DOE/SF/21032--T46

GENERAL ELECTRIC

XL-898-10483
Mr. S. G. Harbison
September 30, 1981

MASTER

Page -2-

cc: Director, Nuclear and Magnetic Fusion Division, SAN
Director, Safety & Physics Division, RRT (B-107)
H. Alter, RRT (B-107)
W. P. Carroll, RRT, (B-107)
Director, FRS-TMC, ANL
A. E. Dubberley, FRS-TMC, ANL (2)
Manager, Technology & Special Projects, GE-ARSD

DOE/SF/71032--T46

DE82 011801

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

ANALYTICAL APPROACH FOR CONFIRMING THE
ACHIEVEMENT OF LMFBR RELIABILITY GOALS

G. E. Ingram
J. G. Elerath
A. P. Wood

General Electric Company
Advanced Reactors Systems Department
Sunnyvale, California

DISCLAIMER

This book was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof nor any of their employees makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

ANALYTICAL APPROACH FOR CONFIRMING THE ACHIEVEMENT OF LMFBR RELIABILITY GOALS

I. INTRODUCTION

Confirming the achievement of system reliability goals has traditionally been through a process of performing "reliability tests" at the system level of interest and then analyzing the test results to make statements regarding the achieved level of system reliability. This process was a viable one for systems having one or more of the following characteristics:

- (1) The system is relatively inexpensive so that the hardware and reliability tests may be replicated a sufficient number of times to draw significant inferences regarding its achieved reliability.
- (2) The system is amenable to a non-destructive test program such that significant statistical inferences may be drawn from the reliability test results regarding its achieved reliability.
- (3) For systems having very high reliabilities (e.g. long times before/ between failures), the necessary reliability testing can be accelerated so that the necessary test data can be acquired in a relatively short period of time and at a relatively low cost.

Systems such as the LMFBR, however, do not generally have these characteristics. That is:

- (1) The overall system is very expensive (usually made up of a relatively few, very expensive subsystems) and consequently a limited number will be produced.
- (2) The system is very complex making it virtually impossible to obtain nondestructive test results from which significant statistical inferences can be drawn.

rences may be drawn regarding the achieved system reliability.

- (3) The required level of system reliability for the LMFBR is such that the time and cost required to obtain significant results from traditional reliability testing is prohibitive. Current accelerated testing techniques at the system level are not adequate at this time to remove this obstacle.

Since it is not economically feasible, or even physically possible in some cases, to confirm the achievement of system reliability goals by traditional reliability test programs, it is necessary to demonstrate reliability achievement through analysis. This means that analytical techniques that simulate the reliability of a real physical system (man and machine) operating within real environmental (system induced or natural) conditions must be available and employed. The underlining of the word real is to emphasize that the analytical techniques must represent the reliability (or conversely the probability of failure) of the operational system as close to reality as possible. For example, if the logical representation of the reliability of the system contains inter-relationships (or conditionalities) among its various elements, then the analytical approach to determine the reliability of the system must properly reflect these inter-relationships. If a valve or switch, for example, is in an open state (failed or otherwise) then it cannot also be in a closed state. Or, if a pump is in a non-operational mode (e.g. fails to start on demand) then it cannot also be in an operational mode which would allow it to fail after t hours. The analytical techniques must be capable of handling these and other types of dependencies.

Realism such as the monitoring and testing of components during the operation of the system must be incorporated into the analytical techniques to assure that the effects of such monitoring and testing (good and/or bad) have been accounted for in the reliability evaluation. This allows the designer/operator to make more rational decisions regarding the configuration and/or operation of the system. The analytical techniques should be amenable to changes in configuration and/or operation of the system to allow the designer/operator to explore the various options available to him and the resulting effects, associated with each option, upon system reliability.

The analytical approach recommended in this report addresses the above issues and describes the techniques used for their implementation. Section II gives an overview of the recommended approach, Section III gives a detailed description of the techniques that have been developed during fiscal year 1981 for implementing this approach, Section IV presents an example of the use of "FRANCALC", one of the computer codes that was developed to implement this approach, and Section V presents recommendations of areas for further development.

II. RECOMMENDED APPROACH

The approach, recommended by GE-ARSD, for confirming the achievement of LMFBR reliability goals relies upon a comprehensive understanding of the physical and operational characteristics of the system and the environments to which the system will be subjected during its operational life. This kind of understanding is required for an approach based on system hardware testing or analyses, as recommended in this report. However, for a system as complex and expensive as the LMFBR, an approach which relies primarily on system hardware testing would be prohibitive both in cost and time to obtain the required system reliability test information. By using an analytical approach, results of tests (reliability and functional) at a low level within the specific system of interest, as well as results from other similar systems can be used to form the data base for confirming the achievement of the system reliability goals. This data, along with information relating to the design characteristics and operating environments of the specific system, will be used in the assessment of the system's reliability.

