DATA VALIDATION AND RELIABILITY CALCULATIONS IN DIGITAL PROTECTION SYSTEMS

by

Gustavo Enrique Gonzalez-Urdaneta, Ing. Elec.

A Thesis submitted for the Degree of Doctor of Philosophy in Engineering

Department of Electrical Engineering,
Imperial College of Science and Technology,
University of London.

September 1976

ABSTRACT

A desirable characteristic in a decision-making process based on any type of measurements is that all data required had been previously validated. Data validation techniques are analysed and a new algorithm is proposed for the detection in real time of bad data in a digital protection system. A simple estimation routine based on the redundancy of the measurement takes account of missing data to form a reliable relaying data base in a high-voltage substation.

Computer methods for reliability evaluation in digital protection systems are presented. The methods are, however, general and can also be used for reliability evaluation of other systems. The minimal cut-set approach to system reliability is presented and three new algorithms are proposed for the minimal cut-set generation process. All are based on the concepts of graph theory. The algorithms enumerate the set of minimal cuts and provide system reliability bounds for nonseparable, separable and acyclic directed graphs. The node failure problem is analysed and a branch-node cut-set algorithm is proposed which allows the analysis of node outage influence on the failure rates and outage durations of each node.

The random nature of component reliability data demands computation methods to take into account the uncertainty of component parameters. Two approaches for confidence limits of the probability and frequency of system failure are presented for a repairable system represented by its minimal cut-sets. The first based on a moment method gives the variance and approximate confidence limits for both reliability indices. The second approach applies Monte Carlo simulation to obtain approximate system reliability limits at any confidence level. Both approaches are also presented for the probability of failure of non-repairable systems. A sensitivity analysis of the component reliability parameters in a digital protection system is evaluated using the repairable approach.

para

Claudia y Adriana

quienes me han dado su amor, abnegacion
y estimulo en todo momento.

ACKNOWLEDGEMENTS

This research work was conducted under the supervision of Dr. B.J. Cory, B.Sc.(Eng.), D.Sc., A.C.G.I., C.Eng., F.I.E.E., Reader in Electrical Engineering, Imperial College of Science and Technology, London, whose help and constant encouragement are gratefully acknowledged.

I am grateful to the Compañía Anónima de Administración y Fomento Eléctrico (CADAFE), Venezuela and the British Council for the financial support which made this work possible.

I would like to thank all my colleagues of the Power System Section, particularly Messrs. E. Horne, A. Ranjbar, A. Traca, S. Molina and E. Arriola who contributed directly or indirectly to this work.

Finally, I would like to thank the Electrical Engineering Department of the Imperial College of Science and Technology for the opportunity given to lecture in the Digital Protection of Power Systems, Post Experience Course in 1975 and 1976.

TABLE OF CONTENTS

CHAPTER II (Contd...) Page

CHAPTER III (Contd....) Page

CHAPTER IV (Contd...) Page

LIST OF ABBREVIATIONS AND SYMBOLS

Abbreviations

CEGB	Central Electricity Generating Board, U.K.
CIGRE	Conference Internationale des Grands Reseaux Electriques a haute tension
Tol	Tolerance
Resc	Residual counter
$\sum I$	Summation of currents
Ph R	Phase red
Ph Y	Phase yellow
Ph B	Phase blue
S/C	Samples per cycle
μs	Microseconds
ADC	Analog to digital converter
A/D	Analog to digital
c.b.	Circuit-breaker
N/O	Normally open
N/C	Normally closed
IO	Input-output
DAP	Data acquisition processor
CPP	Control and protection processor
RDB	Relaying data base
DB	Data base
COMS	Complement and swap instruction
f/y	failures per year
h	Hours
m/y	Maintenance per year
MTTF	Mean time to failure
MTBF	Mean time between failures

Symbols

${}_nC_m$	n-combinations of m-elements
$\prod_{i=1}^{n} a_i$	$(a_1 \cdot a_2 \cdot a_3 \cdots a_n)$
$\cup$	Logical union
$\cap$	Logical intersection
$\oplus$	Ring sum operation
$P_s(s,t)$	Probability of successful communication between nodes s and t
$P_f(s,t)$	Probability of failure between nodes s and t
P_{jd}	Steady-state probability of j-component being failed
P_f	Steady-state probability of system failure
f_f	Frequency of system failure: the mean number of system failures per unit time
E_i	ith event
$Pr(E_i)$	Probability of the ith event
$\overline{E}_i$	Complement of the ith event
p	Probability of success for identical components
p_i	Probability of success of the ith element
q_i	Probability of failure of the ith element
T_i	ith minimal path or tie-set
$\overline{T}_i$	Denotes failure of at least one element of T_i
C_j	jth minimal cut-set
$\overline{C}_j$	Denotes failure of all components in C_j
R_U	Upper bound of system reliability
R_{Ui}	ith upper bound of system reliability
R_L	Lower bound of system reliability
R_{Li}	ith lower bound of system reliability

Symbols (Cont'd..)

λ_j	Failure rate of jth component
μ_j	Repair rate of the jth component
$\bar{\mu}_i$	Sum of μ_j over all $j \varepsilon C_i$
$\bar{\mu}_{i+k+l}$	Sum of μ_j over all $j \varepsilon C_i \cup C_k \cup C_l$
λ_{line}	Failure rate of a transmission line
λ_{node}	Failure rate of a node or busbar
MC_i	ith minimal cut-set
MC_{ij}	ith minimal cut-set represented by a j-tuple, j = 1,2,..,NE
CNV_{ij}	ith cut-node vector represented by a j-tuple, j = 2,3,..,NN
CF_i	Cut-flag of the ith cut-set
$CI_k(i)$	Cut index of the ith cut-set when scanning node k
NK	Node-counter
FN(k)	Flag of the kth node
[k]	Set of edges incident into a node
[p]	Set of edges incident out of a node
NN	Number of nodes
NE	Number of elements
NC	Number of components
BF_{ij}	Busbar-flag of the ith cut-set represented by a j-tuple, j = 2,3,...,NN.
NMC	Number of minimal cut-sets
$\sum_{i=1}^{n} a_i$	$(a_1+a_2+a_3 \ldots+a_n)$

CHAPTER I

INTRODUCTION

Protection systems use real time system measurements for their fault detection and tripping decisions. Due to the inherent uncertainty of raw measurements, system data may not be reliable enough for immediate use because if bad data are processed incorrect or unreliable results are obtained. A significant advantage of the use of digital computers for protection functions over conventional schemes is their ability to handle discrete data and to perform some checks on the incoming crude data. A systematic treatment of the input measurements using data validation techniques enhances the reliability of the decisions taken to protect and control a power system. The results of such data-processing algorithms in a computer-operated protection system have been presented in the form of a data base having common access for all the protection strategies. Since any failure of the computer could affect one of the protections associated with every item of the substation some form of redundancy is normally envisaged for such a system. Reliability theory needs to be used for appraising each redundant configuration and a proper selection can only be made on the basis of quantitative reliability analysis. This thesis deals with the use of data validation techniques and develops reliability calculations required for computer-based protection systems.

1.1 Data validation

For power system protection there is a need to take decisions in the presence of gross errors. Data validation techniques may be used as a

framework to convert observations into reliable data in a computer-operated protection system so that decisions can be made to produce more secure operation of the protection system.

Data validation has been extensively used in communication systems by means of coding techniques for error detection and correction. Unfortunately these techniques are mainly related with transmission errors but do not provide checks for measurement errors. There is a growing literature about the bad data problem in electric power systems but this only applies to the steady state estimation problem. Similarly, considerable work has been done in the application of digital computers to protection function but relatively little has been done on the validation of relaying quantities.

The following tasks are part of the overall relaying data validation problem. First the raw measurement sets are processed to detect bad data. Once gross errors have been identified, the algorithm should replace any missing data and provide the best estimates for the highly corrupted measurements. Since the data acquired is to be useful for system protection, the speed of acquisition and validation must be compatible with the desired fault detection times.

1.2 System reliability

The reliability of the data-acquisition and data-processing system of a computer-based protection system is a very important factor since an objective of protection is to increase the reliability of the power system.

Before considering actual problems let us define the concept of reliability. In its most simple and general form, reliability is a probability of success, but an accepted definition[17] can be stated as:

"Reliability is the probability of a device performing its purpose adequately for the period of time intended under the operating conditions encountered"

Criteria of what is considered an "adequate performance" and how to measure it, have to be exactly specified in each case. For example, a computer-operated protection system must operate in the presence of a fault but it must not produce false operations. As long as the system performs these functions, the scheme is judged as operating satisfactorily. Measurement of adequate performance requires calculation of the probability of failure to operate and the probability of unnecessary operations. K. Conelly et al[73] refer to the former as the "system unreliability" and to the latter as the "system security" against false operations. To judge adequate performance also involves the determination of the frequency of failures and malfunctions.

Failures of digital systems are the consequences of faults which arise as a result of a step-type change in the performance of one or several of the components of the system. These failures can be properly modelled by random variables governed in many cases by an exponential law[48],[67].

The reliability calculations of even simple devices are frequently quite complicated because normally a number of components must operate properly to perform the system function. In the literature there are methods that could be used to calculate the exact reliability[18],[26-27],[34] but unfortunately they involve excessively voluminous computations for complex systems. This situation has prompted considerable research in approximate methods[16],[18],[30-33] for the reliability evaluation of complex systems from known reliability indices of constituent elements. Moreover a deterministic appraisal of the performance of

the system fails in many cases to provide a true picture of the system reliability because of the uncertainties associated with component parameters. Statistical approaches[56-60] are required to construct confidence from which the goodness of the performance can be assessed.

