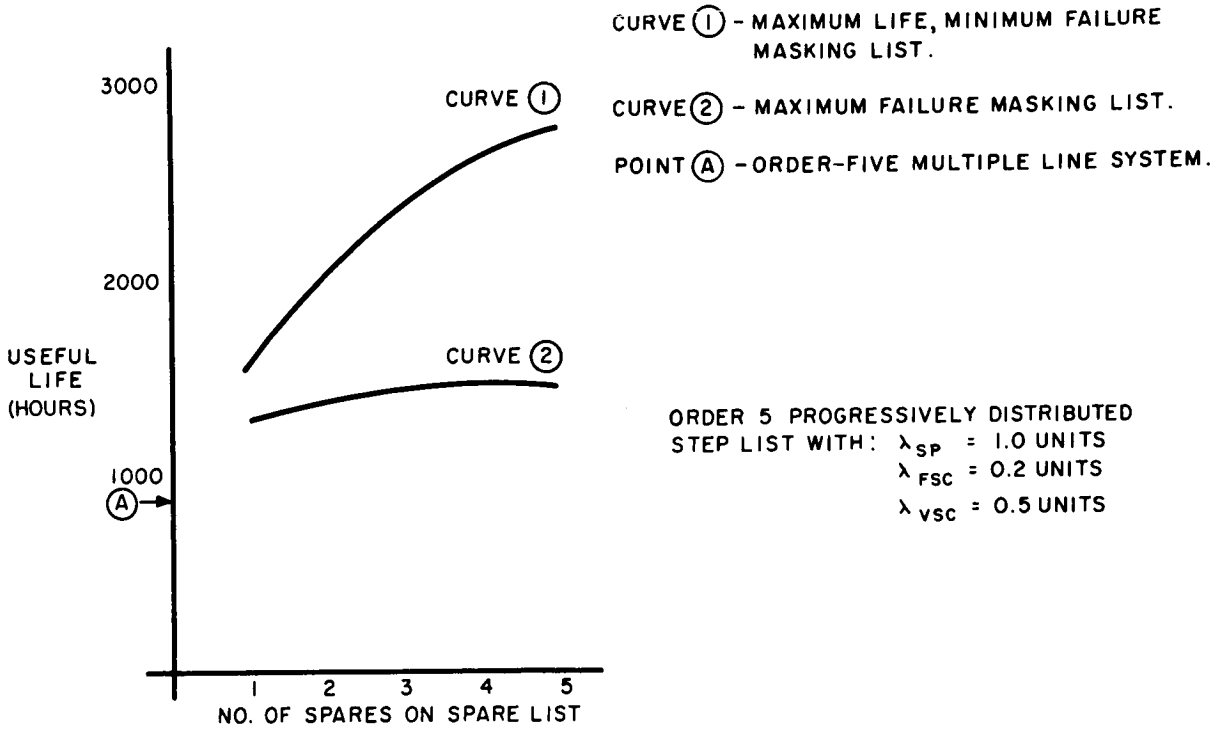


Figure 33. Order-Four Progressively Distributed Step List

IX. SUMMARY AND CONCLUSIONS

A. Summary

The primary objective of this study has been the development of a new technique for more effectively employing redundant equipment to increase the useful life of electronic digital systems. Such a technique has been devised for the class of digital systems having a high degree of homogeneity among the subsystems within each system. This thesis describes the work by the author in developing this technique and in evaluating its effect upon the reliability of this particular class of digital systems.

The sections of this thesis can be divided into three groups. The first group indicates the need for highly reliable systems and describes a few of the techniques which have been developed for achieving high reliability. This group includes a description of the failure responsive systems whose characteristics are of primary interest in this investigation.

The material presented in the second group describes several techniques which were considered in attempts to develop mathematical expressions for the reliability of failure responsive systems. The failure of the techniques to describe adequately these systems resulted in the formation of a computer simulation program. The details of this program are presented in Section VI. The final portion of this group describes the measure of effectiveness which was established as a means for comparing the different organizational strategies discussed in the thesis.

The last group contains a description of the results which have been obtained from the simulation program. The curves presented in Section VIII represent the combined results of thousands of simulated system failures. The conclusions which can be reached from observing the curves of Section VIII are listed in this final section.

B. Conclusions

The curves (figures 30, 31, 32, and 33) presented in Section VIII of this thesis show that the progressively distributed step list response strategies are the most efficient of all the well-ordered strategies which were tested. The observance of this characteristic and the recognition of the value of "rescan" capability leads to the following general conclusions:

1. The capability of individual subsystems to move to new locations should be as evenly distributed among the subsystems as possible.

2. The subsystems which are available for use as spare (or replacements) to any two stages should be chosen so that the mutual dependence by these stages on the same spares is minimized.

3. The systems should be so organized that, in normal circumstances, a subsystem will not move to the aid of a critically failed stage if its movement will leave the stage in which it is presently operating vulnerable to a single failure. A critically failed stage should have the "authority", however, to demand the movement of a spare subsystem if the movement of all of the spare subsystems available to this stage are restricted as above.