The selection of this approach stems from extensive knowledge of analytic reliability methods and experience in this application. This knowledge base has supported the mathematic justification and implementation of desired analytic capabilities, while experience in analysis of Reactor Shutdown and Shutdown Heat Removal Systems of the Clinch River Breeder Reactor, the GE Prototype Large Breeder Reactor and the Large Developmental Plant has shown the harsh realities of applying these methods. Each analysis identified another area of the approach which needed improvement.

In the first application of analytic methods to a shutdown heat removal system, certain specific capabilities were established as a basis for the selection of a method. Five different approaches were evaluated for their compliance to the desired capabilities. As a result, a failure state model approach was identified as that which best met the capabilities and a computer code was developed for the evaluation of the models. A Markov approach was selected as the second choice and identified for future development.

Since the original selection process, many changes have taken place in the computer code for evaluation of failures state models and a code has been developed for Markov evaluation. However, the conclusion that realistic reliability evaluations can be mathematically tractable and economically viable using a failure state approach has not changed.

As computer code development has progressed, so has the ability to realistically model complex systems. Currently, with the merging of the codes FRANTIC (Reference 1) and PROBCALC (Reference 2), the approach provides the analyst the following features:

- 1) Traceability - A failure state approach provides a compact model (graphically) which usually resembles the system to a great extent. This makes the model easy to understand and to explain to others.
- 2) Time Phased Missions - Mission time phasing is often required to model changes of:
 - a) failure rates in time
 - b) system redundancy in time
 - c) component and system initial conditions
- 3) Instantaneous and Average Unavailabilities (Unreliabilities) - Both types can be evaluated, including the effect of testing (sequential and simultaneous) and restoration of items found failed at test.
- 4) Common Cause Failure - Many identified common cause failures can be explicitly modeled. (This capability will be significantly expanded in FY 1982.)
- 5) Mutually Exclusive Systems - Systems, subsystems and components which cannot operate simultaneously can be modeled.

The implementations of these features is made by virtue of the following capabilities in the current computer codes:

- (1) Treatment of dependencies in which,
 - (a) the same element having only one failure mode appears more than once in the probability block diagram, and
 - (b) the same element having two failure modes appears more than once in the probability block diagram.

It should be noted that the capability stated in "a" had been developed previous to FY 1981 but the computation efficiency was significantly improved during FY'81 to facilitate the treatment of larger numbers of dependent elements at the same computation time and cost. This led to the efficient inclusion of the capability stated in "b".

- (2) The treatment of M-of-N redundancies in which,
 - (a) all N elements have the same failure/success probability - therefore, the probability of M of the N elements failing/succeeding is a simple and short computation that can be made in the presence (or absence) of dependencies as stated in "(1) a & b".
 - (b) all N elements do not have the same failure/success probability - however, the probability of M of the N elements failing/succeeding is a relatively simple and short calculation that also can be made in the presence (or absence) of dependencies as stated in "(1) a & b".

- (3) Accounting for changes in reliability (good or bad) through a program of testing/monitoring and repair of specified elements of the system. This capability was developed by merging the two computer codes "FRANTIC" and "PROBCALC". FRANTIC calculates both the instantaneous and average unavailability contributions due to failures, repairs, and periodic tests. These unavailabilities serve as inputs into a user-supplied subroutine which calculates system unavailability/reliability. PROBCALC serves the function of the user-supplied subroutine and provides the capability to evaluate large or small systems.
- (4) Calculation of a probability density function defining system reliability based on probabilistic descriptions of input component reliabilities rather than point estimates of component reliabilities. This capability was developed by, first, updating a computer program (RVCOMB) that had been designed to combine random variables by a numerical process. This program (RVCOMB) was then merged with PROBCALC to allow input component reliabilities (or failure probabilities) to PROBCALC to be in the form of probability density functions, rather than simply point estimates, thus producing an output system reliability (or probability of failure) in the form of a probability density function rather than a single value.

The general analytical approach may be regarded as a series of steps in which the following information is obtained or developed:

- (1) A complete system description which includes hardware configuration, operating environments, operational and maintenance procedures, and procedures for monitoring, testing, etc. is required.
- (2) Using the information of item (1), failure modes and effects analyses are performed to determine how the system can fail and the effects of each of these failures upon other parts of the system.

- (3) The results of item (2) are used to develop probability block diagrams (failure or success) that define the various failure or success paths of the system in performing its intended functions. From these diagrams, the overall probability of system failure or success can be determined from the probabilities associated with the various elements that are indicated.
- (4) Failure probabilities (or failure rates) associated with the various elements of the system (hardware and human) are collected from all available sources to perform the necessary system calculations. These sources include results of tests on hardware to be used in the specific system of interest as well as results of tests on similar hardware from other similar systems.
- (5) Enumerate the events that will place demands upon the system. It is necessary to know the probabilities of occurrence of the events for a given time period (or their frequencies) for use in the probability analysis. In addition to the probability or frequency of an event, it is important to know the condition of the system given that the event has occurred (e.g., Has the system been degraded by having lost one or more components as a result of the event?) The probability (or frequency) of system failure, therefore, is computed using the system condition associated with each of the postulated events.
- (6) Sum (over all events) the products of the probability of occurrence (or frequency) of an event and the probability of system failure when the event has occurred. The summation is the overall probability of system failure. As stated earlier, careful attention must be given to assure that both hardware and functional redundancy, and interdependence of elements throughout the system are accounted for.