1.3 Contents of this thesis

This thesis deals with the development and testing of computer methods suitable for the on-line relaying data validation and for the reliability assessment of computer-based protection systems.

Chapter II covers the area of relaying data validation. Protection statistics are analysed and the incidence of bad data points in fault-clearance equipment is discussed. The relaying data validation problem is described and a new algorithm is proposed and tested for the detection and identification of loss of data and large unexpected errors. Special consideration is given to the identification of zero data. An estimation procedure is presented which provides replacements for missing data or a best value for doubtful measurements. The feasibility of using both algorithms in one corner of a typical 400 kV mesh substation on the CEGB's network is reported. On-line results are given of the data validation algorithm tests on a laboratory substation-computer model system.

Chapter III describes and analyses possible methods for the evaluation of the terminal-pair reliability. Two of the four main lines of attack (tie-set and cut-set) are followed because of the computational advantages of these approaches for complex systems. Consideration is given to the effect of term simplification and ordering of the partial reliability table constructed from the system success paths. A set of

rules are formulated for the selection of the evaluation method in complex systems with unreliable branches and nodes. Three different approximations and bounds from the tie-set and cut-set approaches are numerically compared in both lower and upper reliability regions.

Chapter IV deals with the minimal cut-set enumeration problem. Since much of the work reported is based on graph theory, important concepts and definitions used are summarised. A description of system modelling by reliability graphs is included and the concept of cut-node incidence matrix is introduced. Three new algorithms are presented for separable, nonseparable and acyclic directed graphs. A simple sparsity technique is proposed for the efficient storage of the minimal cut-set and cut-node incidence matrices. Several graphs were used to test the algorithms and computation times are provided in each case. The node failure problem is analysed and reference is made to the incidence of busbar faults on the reliability indices of each node in a power system. A new branch-node cut-set algorithm is presented and the results of tests on three systems reported.

Chapter V presents two different approaches for the variances and confidence limits of the reliability indices derived from the minimal cut-set approach. The first technique involves the orthodox statistical procedure of characterising distributions by their low order moments and uses Chebyshev's inequality to provide probability bounds for the indices. The second approach applies Monte Carlo simulation to generate numerical distributions from where confidence limits are computed. An extensive analysis is made of the effect of component availability, parameter variances and sample size of component tests on the performance of both methods.

Chapter VI treats the problem of sensitivity analysis in reliability evaluations. A new approach is presented which calculates sensitivity coefficients of each reliability index given by the minimal cut-set approach with respect to each component reliability parameter of the system. General considerations used for the reliability evaluation of computer-based protection systems are described. An application of the algorithms presented in this thesis on two particular protection system configurations is reported.

Chapter VII discusses the conclusions reached and suggests possible future developments.

After Chapter VII there are two appendices. Appendix 1 includes the edge-listing of all graphs used in the tests of the algorithms proposed in this thesis. Appendix 2 presents a complete discussion and the proofs developed by the author of his minimal cut-set algorithms described in chapter IV.

The contributions offered by this thesis are the following:

i. the development and testing of a data validation algorithm to carry out validation routines during the fault period (chapter II).

ii. an estimation procedure is proposed to form a reliable data base in high voltage substations (chapter II).

iii. The development and testing of three new algorithms based on graph theory for the minimal cut-set enumeration of separable, nonseparable and acyclic directed graphs (chapter IV, appendix 2).

iv. the design and testing of a branch-node cut-set algorithm particularly suitable for the evaluation of the incidence of busbar faults on the reliability of distribution and transmission systems (chapter IV).

v. the development and testing of a new formulation for the variance and confidence limits of the probability and frequency of system failure (chapter V).

vi. the development of sensitivity coefficients for the location of redundant components to improve system reliability based on the minimal cut-set approach (chapter VI).

CHAPTER II

RELAYING DATA VALIDATION AND ESTIMATION

2.1 Introduction

A computer-based relaying system uses measurements taken in real time to perform the relaying calculations to produce decisions for the different protection strategies. On-line measurements are usually corrupted by some degree of error or can be missed by malfunctioning of the data acquisition system. If some measurement has a large unexpected error or simply becomes not available, the relaying calculations might produce an unwanted operation or the system might become inoperative in the presence of a fault. The risks derived from these possibilities justify the need to use a data validation technique for the detection and identification of bad data points and a process of estimation to replace missing or erroneous data.

This chapter deals with the software problem of validation and estimation of the relaying quantities which form the relaying data base from which the protection strategies can draw the appropriate data. The possibility of using such techniques in one corner of a typical 400 kV mesh substation on the CEGB's network is also analysed.

2.2 Review of the literature

In spite of all the recent interest in this field, very little attention has been paid to the validation and estimation of relaying quantities for real time application in a computer-based relaying system.

The Swedish municipal power distribution system[1] and the Hamburg-West Germany system[2] have reported the use of coding techniques for data validation. The Swedish safeguarding of data is in the form of a combination of parity bit checks, bit length error detection and code restriction checks. The latter is used for validation of data such as switch status and command operations which are generated as defined combinations of bits, other combinations are detected as errors. In the Hamburg system the majority of measured values, indications and commands, are transmitted digitally by an electronic supervisory control system using pulse duration modulation. Commands and indications are transmitted independently of each other and protected by a special code. The decoding of the pulse trains is performed by hardware as well as by computer program. The limitation of these techniques is that it only endeavours to maintain the original data unchanged but it is powerless if the data is wrong in the first place.

Couch[3] developed a configuration analysis technique which he applied, among other, for data validation. He uses the substation internal configuration and Kirchhoff's current and voltage laws to establish zero differential current zones and equal voltage zones. If, under non-fault conditions, for any such zero differential current zone, the differential current is non-zero, then there is an inconsistency. Similarly, if in any such equal voltage zone, the measured voltages are not equal, then there is an inconsistency in the information. In both cases the inconsistency can be due to a measurement error and/or error in switch position indication. The analysis of inconsistencies once detected is less straightforward and no

systematic procedure is proposed for a real time application. Also the main type of measurement errors assumed are major ones such as those due to loss of secondary potential or current.

Another data checking procedure is included in the programs at the CEGB new national control centre[4-6]. Raw data of circuit power flow and switch positions of the complete network are taken at 1 minute intervals and organised into tables to associate them with appropriate nodes and circuits. The checking procedure begins by taking two consecutive samples which are compared to see if they are consistent within a margin of 5% of actual flow or 20 MW if flow is small. If they are, the program continues with the latest of the 1 minute samples of complete data, otherwise one sample is retrieved from the computer backing store and a decision is made on a 2 out of 3 basis. The switch indication check uses two independent pieces of data telemetered to indicate switch position. A (1,0) combination indicates the open state and a (0,1) the closed state. Any other combinations are doubtful and require checking. With a doubtful switch position, the method is to assume it to be closed if current flows in the adjacent line and assume it to be open if no current flows. Alternatively a decision table constructed with the three pieces of data available can be used to make a decision on a 2 out of 3 basis. Such techniques are used to provide information for secure operation of the system.

Horne and Cory[7] propose a data validation method aimed at fulfilling its task in an interval of 625 μs i.e. sampling rate of 32 samples/power frequency cycle. The switch position check uses also two pieces of data which are logic complement of each other. If there is any inconsistency between them the switch data is given

a logic 1 to indicate an error. No other check is made during the validation to correct a doubtful switch indication. For non-zero analogue data, the status of the neighbouring switch(es) is examined. If closed, the data is assumed to be consistent, otherwise the data is tagged as doubtful. After these checks in the data acquisition processor, all data is transferred to the control and protection processor (CPP). It is the job of the CPP to choose either to disregard tagged data or to replace it by correct data calculated from other measurements. The method proposed does not have the capability of detecting and identifying a missing or erroneous data and does not propose any corrective action to replace a doubtful data, missing or erroneous.

The research work reported in this chapter deals with the software problem of data validation during the time interval available between successive samples and specifically, the error detection and identification problem.

2.3 Protection statistics

The probability that a protection system might fail to operate or might produce unwanted operation in the presence of bad data can be found from protection statistics available from real systems. The results of a 1974 survey of failures in fault-clearance equipments(8) are summarised in table I. Only the figures representing failure to operate and unwanted operation are presented. The results from a previous 1958 CIGRE survey(9) are also included in this table.

The largest single cause of failure in both surveys is attributed to relay malfunction. Both reports indicate that in many instances relay failures result from wrong application or incorrect

TABLE I

PROTECTION STATISTICS

Cause of incident	1974		1958 (%)
	Failure to operate (%)	Unwanted operation (%)	
Relay failure	43.2	48.5	43.0
Circuit-breaker mechanism	9.0	3.6	7.0
Circuit-breaker failed to interrupt	7.0	0.2	13.5
a.c. wiring	5.2	4.5	12.0
Current transformer	5.2	3.0	7.0
d.c. supply	5.2	0.2	1.0
Trip wiring	4.2	2.4	5.0
Voltage transformer	4.0	0.9	3.0
Auxiliary switch	2.8	-	3.0
Circuit-breaker trip coil	2.5	0.4	2.5
Multicore cable	1.4	0.4	-
Miscellaneous or unknown	10.3	35.9	3.0

setting. The use of a computer-based relaying system would effectively eliminate problems related to these effects. Unfortunately it is not possible, from this report, to obtain a figure for the availability of the relays because they have not been analysed for all relay trip operations i.e. the figures for correct and desired operations are not reported. However, from previous utility reports[10] it has been found that conventional relays have a typical availability of 97.5% when evaluated on situations where relay action should have taken place or which took place inadvertently. By comparison, digital computer availability in excess of 99% can be reasonably obtained. This availability is based on overall operation and not just on reliability when certain events occur; thus computer implemented relaying should result in more dependable relay functions and should reduce the number of failures attributed to the protection system.