It can also be concluded that order-two-and-one-half redundant failure responsive systems may effectively replace order three redundant multiple-line systems in applications where instantaneous failure masking is not important. Conversely, applications with either high instantaneous failure masking or exceptionally long life requirements may be benefitted by employing order-three-and-one-half or order-four redundant failure responsive systems to replace order-three, or even order-five, multiple-line systems.

From figures 30, 31, 32, and 33 presented in Section VIII, it may be concluded that the beneficial effects obtained from failure responsive capability more than offsets the disadvantages inherent in the relatively complicated circuitry required for system implementation. These curves show that the useful lives of the example systems have been significantly increased over those of the corresponding examples of multiple-line systems. These increases have been realized despite relatively pessimistic assumptions regarding the reliability of the error detection and switching circuitry.

Finally, it may be concluded that the optimum number of spare subsystems which should be made available to any stage is a function of the failure rate of the peripheral circuitry relative to the failure rate of the subsystems. It can be seen from the curves in figures 30 through 33 that for systems having relatively simple subsystems the optimum number of available spare subsystems per stage will be around three to five.

Based on all of the above, the general conclusion may be drawn that failure responsive systems do employ redundant equipment more effectively than the fixed redundnat systems previously developed. The requirement of homogeneous subsystems limits the usefulness of the failure responsive technique, however, because only a relatively small class of digital systems has this homogeneous characteristic.

BIBLIOGRAPHY

1. Kemp, John C., "Redundant Digital Systems," Redundancy Techniques for Computin Systems, Spartan Books, Washington, D. C., February 1962.

2. Mann, W. C. "Systematically Introduced Redundancy in Logical Systems," 1961 IRE International Conv. Rec., 9, P + 2, March 1961.

3. McReynolds, J., "Evaluation of the Majority Principle as a Technique for Improving Digital System Reliability", Hycon Eastern Inc., (now Hermes Electronics, a Division of Itek), Cambridge, Mass., July 8, 1958.

4. Ramer, Paul and Carlo Michel, "Improved System Reliability by Means of Equipment Redundancy," Electronic Systems and Products Division, Martin Marietta Corp., October 1963.

5. Jensen, P. A., "Bibliography on Redundancy Techniques", Redundancy Techniques for Computing Systems, Spartan Books, Washington, D.C., February 1962.

6. Kletsky, E. J., "Self-Repairing Machines," Final Report Part One (RADC-TR-61-01B), Syracuse University Research Institute, Syracuse, New York, April 1961.

7. Seshu, Sundaram, "Self-Repairing Machines", Final Report Part Two (RADC-TR-91B), Syracuse University Research Institute, Syracuse, New York, April 1961.

8. Lofgren, Lars, "Qualitative Limits for Automatic Error Correction Self-Repair," Tech. Report 6, University of Illinois Electrical Engineering Research Laboratories, Urbana, Illinois, June 1960.

9. Lofgren, Lars, "Kinematic and Tessellation Models of Self-Repair," Tech. Report 8, University of Illinois Electrical Engineering Research Laboratories, Urbana, Illinois, December 1961.

10. Landers, R. R., "Machines That Grow", Machine Design Vol. 34, No. 16, July 6, 1962.

11. Esary, J. D. and F. Proschan, "The Reliability of Coherent Systems," Redundancy Techniques for Computing Systems, Ed. by R. H. Wilcox and W. C. Mann, Spartan Books, Washington, D. C. 1962 (pp. 47-61).

12. Mood, A. M., Introduction to the Theory of Statistics, McGraw-Hill Book Co., Inc., New York, 1950, page 107.

13. Sasieni, Maurice, et al, Operations Research Methods and Problems, John Wiley and Sons, Inc., New York, 1961, page 126.

APPENDIX

The assumption has been made in this thesis that individual subsystems fail at random times but at some constant rate, lamda (λ). The fact that the rate is independent of time implies that the probability of failure (13) of any one subsystem in any interval Δt is

$$\Pr(\text{failure}) = \lambda \Delta t \tag{26}$$

if the interval is sufficiently small. If a subsystem failure is known to have occurred in some interval Δt about the time t, and one is interested in the conditional probability that the failure occurred in one of a set of identical subsystems, the following relationship can be seen to exist:

$$P(\text{failure of the ith subsystem/one subsystem has failed}) = \frac{\lambda_i \Delta t}{\sum_{\text{All } i} \lambda_i \Delta t} \tag{27}$$

but

$$\frac{\lambda_i \Delta t}{\sum_{\text{All } i} \lambda_i \Delta t} = \frac{\lambda_i \Delta t}{\Delta t \sum_{\text{All } i} \lambda_i} = \frac{\lambda_i}{\sum_{\text{All } i} \lambda_i} \tag{28}$$

therefore,

$$P(i/1) = \frac{\lambda_i}{\sum_{\text{All } i} \lambda_i} \tag{29}$$

NASA-Langley, 1965 CR-343