As an example of system complexity that this approach can evaluate, consider the Reactor Shutdown System (RSS) of the current LDP. The RSS is comprised of a primary and secondary subsystem and each must be capable of automatically shutting down the reactor when one or more of a prescribed set of events have occurred. The events are those that would have a detrimental effect upon the

system if the reactor were not shutdown. Some typical events of concern are control assembly withdrawal at full power, large gas bubble through reactor, loss of off-site electrical power, spurious pump trip, loss of normal feedwater, steam line pipe break, and misoperation of reactor plant controller. The system must be capable of sensing the events and transmitting the appropriate signals to the control rod system in order to insert the required number of rods to shut the reactor down. Adequate redundancy must be provided for reactor safety while achieving an availability that will result in an economically viable power producing plant.

The events of concern are detected by means of sensors that sense trip parameters which indicate the presence of the particular event(s). A few parameters are sensed by more than one type of sensor and some of the sensors are used to sense more than one parameter. Most of the trip parameters will indicate the presence of more than one event of concern. It is this versatility of the sensors and trip parameters that provide the functional redundancy and much of the diversity within the RSS.

Due to the fact that some sensors are used to sense more than one parameter/event, the model must properly reflect the existing interdependencies. These interdependencies can exist, for example, because the actual sensor is the same or, even if the hardware is different, the sensor type is the same for detecting certain events. Dependencies also occur between the primary and secondary shutdown systems. For example, each system senses many of the same events and some of the sensors within each system are of the same type though physically different hardware.

In Section IV of this report is a hypothetic, simplified example of a Shutdown Heat Removal System (SHRS). It has a two-fold purpose: 1) to illustrate the actual implementation of the steps of the approach and 2) to illustrate the application of a newly developed computer code called FRANCALC. (The code is discussed in Section III-C as well). Inherent to this sample analysis is the acknowledgement that not every analysis needs to employ all aspects of the approach above. Each system must be understood well enough from the outset for the analyst to know which parts of the approach to employ. Although the purpose of the sample SHRS is intended to illustrate the approach, many of the concepts it embodies must be understood before it will make sense. Therefore, it has been placed last in this report, after the explanations of the areas of methods development in FY 1981.

III. AREAS OF METHODS DEVELOPMENT DURING FY'81

During the past fiscal year, the following areas of methods development have been pursued:

- A. The proper treatment of mutually exclusive failure modes in reliability analyses.
- B. An efficient technique for treating M-outof-N redundancy within the context of analyses having many dependent elements.
- C. Accounting for changes in reliability in systems which can be tested/monitored and repaired.
- D. Defining system reliability as a range of possible values, each with an associated probability of occurrence.
- E. Improvement of the computational efficiency to enable an increase in the number of dependent elements that may be treated in a reliability analysis without causing excessive computational time and cost.
- F. Improvement of the efficiency of the MARK computer code to perform reliability analyses of repairable systems when operations are described by Markov processes.

The work performed in each of the above areas of methods development will be discussed in detail in the following paragraphs. Each area of development is supported by a computer program user's manual. In one area (combining FRANTIC and PROBCALC) a draft of the user's manual has been prepared and will be available in the near future. The supporting user's manual for the other areas will be a revision and update of the current PROBCALC user's manual. This revision and update is currently being done and will be issued at a later date.

A. Mutually Exclusive Failure Modes

A commonly encountered dependency that occurs in reliability analyses is the mutually exclusive failure mode. For example, a valve or switch may fail in the "open" position or in the "closed" position. The effect of each of these failures upon the system may be quite different and, therefore, must be treated separately in the reliability model. However, if the valve or switch has failed in one position it cannot, at the same time, have failed in the other position and the probability calculation for the system must reflect this fact. The failure modes, in this case, are said to be mutually exclusive. If event "A" is failure in the open position and event "B" is failure in the closed position then the probability of A given that B has occurred, $P(A/B)$, is zero or, inversely, the probability of B given that A has occurred, $P(B/A)$, is also zero. The primary function that the computer program must perform to properly treat mutually exclusive failure modes, therefore, is to insure that the mutually exclusive failure modes do not occur simultaneously in any of the calculations.

For example, as PROBCALC was previously programmed, the dependencies resulted from the fact the same element appeared more than once within the probability block diagrams (PBDs) for the system. The function of the computer program, in this case, was to treat the calculations as 2^n mutually exclusive states. That is, each of the n elements had two possible states, or modes (one failure mode and one success mode). The program assured that for each of the 2^n states, a given element was in the same state, or mode, throughout the calculations. The inclusion of mutually exclusive failure modes increases the possible number of states that an element may be in to three (two failure modes and one success mode).