The next highest number of failures is for those resulting from some form of circuit-breaker malfunction. However, if we assume that failures in a.c. wiring, current transformers and voltage transformers produced that the measured quantities were not available or were wrong, then such conditions can be considered as bad data. The figures attributed to such causes are comparable or higher than the overall percentage of circuit-breaker failures for both types of incidents. If one considers valid the previous assumptions, the actual number of failures that can be attributed to the presence of bad data demand the use of data validation and estimation routines in a computer-based relaying system. Protection statistics of the CEGB network[11] also indicate that the second highest maloperation rate of distance protection was due to loss of voltage transformer supply for both types of relays, electromagnetic and static.

2.4 Relaying data validation

The relaying data validation problem can be described as the detection in real time of any bad data, missing or erroneous, which may lead to possible failure, misoperation or deterioration of the protective relaying system.

Its solution requires the use of a data processing algorithm which fits the requirements of the fault clearance time desired and form a relaying data base from which all the protection strategies can draw appropriate data.

2.4.1 Information from the system

The information from the system can be divided for the purpose of validation into:

(i) analogue data corresponding to system current and voltage information which are derived from the output of the current and voltage transformer.

(ii) digital data representing the information of the switch status which is usually obtained by monitoring auxiliary contacts of each switch.

In a computer-based relaying system within the substation environment, current and voltage waveforms are sampled at all selected points in the substation and are then input, together with the switch status information, into the processor which is responsible for data acquisition and validation. The processed data is assembled into the relaying data base from which all the protection strategies can draw the appropriate data. Fig. 1 illustrates the complete process.

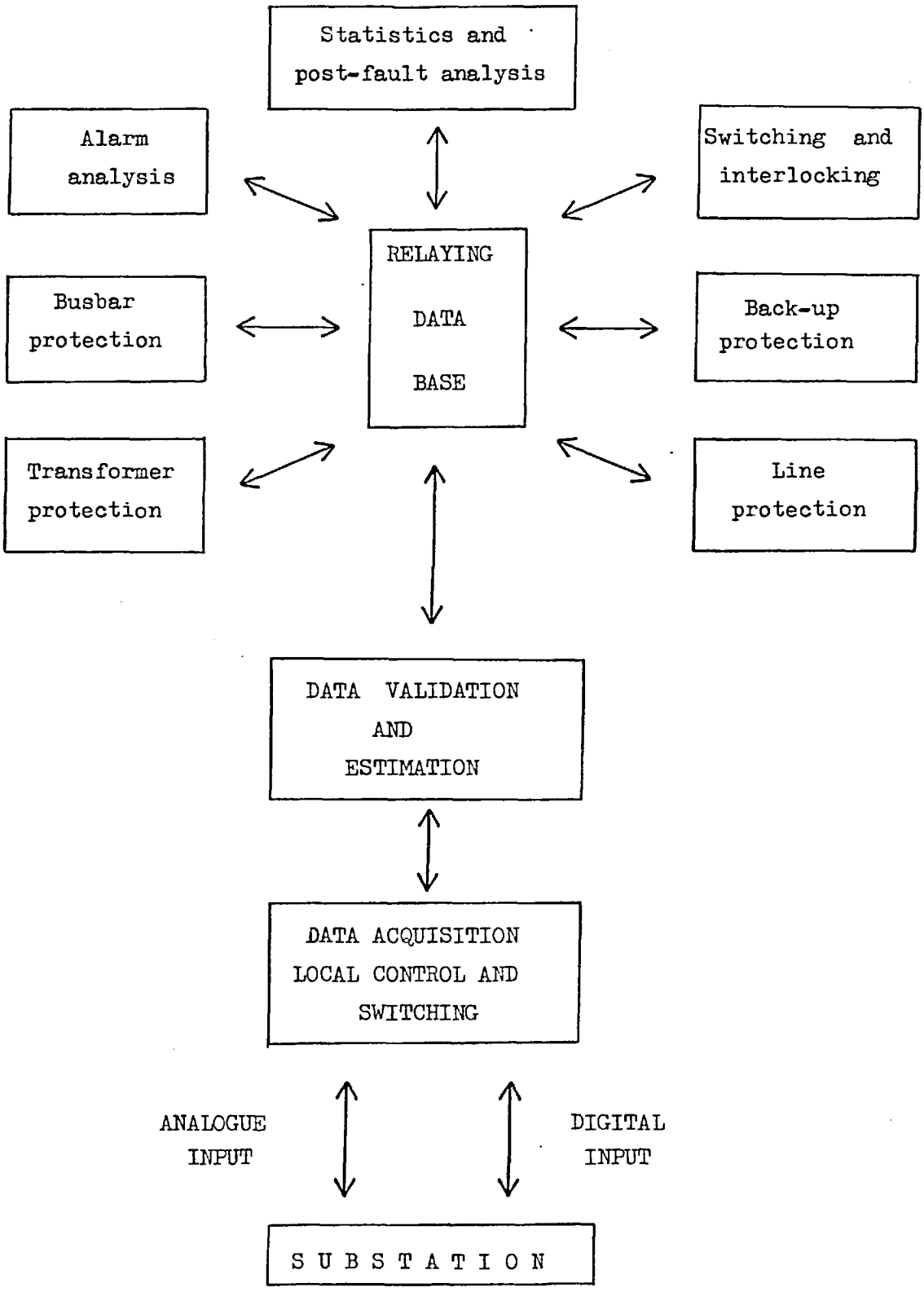


Figure 1. Block diagram of a computer-based relaying scheme

2.4.2 Types of error

Fig. 1 shows that there are two main types of error in the input information to the data validation process namely:

(i) measurement error

(ii) switch indication error

Type (ii) errors arise normally from failure in the auxiliary contacts of the switch. Type (i) errors are due to the fact that no physical measurement is perfect being usually corrupted by some degree of error. However, the measurement errors of concern in the relaying data validation problem are not the normal random errors but gross errors due to failures in the data acquisition link or in the voltage and current transducers.

To solve the bad data problem the relaying data validation process must be able to identify two possible measurement errors that might produce the worst effects from a protection point of view namely:

(i) loss of data that might result in failure to operate in the presence of a fault

(ii) large unexpected errors that might degrade the relaying computations and eventually produce unwanted operations

2.4.3 Problem formulation

Given the set of analogue and digital data from the substation the critical functions during the validation process are

(i) detection and identification of bad data

(ii) estimation of the most probable value for each single quantity in the data base

The main constraint in the design of a relaying data validation algorithm is the short time available between samples for validation assuming that the data base must be updated after each sample cycle. For instance, if the data is sampled at a rate of 32 samples per cycle corresponding to a sampling interval of 625 μs for a system of 50 Hz then during this time the processor must read in system data, perform data validation and assemble the data base.

Considering this time restriction, the design of a relaying data validation algorithm to fulfil the previous functions must be based on the following principles:

(i) the algorithm must be simple so that the speed of acquisition and validation will be compatible with the desired fault clearance times.

(ii) the method must be flexible enough to cope with any changes with the minimum of modifications

(iii) the data base must be reliable i.e. the presence of gross errors must be detected, identified and each single quantity presented with the most probable value.

2.5 Data validation and estimation procedure

There is normally considerable redundancy in the information available in a substation which aids malfunction detection. In a computer-based relaying system there must be special concern for the reliability of the information because it is used to automatically control elements of the substation. The data required must be previously validated.

The checks proposed in this section are to supplement the techniques proposed by Horne and Cory[7], but are more general in that they supply means to detect and identify gross measurement errors and an estimation procedure is proposed to provide replacements for missing or erroneous data.

2.5.1 Digital data validation

Normally in a computer-based relaying system the information on the open and close status of each switch will be read as two bits. These two bits reflect the status of a pair of contacts operated by the opening and closure of the switch.

These two bits are logic complement to each other and the validation consists of checking if both pieces of data are complements of each other. If there is an inconsistency the switch-data is given a logic 1 as an error flag to indicate a doubtful switch indication[7]. The corrective action for error in the switch position is taken during the validation of the analogue data using a sample-switch routine. Fig. 2 shows a flow diagram for check of switch-status information.

Whatever the structure of the switch-data word, the previous check requires one bit to be used as error flag.

2.5.2 Analogue data validation[12]

A zero data can be associated with any of the following conditions:

(i) the waveform can be passing through zero

(ii) the circuit is de-energized

(iii) a fault in the data acquisition link

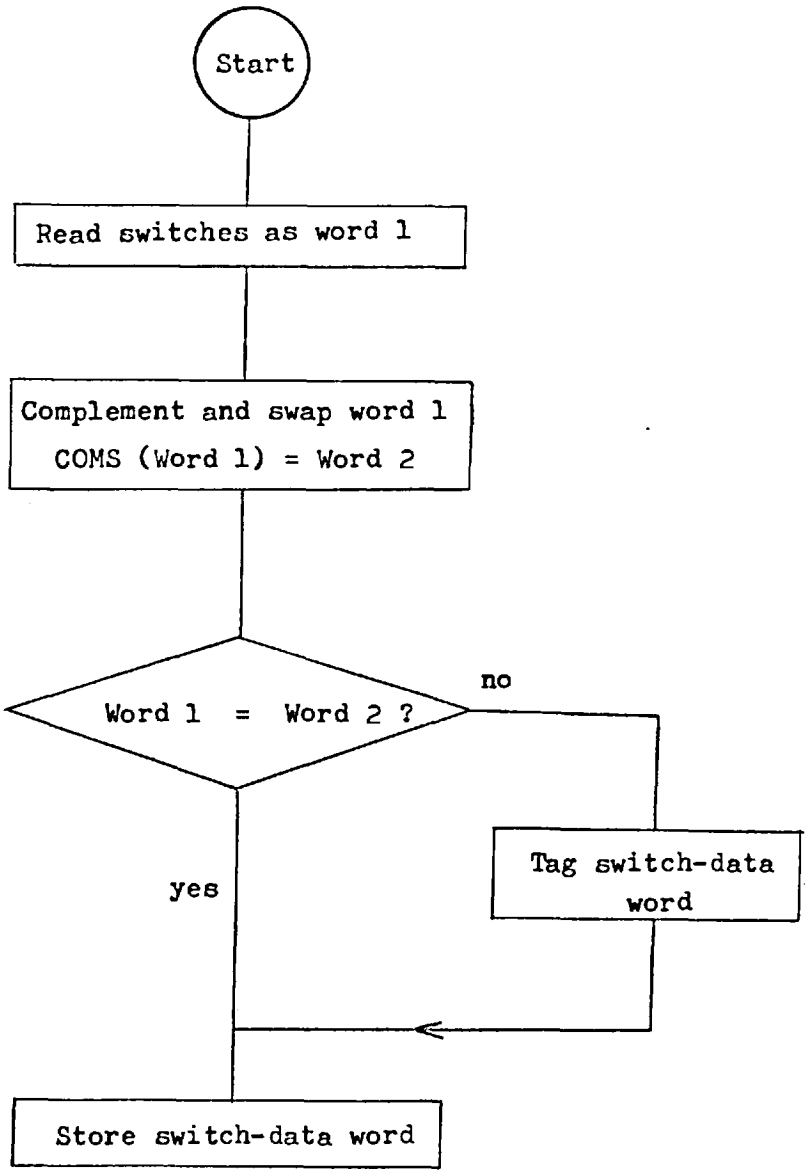


Figure 2. Flow diagram for check of switch-status.

The third condition affects the reliability of the data base, therefore the validation process must provide the steps to discriminate between these three conditions.

Zero data recognition by software will generally take more time than that available for all the validation process due to the conversion tolerance of the A/D converter. For instance in a NOVA minicomputer it takes 14 μs/phase. To avoid this time-consuming check the zero data identification, within the tolerance of the A/D converter, must be done by hardware and a zero data can be given a logic 1 as a zero flag. This check by hardware takes just a few nanoseconds for each three-phase point.

With zero data recognition by hardware the analogue data validation checks the zero flag in each phase and this check over all three phases enables the identification that zero data in just one phase is due to the sample being taken when the waveform passes through zero.

If all three phases have been tagged as zero by hardware, a positive indication that the adjacent switch(es) is open is used to indicate that the circuit is de-energised; otherwise the data is given another logic 1 as error flag to indicate doubtful data. The missing data will be replaced or estimated during the estimation process.

A non-zero value in any phase of a current transformer is an indication that the circuit is energized and therefore used for validation that the adjacent switch is closed. This sample-switch routine together with the previous validation of the switch by the two complementary bits provides a 2 out of 3 decision.

The presence of a large unexpected error or the loss of data in one or two phases is detected by comparison of the residual value against a tolerance, only if the tolerance is not exceeded the data will be accepted as valid, otherwise it is given a logic 1 to indicate doubtful data which needs to be replaced or estimated. This tolerance will also be exceeded during any system abnormality and therefore a residual counter is used to discriminate between transducer error and a system abnormality. The latter will be reflected in all the active monitoring point.

The checks mentioned for analogue data require that the data-word uses two bits as zero and error flags. A detailed flow diagram of the algorithmic procedure for digital and analogue data validation is shown in Fig. 3.

2.5.3 Relaying data base estimation

The estimation concept here is used to describe a process which can provide replacements for missing or erroneous data. Such a process can only be effective if there is some redundancy in the information. In the substation environment this redundancy must be provided either by additional measurements (type I) or using the relationship implicit by the Kirchhoff's current law (type II). The latter is an inherent redundancy in electrical networks and the former is very often present when two different types of protection are applied to one item of plant, e.g. first and second main line protection. The estimation process should use any type of redundancy available to provide the best replacement for missing or erroneous data or the best estimate for clean data.

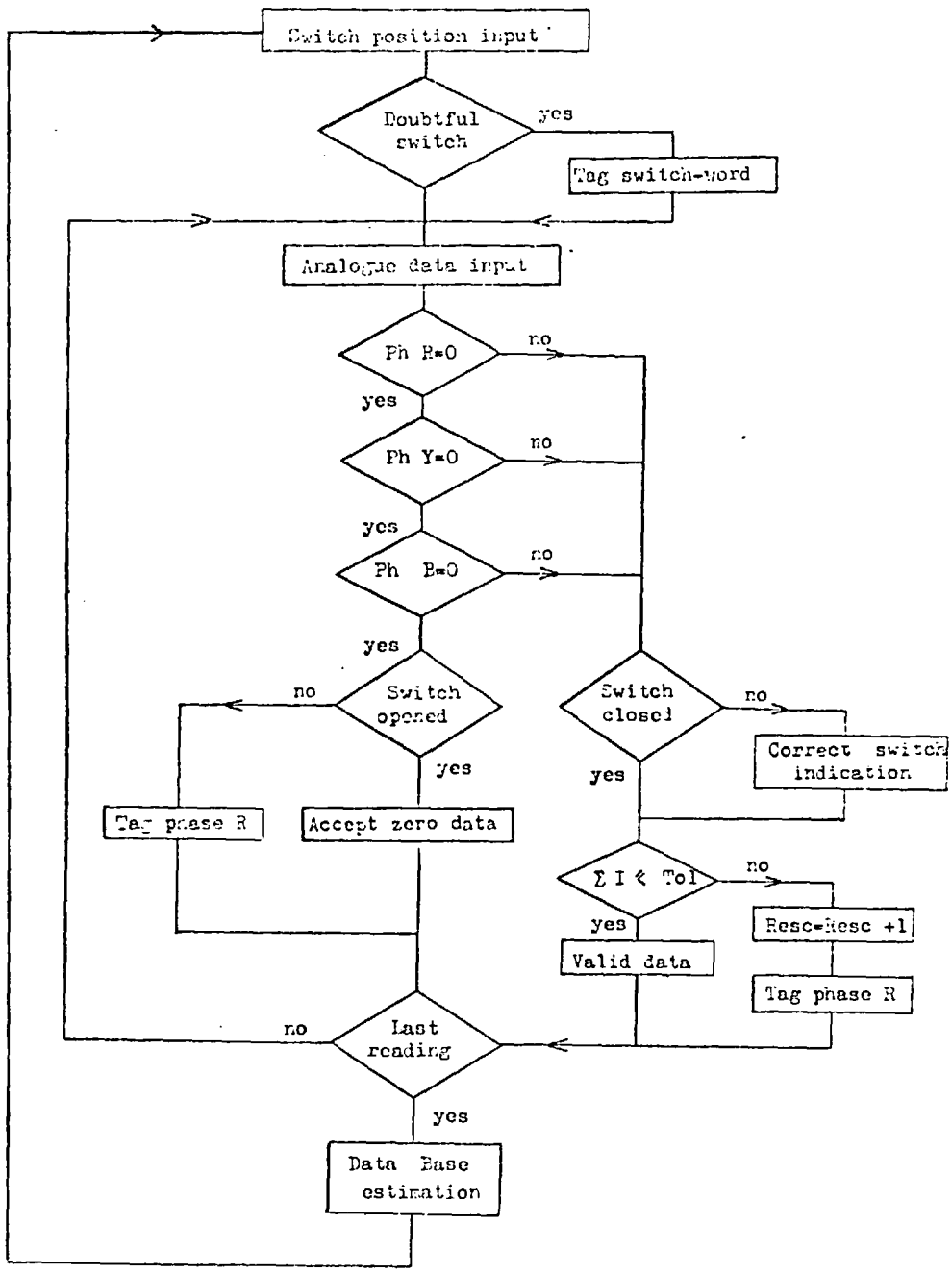


Figure 3. Flow diagram for data validation

The relaying data base (RDB) must be constructed considering the specific tasks to be performed by the protection strategies. There is no need to include any redundancy for a single quantity if this has been previously validated e.g. the line protection routines must have available three-phase line current and voltage.

Once the presence of a missing or doubtful data has been detected and identified the estimation procedure for each single quantity of the RDB is based on the following principles[12]:

(i) a zero data with error flag clear is transferred directly to the RDB

(ii) a zero data with error flag set is estimated from any type of redundancy available

(iii) a non-zero value with error flag clear is transferred directly to the RDB if at least one of the correlated measurements has an error tag, otherwise the redundancy is used to estimate the best value

(iv) for the case of a non-zero data with an error tag, the information from the residual counter is used to discriminate between a transducer error and a system abnormality. In both cases an estimation is done for that value but in the first case the value being checked is not used in the estimation unless the redundancy available has the error flag set.

The estimation routine for each quantity to be entered in the RDB can be done considering two types of measurement-redundancy relationship:

I) Measurement with single redundancy : The estimate will be given by

1. any one of the measurements which has a clear error flag if the other is doubtful (missing or erroneous) and the residual counter limit has not been exceeded.

2. the average value, if both measurements have clear or set error flags and the residual counter limit has not been exceeded.

3. the average value, if the residual counter limit has been exceeded except in case 4.

4. the non-zero measurement if the other is zero and the residual counter is exceeded.