As stated previously, the n single failure mode elements that appeared more than once throughout the probability block diagrams constituted 2^n mutually exclusive states. In the updated version of PROBCALC, the probability block diagrams can have m elements having two failure modes each (three states total) thus creating 3^m mutually exclusive states as well as the 2^n mutually exclusive states created by the n single failure mode

dependent elements. This yields a total of $2^n \times 3^m$ mutually exclusive states. If n and m are large, as they often are, this is a very large number of states that could result in an excessive amount of computation time. In order to avoid that problem, techniques were developed to order the calculations in such a way to assure a specified accuracy with the minimum number of calculations. This will be discussed in section III-E.

B. M-out-of-N Redundancy

As PROBCALC was previously programmed, the user was required to describe by means of the probability block diagram (PBD) the M-out-of-N redundancy in which M was greater than unity. This was done by showing $N!/(N-M)! \times M!$ parallel paths (the number of combinations of N things taken M at a time) with each path having in series M of the original N total paths. For example, if A, B, and C were three possible heat removal paths and at least two of the three paths were required to remove sufficient heat then the PBD would have been shown as

The shaded boxes indicate the dependencies (the same item appearing more than once) and the calculation was made on this basis. However, it is more convenient to simply specify the 2-out-of-3 redundancy (for this example) and have a separate routine in the program perform the calculation. The PBDs are simpler to draw, there is a little less input to the program required, and the computation time is shortened some, depending upon the values of M and N .

The capability has been added to the program to take care of the case in which all N elements have the same probability and also the case for which they are not all the same. The user, however, has the option of using the program in the same way as it was originally programmed for M-out-of-N redundancy.

C. Accounting for Monitoring, Testing, and Repairing

The probability that a component or system will fail to perform its task when called upon is termed unreliability or unavailability. Most reliability analyses include an estimate of average unavailability but do not give an account of instantaneous unavailability. This could be a serious omission since a system with low average unavailability could have some high peak unavailabilities which render the system vulnerable at certain points in time. Figure 1 (taken from Reference 1) shows two systems which have the same average unavailabilities. The system with higher peak unavailabilities is generally considered to be the poorer of the two systems. Cases like this are especially prevalent when some of the components undergo periodic testing. The unavailability function in the left half of Figure 1 might come from a system with staggered test intervals while the unavailability function on the right might come from a system in which several components are tested simultaneously. Reliability analyses can be improved by considering instantaneous unavailability.

The FRANTIC computer code (Reference 1) was developed to calculate both the instantaneous unavailability and the average unavailability of a system and to give a breakdown of the unavailability contributions from failures, repairs, and tests. FRANTIC determines the unavailability for each of four types of components, including (1) constant unavailability components (demand or cyclic failures), (2) non-repairable components, (3) continuously monitored components with or without immediate commencement of repair when a failure occurs, and (4) periodically tested components. FRANTIC can also determine the unavailability of components which have both time-dependent failure modes and cyclic-type failure modes. In the evaluation of continuously monitored components, FRANTIC also allows for the possibility that the failure might not be detected. Several options are available for calculating the unavailability of periodically tested components. The testing schedule, test duration, and repair time are necessary inputs, but FRANTIC can also account for the probability of a test-caused failure, of detection inefficiency, and of the possibility of overriding a test and using the component if a demand occurs.

Figure 1
Two Systems with the Same Average Unavailability
($\bar{q}$) but with Different Instantaneous Unavailabilities ($q(t)$)

FRANTICs evaluation process begins with the calculation of all component unavailabilities that are to be used in the calculation of the system unavailability. These component unavailabilities are calculated by FRANTIC. They are then used as input to an equation for system unavailability, which must be supplied by the user. Because each such calculation represents a single point in time, this process is iterated to generate the system unavailability as a function of time.

One of the draw backs with the FRANTIC code is the need for the user to supply an equation representing the modelled system reliability. For large, complex systems such as LMFBR Heat Removal Systems the task of deriving an equation is virtually insurmountable from both time and accuracy aspects. The purpose of this task was to replace the user-supplied subroutine with the PROBCALC computer code (Reference 2).

While PROBCALC does not develop a closed form equation it does numerically evaluate the modelled system. Thus, the combination of FRANTIC with PROBCALC into a computer code, called FRANCALC, provides the capability to analyze large complex systems including dependencies, mutually exclusive events, general M-out-of-N redundancy as well as the various aspects of testing discussed above.

To use FRANCALC, the analyst first determines which components will have their unavailability determined by FRANTIC and provides the necessary data. The computer code then determines time points at which the system unavailability function changes. This happens when any component starts a test or repair cycle. In Figure 1, there are 4 points where this occurs in the graph on the left and 8 points where it occurs in the graph on the right. (FRANTIC assumes linearity between these time points, which is a slight conservatism when redundancy exists, see Reference 1.) PROBCALC is used to calculate system unavailability immediately before and after each of these time points using components unavailabilities which are calculated in FRANTIC or input directly to PROBCALC. FRANTIC then uses these system unavailabilities to compute average and instantaneous system unavailabilities.