II) Measurement with double redundancy : The estimate will be given by

1. any of the measurements which has a clear error flag if the other is doubtful (missing or erroneous) and the residual counter limit has not been exceeded.

2. the average value of the measurement and the series redundancy (type I) if the residual counter limit has been exceeded or both data have clear error flags.

3. the redundancy type II if the other two data are doubtful (missing or erroneous) and the residual counter limit has not been exceeded.

4. the non-zero measurement between the measurement and the series redundancy if the other is zero and the residual counter limit is exceeded.

Table II summarises the possible cases considering all combinations of the zero and error flags between the measurement (first measurement)

TABLE II

Estimate values using zero and error flags information

Case	First measurement		Second measurement		Estimate Values			
					Single redundancy		Double redundancy	
	ZF	EF	ZF	EF	RC $\leqslant$ lim	RC > lim	RC $\leqslant$ lim	RC > lim
1	0	0	0	0	2	3	2	2
			0	1	1	3	1	2
			1	0	1	4	1	4
			1	1	1	4	1	4
2	0	1	0	0	1	3	1	2
			0	1	2	3	3	2
			1	0	1	4	3	4
			1	1	2	4	3	4
3	1	1	0	0	1	4	1	4
			0	1	3	4	3	4
			1	0	1	3	1	2
			1	1	2	3	3	2

and its redundancy (second measurement). For the case of measurements with only redundancy type II, the flags from all contributions must be equal to allow estimation, otherwise the first measurement is preferred except in case 3 of table II where the estimation is based on the flags of a single contributor. The case when the measurement has been identified as a real zero data (ZF = 1 and EF = 0) is not considered because a zero data without error flag is transferred directly to the RDB. The solutions presented in table II provide estimates more conservative than otherwise and are intended to use the flag information provided by the validation algorithm of sections (2.5.1) and (2.5.2) to provide replacements for a doubtful data (missing or erroneous).

2.6 Substation-computer model system

For research and development purposes[7],[13], the requirements for data acquisition and control of a typical 400 kV mesh substation on the CEGB's network has been chosen. As indicated in Fig. 4(a), one corner of such a station has been modelled by setting up 220 V, three-phase circuits in the laboratory in which the circuit breakers and isolators have been simulated by three-phase contactors. These are slugged electronically to provide realistic operating times and the control circuitry enables them to be operated from a digital processor. In the mesh corner, full instrumentation is applied by means of voltage and current transducers, but in the remainder of the substation only the status of breakers and isolators is modelled by single phase relays so that the station configuration can be determined.

The digital system connected to the model is shown in Fig. 4(b) and consists of two processors having a common data base (DB). The

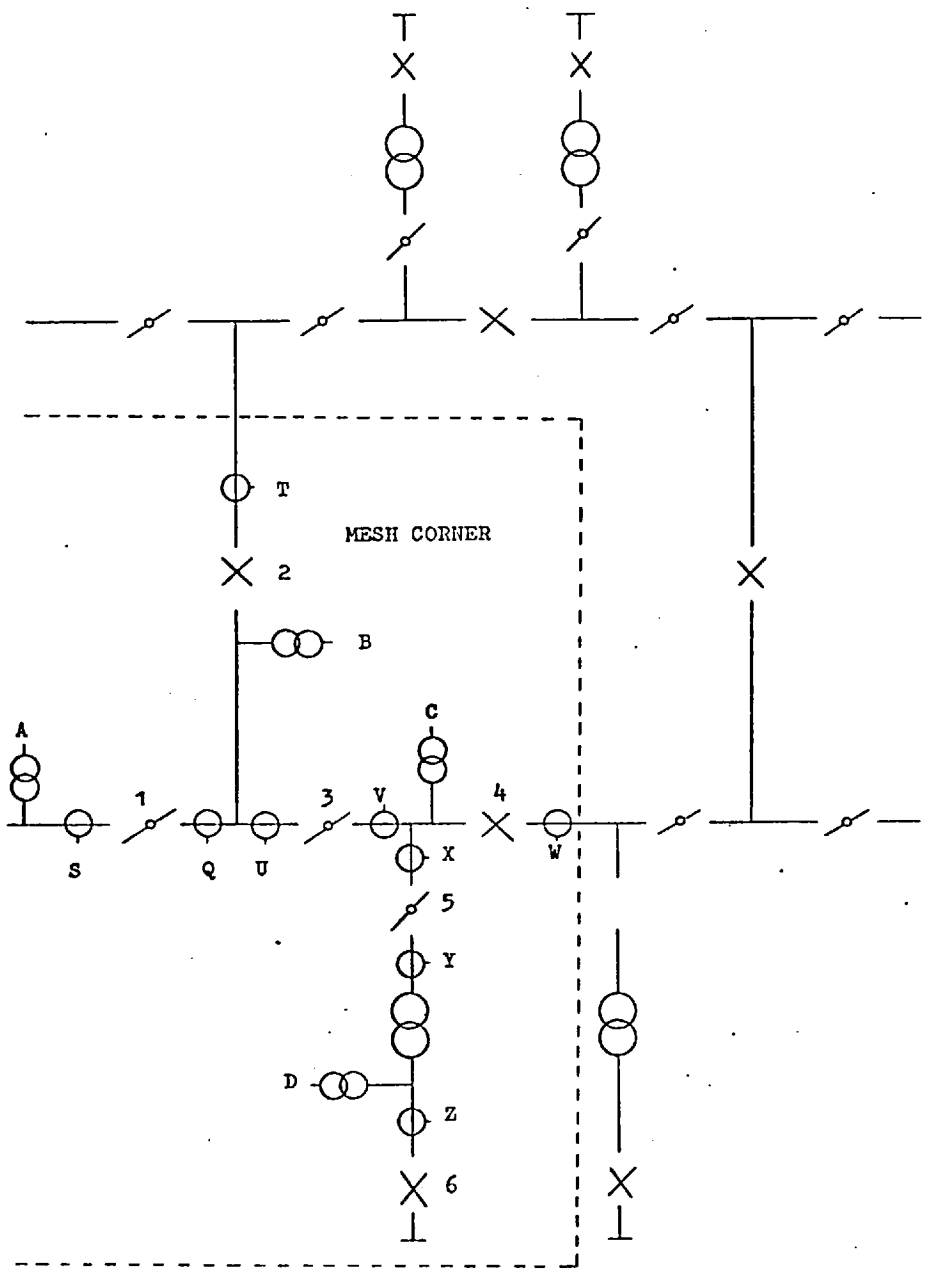


Figure 4(a). Mesh substation

data acquisition processor (DAP) is used to scan both analogue (transducer) signals and digital (c.b. and isolator position) signals from the model, to make validation checks, estimate each single quantity of the relaying quantities and to send the data to the DB. The control and protection processor (CPP) is responsible for running all the algorithms which determine the state of the substation and surrounding system, to switch it to eliminate faults or from a manual command from the local control room.

In this section only the hardware and software which are relevant to the data validation and estimation process in the one corner substation are described. Computer requirements in terms of memory and execution time are presented.

2.6.1 Data acquisition processor

In the substation-computer model system (Fig. 4(b)) the DAP is a NOVA 1210 mini-computer having a 12 k word core store, a 1.20 μs memory cycle time, 16-bit length word and 4 programmable accumulators. The arithmetic instructions, with an execution time of 1.35 μs, operate on fixed point binary numbers either unsigned or the equivalent signed numbers using two's complement conventions.

2.6.2 Data system hardware[7],[13],[14]

The interface provides 4-16 bit digital word inputs for switch-status information and 48 multiplexed analogue inputs for current and voltage measurement.

Each switch position is read as two bits which reflect the status of a pair of auxiliary contacts operated by the opening and closing of the switch. This data is packed into 16 bit words each

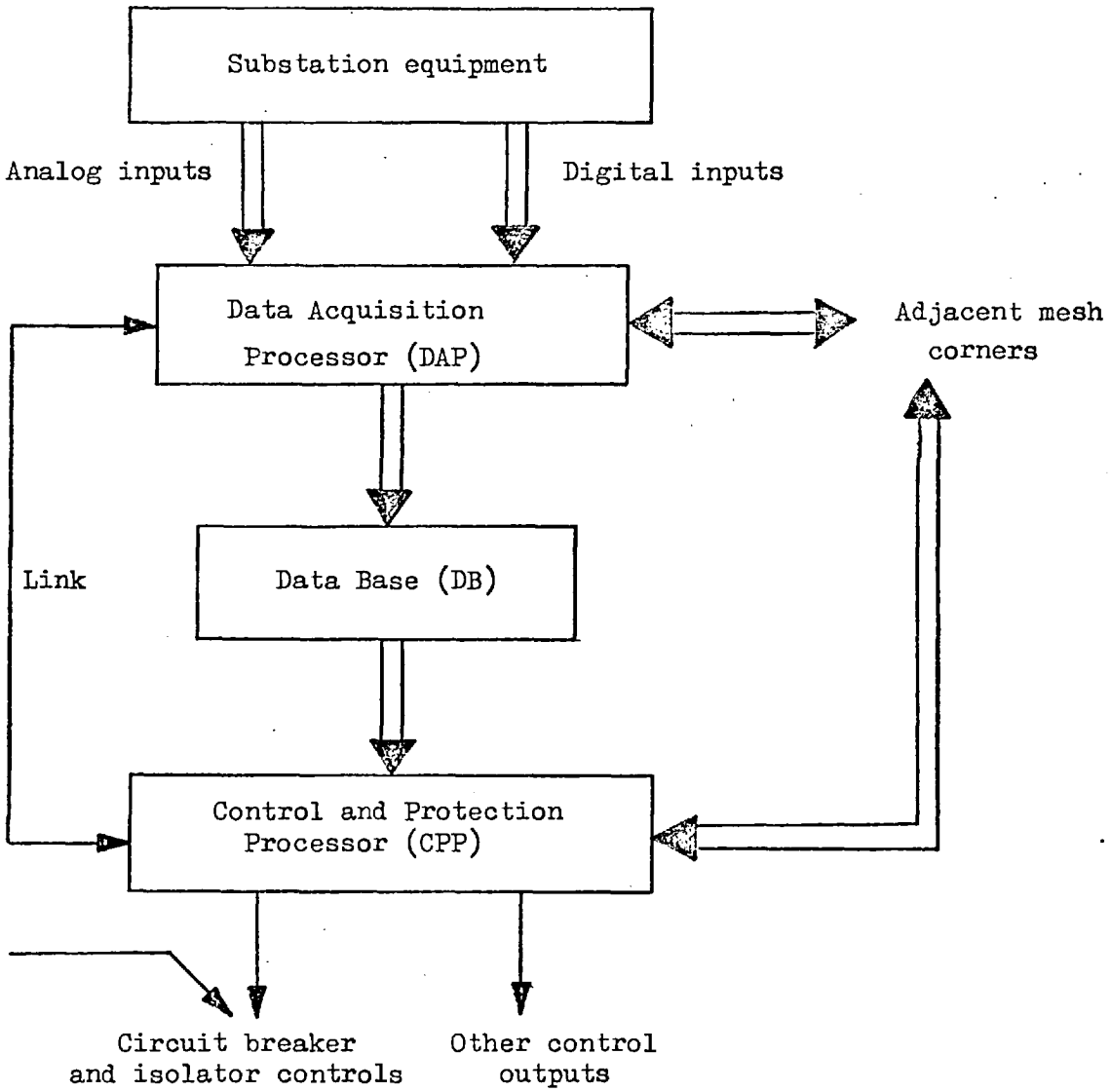


Figure 4(b). Proposed hardware configuration for mesh corner protection and control [13] .

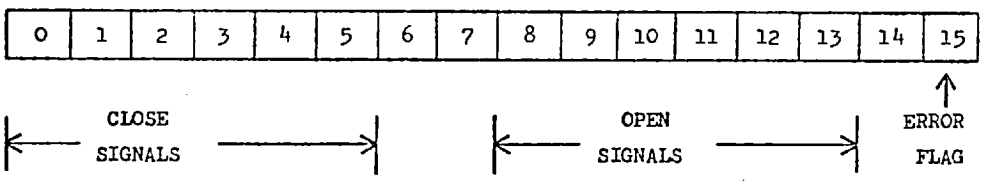
containing 6 switch positions (12 bits) and 4 spare bits which can be used as error flags. Fig. 5(a) presents the switch-data word format.

Analogue signals which are monitored by the computer are currents and voltages. The current signals are measured by air cored current transformers (linear couplers). The sensitivity of the linear couplers is about 6 mv pp/amp r.m.s. The voltage, measured between phase and neutral, is obtained via a small step down transformer of 240/6 V r.m.s.

Each analogue output from the model is connected to a sample/hold circuit controlled by a single phase locked clock synchronised to the 50 Hz supply. A clock pulse causes all sample/hold circuits to sample simultaneously with a window of 1 μs. The sampling rate can be varied by computer control at 32, 24, 16, 12, 8, 6, 4, and 2 samples per cycle and an integrator is included in the sample/hold circuit to give a measure of filtering.

The analogue/digital conversion is performed by 3-10 bit, 25 μs converters within the interface, thus enabling all 48 inputs to be converted and input in less than 525 μs. For program control purposes the converters may be considered as a single peripheral device but for data input purposes each one is treated separately and 3 input instructions are required. For validation purposes a data "zero within tolerance" flag is available from each converter and is input with the converted data to bit 0 of the chosen accumulator. The flag is a logic 1 for the zero data condition. The tolerance limit is hardwired on each converter card and may be altered as required. Initial tolerance is preset to $\pm$ 2 least significant bits of the converted word. Fig. 5(b) presents the analogue-data format.

(a) SWITCH - DATA WORD



(b) ANALOGUE - DATA WORD

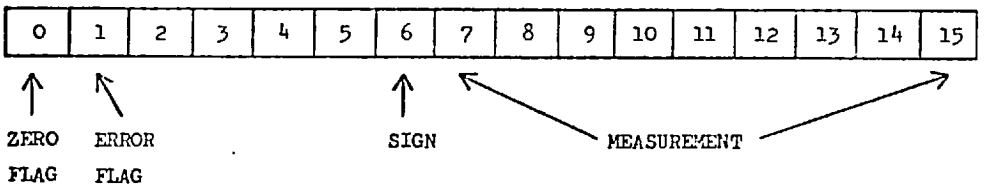


Figure 5. Data-word formats

Each of the analogue/digital converters is served by a 16 way analogue input multiplexer which may be operated either by program control or in an automatic mode slaved to the A/D converters.

2.6.3 Intermediate storage

The DAP is linked to the CPP (a PDP 15 with 16 k store in this case) via a first-in, first-out (FIFO) buffer. Data output to this link is under NOVA program control and on completion of a data block the link generates a data channel request to the CPP and the block is transferred to the DB which is a portion of the PDP 15 core store. The DB stores two complete 50 Hz cycles of data before being overwritten.

2.7 Software system

During the time interval available between samples e.g. 625 μs for a sampling rate of 32 s/c, the DAP is responsible for reading in system data (i.e. analogue to digital conversion, multiplexing and read in), performing the validation checks to identify doubtful data, labelling and storing it, performing a simple estimation and sending the data to the CPP.

With the NOVA, input and output of a single quantity takes about 2.55 μs and 3.15 μs respectively. In the model of Fig. 4(a) the total data movement time for the 4 switch-data words and 39 data words for current and voltage takes 245.10 μs. Also the analogue/digital conversion of each three-phase quantity takes 25 μs i.e. a total conversion time of 325 μs for all 13 three-phase monitored points.

So, besides the interface, specifically designed to minimise data acquisition overheads, it is also necessary that the validation be structured to optimise the overlap between input, conversion and validation procedures.

2.7.1 Programming considerations

The data acquisition cycle starts upon receiving a pulse from the synchronised clock which initiates a sample/hold operation and informs the computer to start an acquisition cycle.

After considering various alternatives it was decided that the validation process should read in the data and then, while the data is still in the accumulator, perform validation before storage. This process is quite attractive to the NOVA architecture which has four programmable accumulators. Also the validation checks (sections 2.5.1 and 2.5.2) were designed to treat each monitoring point individually without bothering too much about correlated measurements through the substation switch configuration. Zero data recognition by hardware was found essential to maximise the time available for validation between samples. Priority interruption is deliberately not used because once an interrupt is requested on completion of conversion, data needs to be read in and stored. Moreover, interrupt handling, restoration of the interrupted program, reading and storing of the data are very time-consuming. If the processor can read data in any time at its "convenience", it can wait until validation on the current data has finished and stored before reading in another set of data. While the processor is validating a particular word, the ADC is doing its conversion for the next data and is timed such that when the processor has finished with the present data, the ADC has also

completed its conversion. This arrangement allows conversion and validation to be carried out simultaneously and the processor is not idling during conversion periods.

The sequence of events at the start of data acquisition is shown in Fig. 6 and the corresponding flow diagram in Fig. 7. However, testing for completion the conversion takes too much processing time so the program is constructed to repeat the section of the analogue data validation for all the monitoring points. This requires increase core storage but the program is simple and storage is not a critical problem. So, once started the validation procedure continues until all monitoring points have been validated and stored in labelled locations for data retrieval by the estimation routine. Once each single quantity of the RDB has been estimated and sent to the CPP, the DAP waits to receive the next pulse from the synchronized clock.

The validation checks of sections (2.5.1) and (2.5.2) demand data manipulation at a "bit-level". Moreover, since very little arithmetic calculations are required, full advantage of assembly language programming can be taken.

2.7.2 Computer requirements

Memory size and execution times required for the validation checks of digital and analogue data are presented in table III (excluding data output). The memory requirements shown are in words (groups of 16 bits). All routines have been written in assembly language. The same requirements for the laboratory substation-computer model described in section (2.6) are presented in table IV. The model considered in table IV includes 4 data-switch words (24 switches) and 13 three-phase monitoring points i.e. 9 three-phase