Frequently, one wishes to know not only the probability of a system being available at different points in time but also the probability of a system being available and being able to perform a task of some duration. In this case, FRANTIC determines time points as before, but at each of those points, PROBCALC calculates the probability of the system performing its mission given the component unavailabilities at the beginning of the mission. This allows, for example, the following question to be answered: What is the probability that a pump can be turned on and operated for four hours if it is called upon at random twice a year? The example in Section IV illustrates the use of FRANCALC.

D. System Reliability as a Probability Density Function

Due to the uncertainties associated with the data that is used in the performance of reliability analyses (such as failure rates), it is often useful to be able to input these uncertainties to the analyses to determine the uncertainty associated with the resulting system reliability. This capability is being added to PROBCALC but is not yet completed. A subroutine has been prepared for use with PROBCALC but there are still some refinements necessary to make it work efficiently.

The technique used for combining the random variable (rv_s) is a numerical process in which the rv_s are systematically combined pairwise. Two rv_s are combined according to the appropriate mathematical function (e.g. add, subtract, multiply, etc.) resulting in a new rv which is defined by a range of values, each with an associated probability of occurrence. This process continues until the system reliability value has been obtained.

Consider the following simplified example by way of explanation of the process (A,B,C,&D are rv_s).

As indicated by the probability block diagram (PBD) rv_s A and B are combined as a product to form a new rv which will be called W (i.e. $A \times B = W$). Also, rv_s C and D are combined as a product to form a new rv which will be called Y (i.e., $C \times D = Y$). W and Y are each subtracted from unity and the product of these quantities is called Z (i.e. $(1-W) \times (1-Y) = Z$). Z is subtracted from unity, which is the probability for the system. If the PBD is a failure diagram then $(1-Z) = \bar{R}$, the unreliability, or if it is a success diagram then $(1-Z) = R$, the reliability.

The random variables are combined by considering each of them as a set of discrete values with associated probabilities of occurrences. If, for example, each of the rv_s A and B were defined by ten discrete values then $A \times B = W$ would have one hundred values which would be appropriately regrouped into ten values. This avoids an excessive number of computations when the number of random variables is large. When the random variable is subtracted from unity, the computation is simply one of subtracting each of the discrete values from unity but retaining the same probability of occurrence. However, when two random variables are combined, the probability associated with each outcome is the product of the probabilities associated with the discrete values entering into the outcome if the rv_s are independent and the product of conditional probabilities if they are not independent rv_s .

E. Improvement of PROBCALC Computational Efficiency

Computational efficiency should have a high priority in the development of any calculation process, whether it be computerized or not. However, when the process can encompass an extremely large number of computations, e.g. often in excess of 10^{12} sets of computations, then even on the fastest electronic computers it is very important to shortcut the process while achieving the desired level of accuracy. In the areas of methods development reported in this document, computational efficiency was given constant consideration.

The computer code, PROBCALC, was originally developed to perform probability calculations which had many interdependencies among elements of the model.

The type of element treated was one having a single failure mode which appeared more than once within the model (i.e., a dependent element). The calculation process was one of treating the "n" dependent elements simultaneously to consider all possible states of the system. Since each dependent element had two states (one failure and one success), the total number of system states to consider in the calculation was 2^n . If there were forty dependent elements appearing in the system model (i.e., $n=40$), which was not unusual in some early analyses of the Shutdown Heat Removal System (SHRS), the total possible number of system states was 2^{40} , which is approximately 1.1×10^{12} states. If it had been necessary to do the computations for that many system states the analysis would have been intractable. However, by ordering the numerical process to consider the highest probabilities first, it was possible to achieve third significant figure accuracy in the system unreliability by considering only a small fraction of the total number of system states.

The updated version of PROBCALC now includes the capability to consider another commonly encountered type of dependent element (i.e., an element having two mutually exclusive failure modes.) The effect of each failure mode upon the system can be quite different and, therefore, must be represented properly within the system probability model. This type of element produces three possible mutually exclusive states (i.e., two failures and one success). The number of system states that can be created by "m" of these elements is 3^m . The total potential number of system states created by the two types of dependencies (i.e., n- one failure mode, and m- two failure mode elements) is $2^n \times 3^m$. If the overall number of dependent elements is the same (e.g., forty, $n+m=40$), then the total number of system states will be even greater; that is $2^n \times 3^m = 2^{(n+m)} \times 1.5^m = 2^{40} \times 1.5^m$. This posed an even greater computation time and cost problem so the computational efficiency of the program was reconsidered.

Previous to this update, the process had been to systematically (after ordering the failure probabilities from high to low) consider the failure of combinations of 0, 1, 2, ---, i, ---, n dependent elements. For example, there are $n!/(n-i)!i!$ ways to have exactly "i" elements fail out of a total of "n" elements. The

process of considering first zero failures, then all possible ways of getting one, two, etc. failures was continued until the total probability of the system states not yet considered (i.e. one minus those so far considered) was sufficiently small--that is, the desired accuracy. However, some combinations of, say, three dependent element failures (i.e. those having low individual failure probabilities) had lower probabilities of occurrence than some combinations of four or more dependent element failures (i.e., those having high individual failure probabilities). The efficiency improvement, therefore, was to check the magnitude of the combination's probability of occurrence and move on to the next number of failures when the magnitude was sufficiently small. The criteria for sufficiency is determined by the program based on the desired overall accuracy and the relative values of the dependent element failure probabilities. This procedure reduces the total number of system states to be considered by a significant amount. Without the computational efficiencies provided in the update version of PROBCALC, it would not be feasible to perform the functions of the computer code FRANCALC for large complex systems having dependencies.

F. Reliability Analysis of Repairable Systems When Operations are Described by Markov Processes

The MARK computer code has been developed in ARSD to perform the reliability analysis of repairable systems when their operations are described by Markov processes. The Markov approach requires the solution of a set of linked linear differential equations, the number of which equals the number of states required to describe the system operation. For a complex power plant safety system operation, the number of states required to set up a Markov model can be very large, requiring expensive computer time for quantitative evaluation. The computation effort can be substantially reduced if the Markov model describing the system operation is mergeable.

A mergeable Markov process is a random process. The states of this random process can be grouped together to form superstates in such a way that the result is also a Markov process. In mathematical terms this can be written

as follows:

$$\sum_{j \in S_l} P_{ij} = \sum_{j \in S_l} P_{i'j} \quad \text{for } i, i' \in S_k$$

where i and j are original states and k and l are super states of the new Markov process. P_{ij} is the transition probability from state i to state j .

In general it is very difficult to determine all of the mergeable states of a Markov process. There is no straight forward approach for generating all of the possible partitions of the set of transition states and testing for mergeability criteria. A search algorithm has been developed for determining the mergeable states of Markov processes. It has been tested and found to yield the same result as is obtained from the original Markov process. A computer code is being developed to identify mergeable states by using the search algorithm.

When the code is developed it will reduce the large number of mutually exclusive states associated with a Markov process by grouping them to form a smaller set of superstates. This will render the reliability analysis of repairable systems to be performed efficiently and therefore viable from an economics standpoint.

IV. EXAMPLE USING FRANCALC COMPUTER CODE

This example is intended to illustrate some of the flexibility of FRANCALC when used to model a hypothetical Shutdown Heat Removal System (SHRS) as well as to illustrate the implementation of the recommended approach. This example, therefore, is not intended to represent any particular system.

A diagram of the system is shown in Figure 2. This figure shows that there are five paths through which decay heat can be removed using this SHRS. They are as follows:

- (1) The Direct Reactor Auxiliary Colling System (DRACS).
- (2A) Loop 1 - both sodium and steam systems in Loop 1 need to function properly.
- (2B) Loop 2 - both sodium and steam systems in Loop 2 need to function properly.
- (3A) The intermediate Reactor Auxiliary Cooling System (IRACS) in Loop 1 - the sodium systems in Loop 1 also need to function properly.
- (3B) The IRACS in Loop 2 - the sodium systems in Loop 2 also need to function properly.

In addition, electricity is supplied to the steam systems from an offsite power source while each IRACS has its own independent diesel generator. The DRACSS are assumed to operate via natural circulation and thus, do not require electric power. The loops and offsite power are constantly monitored, the IRACSS and associated diesel generators are tested once a month, and the DRACSS are not monitored. A failure probability block diagram of this SHRS is shown in Figure 3. Since loss of offsite power fails the steam systems in both Loop 1 and Loop 2, this failure must be treated as a dependency. This is indicated by the inscribed circle in the offsite power block. Failure rates and other parameters are contained in Table 1.

Figure 2. SHRS Schematic

Figure 3. SHRS Failure Probability Block Diagram

Table 1. Component Parameters

Component	Type	Shutdown Initiator	Failure Rate	Cyclic Failure Rate	Mean Repair Time (Hours)
DRACS	Nonrepairable	No	$1 \times 10^{-7}/\text{hr.}$	----	---
IRACS	Periodically Tested	No	$1 \times 10^{-5}/\text{hr.}$	----	48
Diesel Generator	Periodically Tested & Constant Unavailability	No	$1 \times 10^{-6}/\text{hr.}$	$1 \times 10^{-3}/\text{demand}$	48
Sodium Loop	Constant Monitored	Yes-240 hr.	$1 \times 10^{-5}/\text{hr.}$	----	---
Steam Loop	Constant Monitored	No	$1 \times 10^{-4}/\text{hr.}$	----	120
Offsite Power	Constant Monitored	Yes-48 hr.	$1 \times 10^{-4}/\text{hr.}$	----	---
Component	Duration of Test (hours)	Time Between Tests (hours)	Probability of a Test-Caused Failure	Probability of Detecting a Failure (test efficiency)	Probability of Overriding the test
IRACS	8	730	.01	.95	.5
Diesel Generator	8	730	.01	1.0	.5