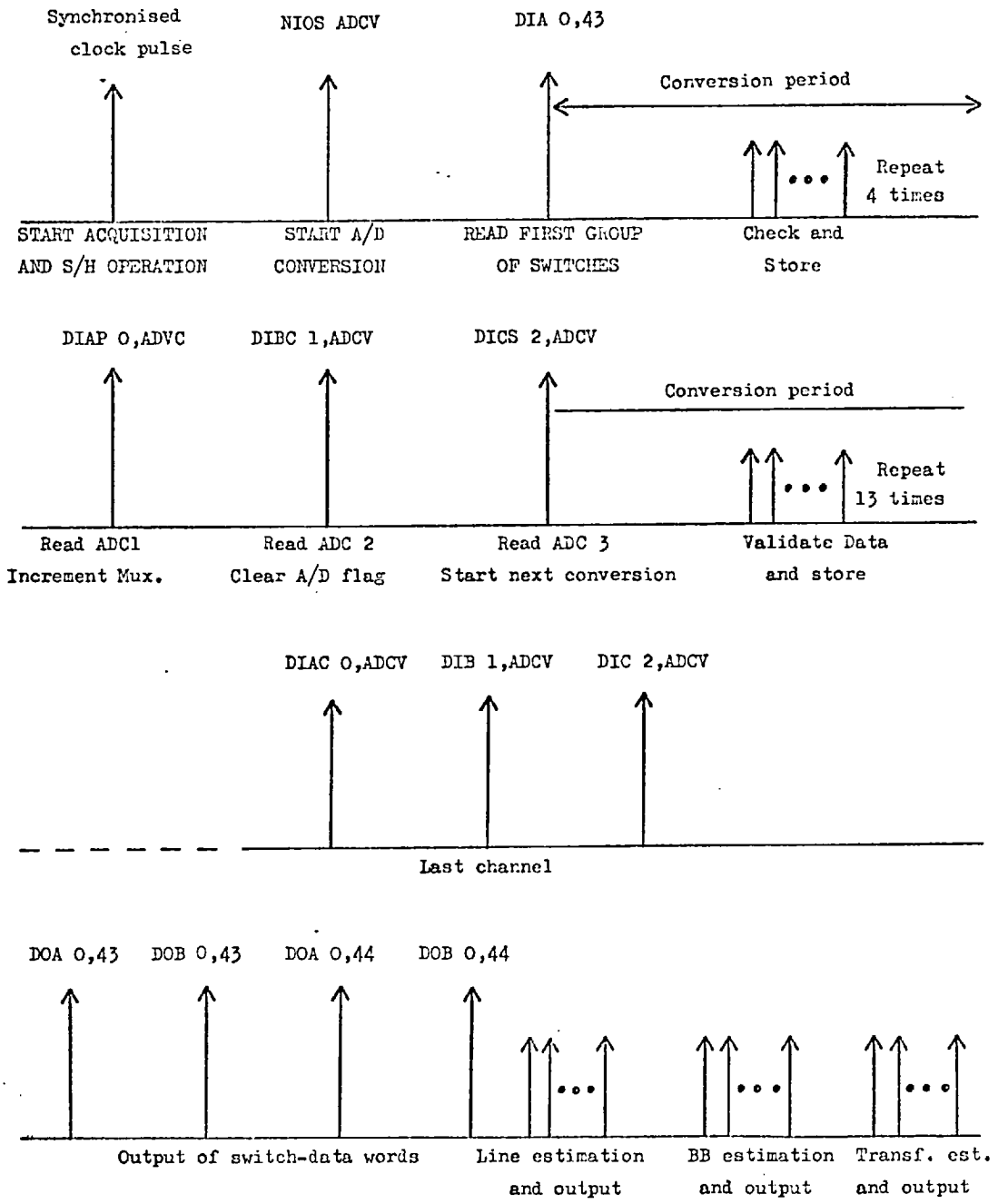


Figure 6. Sequence of events in the data acquisition and validation cycle

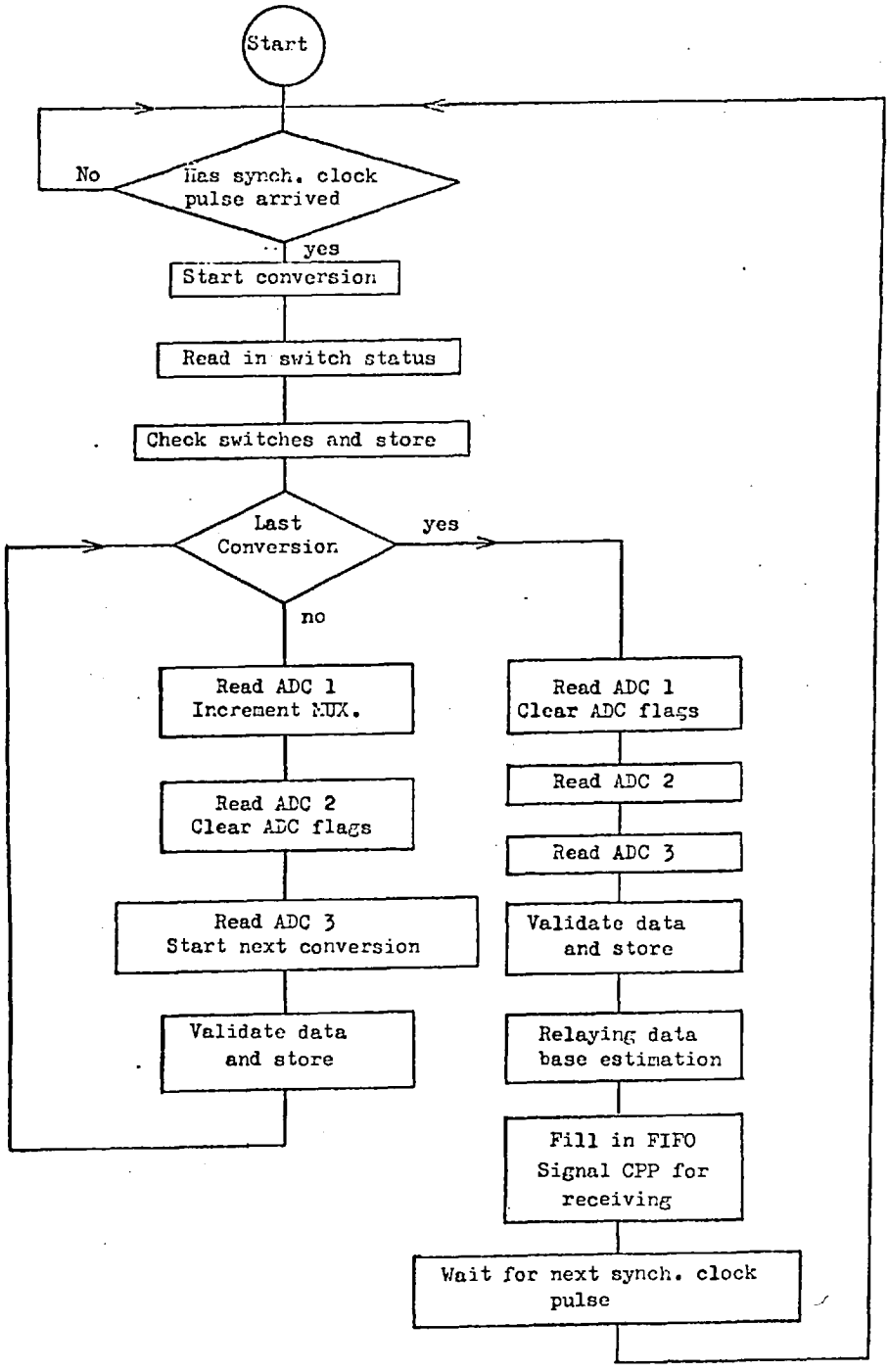


Figure 7. Flow diagram for data acquisition and validation

TABLE III

Memory Requirements and Execution Time
for Data Validation

Routine	Memory Requirement (Words)	Execution Time (μs)
Switch Status	8	15.60
Three-phase Current	43	59.25
Three-phase Voltage	35	43.65

TABLE IV

Computer Requirements for Laboratory
Substation-Computer Model

Routine	Memory Requirement (Words)	Execution Time (μs)
Data and Symbol Table	54	-
Start and Switches	40	91.45
Analogue Data Validation	613	707.85
Line Estimation	132	101.00
Busbars Estimation	136	194.40
Transformer Estimation	71	70.95
Total	1046	1165.65

currents and 4 three-phase voltages. Whilst this provides a basis for comparison, it should not be inferred that the routines have been developed to a high degree of programming efficiency.

2.8 Off-line tests

The aim of the tests was to check the performance of the algorithms of section (2.5). A test program written in NOVA assembly language was controlled from a teletype and provided special codes to select the validation algorithms for switch status, analogue data or the estimation procedure. Once one of the algorithms was selected, the appropriate routine was run using the data previously stored in the NOVA and the results were written out on the teletype.

2.8.1 Trials 1

The objective of these trials was to evaluate the performance of the algorithm of section (2.5.1) to check the switch status information. Different simulated combinations of the substation switches (Fig.4(a)) were used for the tests.

Figure 8 presents, as an example, the results of 4 tests and each case shows the data-switch word for each corner of the mesh-substation model. The results of the first test shows that words 1 and 4 which have no inconsistency in any of the switches is given a logic "0" as error flag but words 2 and 3 have set their error flags to indicate the inconsistency of the pair of bits of the first switch in both cases. Similar results can be seen from test 3 where the error flags in words 1, 3 and 4 correspond to a doubtful indication of switches 1, 3 and 1 respectively. The results of test 2, where the error flag of words 1 and 3 arise from a doubtful information for all

the switches indicate that each data-word gets the same error flag whatever the number of wrong switch information is in the word. The result confirms the need to provide within the relaying data validation algorithm the corrective action to clear any doubtful switch-data word i.e. all possible doubtful switches of each word. Test 4 presents a case where all 4 error flags are clear because there is no inconsistency in any one of the switches of each data-word.

2.8.2 Trials 2

This group of trials were performed to study the ability of the algorithm of section (2.5.2) to detect and identify gross measurement errors i.e. missing or erroneous data and to check the performance of the sample-switch routine to provide corrective action for a doubtful switch indication.

All tests start by checking the data-word for the switches of one corner. Two types of tests were performed to evaluate the algorithm for both types of data, current and voltage. Both routines are similar but for current values the presence of a non-zero value is used to correct a doubtful indication of the switch adjacent to the value being checked.

Figures 9 to 17 present the results from selected tests. The first test of Fig. 9 shows the identification of a real zero value using the zero flag of the 3 phases and the positive indication that the adjacent switch is open. All other tests detect a doubtful zero data either by the adjacent switch being closed (test 2) or having a doubtful indication (tests 3 and 4). Fig. 10 presents similar results when 2-phases are zero and the adjacent switch is closed. Similar tests were done for the current routine and selected results presented in Figs. 11 and 12.