It is assumed that a one-month (730 hours) shutdown caused by an event external to the SHRS will occur sometime during a year of reactor operation. For the SHRS to succeed, it must be capable of removing decay heat from the reactor for the entire duration of this shutdown. The reliability analysis also accounts for shutdown initiators which are defined as SHRS components whose failure lead to immediate reactor shutdown. In this example, it is assumed that loss of offsite power will initiate a two-day (48 hour) shutdown and that a sodium loop failure will initiate a 10-day (240 hour) shutdown. These outages can occur at any time throughout the one year period of operation. To succeed, the SHRS must not only be available at the time of the forced shutdown, but it must also be capable of removing decay heat for the entire duration of the outage. The frequency of these shutdowns are computed by multiplying the failure rate of the initiating component by one year (8760 hour). Thus, the frequency of loss of offsite power is $(1 \times 10^{-4}/\text{hr.}) (8760 \text{ hour}) = .876$, and the frequency of sodium loop failure is $2(1 \times 10^{-5}/\text{hr.}) (8760 \text{ hrs.}) = .175$ where the factor of 2 comes from having two sodium loops.

FRANCALC is not used to evaluate the availability of the components which initiate shutdowns. The availability of all components which initiate shutdowns is 1.0 at the commencement of the shutdown heat removal mission with the exception of the component which initiated the shutdown; its availability is zero.

The probability of SHRS failure for this example is the sum of three separate probabilities: (1) the probability of SHRS failure during the externally-caused one-month shutdown, (2) the probability of SHRS failure during a 10-day shutdown caused by sodium loop failure, and (3) the probability of SHRS failure during a 2-day shutdown caused by loss of offsite power. This is expressed by Equation (1):

$$\begin{aligned} P(\text{SHRS Failure}) = & P(\text{SHRS failure during externally - caused shutdown}) \\ & + P(\text{SHRS failure given failure of one sodium loop}) \times (\text{Frequency of sodium loop failures}) \\ & + P(\text{SHRS failure given loss of offsite power}) \times (\text{Frequency of loss of offsite power}) \end{aligned}$$

(1)

Two evaluations of system reliability were made. For the first case, it was assumed that both the IRACS and their associated diesel generators would be tested simultaneously while in the second case, it was assumed that testing of the two systems was staggered. Calculations were performed by FRANCALC. The results of these two cases are shown in Tables 2 and 3. Notice that the simple change from non-staggered to staggered test intervals has decreased the probability of SHRS failure from 2.9×10^{-6} to 4.5×10^{-7} , a factor of 6. Also, the peak probability of SHRS failure has decreased from 4.6×10^{-4} to 1.4×10^{-5} , a factor of 33. In both cases, all of the largest unavailabilities occur during testing of the IRACS. This is because the IRACS testing can be overridden when necessary only 50% of the time. Redesigning the system to allow IRACS testing to be overridden most of the time could lead to a significant reliability improvement.

The peak unavailabilities seem to increase toward the end of the year. This occurs because the DRACS is not monitored, so the probability of DRACS failure increases throughout the year. Monitoring or periodic testing of the DRACS might significantly improve system reliability. Sensitivity studies of the failure rates can also be performed, or the tradeoff in reliability from performing a shorter test which was not as efficient could be examined. FRANCALC can be used for these types of reliability sensitivity studies as well as many others.

Combining FRANTIC and PROBCALC into the FRANCALC code has provided a powerful reliability tool. It will be especially helpful in SHRS modelling as shown by the aforementioned example. Because FRANCALC essentially has all the attributes of FRANTIC and PROBCALC, it can evaluate probability block diagram models which include conditional dependencies, mutually exclusive events, and general M-out-of-N redundancy, where the system components may be nonrepairable, constantly monitored, constant unavailability, or periodically tested. The example contained herein was a small system, but FRANCALC has the capability to analyze large systems and determine instantaneous as well as average unavailability. Sensitivity studies can be performed to determine optimum testing schemes and analyze system reliability improvements. The major remaining problem is the cost of determining the reliability of a large, complex system because it is necessary to make many calculations for sensitivity studies. Current effort is being directed towards solving this problem.

Table 2. Results - Average Unavailabilities

Outage Type	Frequency	CASE 1		CASE 2	
		Probability of SHRS Failure	Total	Probability of SHRS Failure	Total
1. 30-day, Externally Caused	1	3×10^{-7}	3×10^{-7}	7.7×10^{-8}	7.7×10^{-8}
2. 10-day, Sodium Loop Failure	.175	1.6×10^{-6}	2.8×10^{-7}	1.6×10^{-6}	2.8×10^{-7}
3. 2-day, Loss of Offsite Power	.876	2.6×10^{-6}	2.3×10^{-6}	1.1×10^{-7}	9.6×10^{-8}
Total Probability of SHRS Failure per year			2.9×10^{-6}		4.5×10^{-7}

Table 3. Results - Peak Unavailabilities

		Case 1			Case 2		
Outage Type	Frequency	3 Peak Probability of SHRS Failure	Time of Occurrence (in hours)	TOTAL	3 Peak Probability of SHRS Failure	Time of Occurrence (in hours)	TOTAL
1. 30-day	1	4.1×10^{-5}	8038	4.1×10^{-5}	1.8×10^{-6}	8403	1.8×10^{-6}
		3.8×10^{-5}	7308	3.8×10^{-5}	1.7×10^{-6}	8038	1.7×10^{-6}
		3.4×10^{-5}	6578	3.4×10^{-5}	1.6×10^{-6}	7673	1.6×10^{-6}
2. 10-day	.175	3.9×10^{-5}	8038	6.8×10^{-6}	4.1×10^{-5}	8403	7.2×10^{-6}
		3.5×10^{-5}	7308	6.1×10^{-6}	3.9×10^{-5}	8038	6.8×10^{-6}
		3.2×10^{-5}	6578	5.6×10^{-6}	3.7×10^{-5}	7673	6.5×10^{-6}
3. 2-day	.876	4.7×10^{-4}	8038	4.1×10^{-4}	6.4×10^{-6}	8403	5.6×10^{-6}
		4.2×10^{-4}	7308	3.7×10^{-4}	6.0×10^{-6}	8038	5.3×10^{-6}
		3.8×10^{-4}	6578	3.3×10^{-4}	5.7×10^{-6}	7673	5.0×10^{-6}
Total peak probabilities of SHRS Failure			8038	4.6×10^{-4}		8403	1.5×10^{-5}
			7308	4.1×10^{-4}		8038	1.4×10^{-5}
			6578	3.7×10^{-4}		7673	1.3×10^{-5}

V. DEVELOPMENT NEEDS

This report has described an analytical approach for confirming the achievement of LMFBR reliability goals and some techniques that have been developed to implement this approach. There are still areas of development that are needed for a complete implementation of the approach. These areas of development relate to the following topics:

- (a) Common cause failure analysis
- (b) Markov processes
- (c) Statistical confidence (combining random variables)
- (d) Demonstration/verification of total capabilities using examples from the reactor shutdown and the shutdown heat removal system.

A brief discussion of each of these topics will be given in the following paragraphs.

The term "Common Cause Failure" is used to cover a broad spectrum of failure events, from the direct cause and effect relationships which can be defined and have been observed, to those which represent the unknown and the undiscovered. Since the unknown and the undiscovered is not yet known nor discovered, then it is most useful to treat common cause failures from a "cause and effect" point of view while continuing to learn and discover as much as possible about those potential failure events. One of the necessary ingredients of a good common cause failure analysis is the capability to properly represent, through the model and the mathematics, the interdependence among the various physical elements (man and machine) and operational and natural environments associated with the system being modelled. This capability has been developed to a large extent through the treatment of dependencies as described in this report. However, there is still more work to be done in developing the capability to treat all types of dependencies properly. The general case of a dependency, which is associated with a common cause failure analysis, is the one in which the probability of an event changes (e.g., the probability of failure of a piece of equipment) given the occurrence of another event (e.g. an earthquake). This general capability has not yet been completely developed. It is also important to

carefully define the type of data that is required in order to perform a common cause failure analysis from this point of view.

There is still development required for the efficient use of Markov analyses. As stated earlier in this report, a search algorithm has been developed for determining the mergeable states of Markov processes used for the reliability analysis of repairable systems. This, in general, will overcome the computational difficulty that is the major obstacle inherent to Markov analyses of complex power plant safety system operation. However, there is still much work yet to be done in developing a computer code to identify mergeable states by using the search algorithm. When the code is developed, it will reduce the large number of mutually exclusive states associated with a Markov process by grouping them to form a smaller set of "superstates". This will make it possible to more efficiently perform reliability analyses of repairable systems when their operations are described by Markov processes. It would then be desirable to combine this capability with that of PROBCALC to facilitate the analyses of more complex systems. Some initial thought has been given to this problem but there is still work to be done before it would be possible to combine these capabilities.

Often, the question is asked "...What confidence can be associated with the result of a reliability analysis?" The only way to answer such a question is by having performed the analysis in a manner in which the result is not a single point estimate of reliability but rather a range of values, each with an associated probability of occurrence. This, then, makes it possible to state an upper and/or lower prescribed percentile (i.e., confidence limits) for the system reliability. As described in Section III-D, work on this type of problem has progressed but is not yet complete. A subroutine has been prepared for use with PROBCALC but there are still some refinements necessary to make it work efficiently.

Finally, it is important to develop example analyses to demonstrate/verify the methods and capabilities that have been developed for reliability analyses of LMFBR systems. The Reactor Shutdown System (RSS) and the Shutdown Heat Removal System (SHRS) should be good example systems for this purpose. Each

of the system configurations is such that the various capabilities that have been developed will be required for their respective reliability analyses. Both the RSS and SHRS are relatively large and complex and each of them will have probability block diagrams that have a fairly large number of interdependencies among the elements of the diagrams. Both the accuracy and efficiency of the computer codes can be demonstrated and verified as well as the facility of using the codes.

REFERENCES

1. W. E. Vesely and F. F. Goldberg, "FRANTIC - A Computer Code for Time Dependent Unavailability Analysis", NUREG-0193, NRC-T, October, 1977.
2. J. G. Elerath and G. E. Ingram, "PROBCALC User's Manual: A Computer Program For Probability Calculations," GEFR-00408, UC-79P, August, 1980.