Figure 13 illustrates the use of the sample-switch routine to correct a doubtful switch indication. Figures 14 and 15 show results from tests made to detect measurement errors when the residual value exceeds or not a pre-set tolerance (3 least significant bits for all tests). Tests were made to check the validity of such checks for positive and negative tolerance and some results are presented in Figures 16 and 17 respectively.

2.8.3 Trials 3

These trials were performed to check the ability of the estimation procedure proposed in section (2.5.3) to provide replacements for missing or erroneous data. The routine provided for the transmission line of the substation model (Fig. 4(a)) was used because it enabled a check on the procedure with measurement (voltage A or current S) having single redundancy (voltage C) or double redundancy (currents Q and T + U).

Figure 18 presents the results of a case where the current and voltage estimates are provided by the duplicate measurement (redundancy type I). Fig. 19 shows similar results but the line current estimate is taken from the line measurement (current S) because the series redundancy (current Q) has been tagged as doubtful data. The results of Figs. 20 and 21 correspond to the case when the estimates are given by the average value, if both measurements have either an error flag (voltages A and C) or have been accepted as valid data (currents S and Q). Fig. 22 illustrates the case when the averages provide the estimates as soon as the residual counter limit is exceeded. Fig. 23 shows the use of the redundancy type II (currents T and U) for the estimate when the measurement and the duplicate measurement are doubtful data.

SWITCH STATUS VALIDATION

CLOSE	OPEN	FLAG
111111	000000	0
111111	100000	1
011111	000000	1
000010	111101	0

SWITCH STATUS VALIDATION

CLOSE	OPEN	FLAG
100110	100110	1
011001	100110	0
001101	001101	1
101010	010101	0

SWITCH STATUS VALIDATION

CLOSE	OPEN	FLAG
001101	010010	1
110011	001100	0
101011	011100	1
111111	100000	1

SWITCH STATUS VALIDATION

CLOSE	OPEN	FLAG
111000	000111	0
010111	101000	0
100101	011010	0
001110	110001	0

Figure 8. Data-switch word validation

CLOSE	OPEN	FLAG
011111	100000	0
100000	000000	1
100000	000000	1
100000	000000	1

T. VALIDATION

PHASE	ZF	EF	MEASUREMENT
R	1	0	0000000000
Y	1	0	0000000000
B	1	0	0000000000

RESIDUAL COUNTER LESS THAN THREE

CLOSE	OPEN	FLAG
111111	000000	0
100000	000000	1
100000	000000	1
100000	000000	1

T. VALIDATION

PHASE	ZF	EF	MEASUREMENT
R	1	1	0000000000
Y	1	0	0000000000
B	1	0	0000000000

RESIDUAL COUNTER LESS THAN THREE

CLOSE	OPEN	FLAG
111111	100000	1
100000	000000	1
100000	000000	1
100000	000000	1

T. VALIDATION

PHASE	ZF	EF	MEASUREMENT
R	1	1	0000000000
Y	1	0	0000000000
B	1	0	0000000000

RESIDUAL COUNTER LESS THAN THREE

CLOSE	OPEN	FLAG
011111	010000	1
100000	000000	1
100000	000000	1
100000	000000	1

R	1	1	0000000000
Y	1	0	0000000000
B	1	0	0000000000

RESIDUAL COUNTER LESS THAN THREE

Figure 9. Identification of three-phase zero voltage

```
CLOSE     OPEN    FLAG
111111    000000    0
111111    000000    0
111111    000000    0
111111    000000    0


 T. VALIDATION

PHASE   ZF   EF   MEASUREMENT
  R     0    1    0001010000
  Y     1    0    0000000000
  B     1    0    0000000000

RESIDUAL COUNTER LESS THAN THREE

 T. VALIDATION

PHASE   ZF   EF   MEASUREMENT
  R     1    1    0000000000
  Y     0    0    0001010000
  B     1    0    0000000000

RESIDUAL COUNTER LESS THAN THREE

 T. VALIDATION

PHASE   ZF   EF   MEASUREMENT
  R     1    1    0000000000
  Y     1    0    0000000000
  B     0    0    0001010000

RESIDUAL COUNTER LESS THAN THREE
```

Figure 10. Identification of two-phase zero voltage

```
CLOSE      OPEN     FLAG
011111     100000    0
111111     000000    0
111111     000000    0
111111     000000    0


CT. VALIDATION

PHASE   ZF   EF   MEASUREMENT
  R     1    0    0000000000
  Y     1    0    0000000000
  B     1    0    0000000000

CLOSE      OPEN     FLAG
011111     100000    0

RESIDUAL COUNTER LESS THAN THREE


CLOSE      OPEN     FLAG
111111     000000    0
111111     000000    0
111111     000000    0
111111     000000    0


CT. VALIDATION

PHASE   ZF   EF   MEASUREMENT
  R     1    1    0000000000
  Y     1    0    0000000000
  B     1    0    0000000000

CLOSE      OPEN     FLAG
111111     000000    0

RESIDUAL COUNTER LESS THAN THREE
```

Figure 11. Identification of three-phase zero current with clear switch indication

CLOSE	OPEN	FLAG
111111	100000	1
111111	000000	0
111111	000000	0
111111	000000	0

CT. VALIDATION

PHASE	RF	EF	MEASUREMENT
R	1	1	0000000000
Y	1	0	0000000000
B	1	0	0000000000

CLOSE	OPEN	FLAG
111111	100000	1

RESIDUAL COUNTER LESS THAN THREE

CLOSE	OPEN	FLAG
011111	000000	1
111111	000000	0
111111	000000	0
111111	000000	0

CT. VALIDATION

PHASE	RF	EF	MEASUREMENT
R	1	1	0000000000
Y	1	0	0000000000
B	1	0	0000000000

CLOSE	OPEN	FLAG
011111	000000	1

RESIDUAL COUNTER LESS THAN THREE

Figure 12. Identification of three-phase zero current with doubtful switch indication

CLOSE	OPEN	FLAG
011111	000000	1
111111	000000	0
111111	000000	0
111111	000000	0

CT. VALIDATION

PHASE	PF	EF	MEASUREMENT
R	0	0	0000101000
Y	0	0	0000101000
B	0	0	1110110000

CLOSE	OPEN	FLAG
111111	000000	1

RESIDUAL COUNTER LESS THAN THREE

CLOSE	OPEN	FLAG
111111	000000	0
111111	000000	0
111111	000000	0
111111	000000	0

CT. VALIDATION

PHASE	PF	EF	MEASUREMENT
R	0	0	0001000001
Y	1	0	0000000000
B	0	0	1110111111

CLOSE	OPEN	FLAG
111111	000000	0

RESIDUAL COUNTER LESS THAN THREE

Figure 13. Illustration of the sample-switch routine

```
CLOSE     OPEN    FLAG
111111    000000   0
111111    000000   0
111111    000000   0
111111    000000   0


CT. VALIDATION

PHASE  RF   EF   MEASUREMENT
  R     1    1   0000000000
  Y     0    0   0001000001
  B     1    0   0000000010

CLOSE      OPEN    FLAG
111111    000000   0

RESIDUAL COUNTER LESS THAN THREE


CT. VALIDATION

PHASE  RF   EF   MEASUREMENT
  R     1    1   0000000000
  Y     1    0   0000000000
  B     0    0   1110111111

CLOSE      OPEN    FLAG
111111    000000   0

RESIDUAL COUNTER LESS THAN THREE


CT. VALIDATION

PHASE  RF   EF   MEASUREMENT
  R     0    1   1110111111
  Y     1    0   0000000000
  B     1    0   0000000000

CLOSE      OPEN    FLAG
111111    000000   0

RESIDUAL COUNTER LESS THAN THREE


CT. VALIDATION

PHASE  RF   EF   MEASUREMENT
  R     1    1   0000000000
  Y     0    0   0001000001
  B     0    0   1111110110

CLOSE      OPEN    FLAG
111111    000000   0

RESIDUAL COUNTER LESS THAN THREE
```

Figure 14. Detection of doubtful data using the residual value and residual counter with zero data in one or two phases

```
CLOSE      OPEN     FLAG
111111     000000    0
111111     000000    0
111111     000000    0
111111     000000    0


CT. VALIDATION

PHASE   PF   EF   MEASUREMENT
  R     0    1    0001010000
  Y     0    0    1111101100
  B     0    0    1111110110

CLOSE      OPEN     FLAG
111111     000000    0

RESIDUAL COUNTER LESS THAN THREE


CT. VALIDATION

PHASE   PF   EF   MEASUREMENT
  R     1    0    0000000000
  Y     0    0    0001000001
  B     0    0    1110111111

CLOSE      OPEN     FLAG
111111     000000    0

RESIDUAL COUNTER LESS THAN THREE
```

Figure 15. Detection of doubtful data using the residual value and residual counter

```
CT. VALIDATION

PHASE   PF   EF   MEASUREMENT
  R     0    0    1111011101
  Y     0    0    0000010100
  B     0    0    0000010100

RESIDUAL COUNTER LESS THAN THREE

CT. VALIDATION

PHASE   PF   EF   MEASUREMENT
  R     0    0    1111011110
  Y     0    0    0000010100
  B     0    0    0000010100

RESIDUAL COUNTER LESS THAN THREE

CT. VALIDATION

PHASE   PF   EF   MEASUREMENT
  R     0    0    1111011111
  Y     0    0    0000010100
  B     0    0    0000010100

RESIDUAL COUNTER LESS THAN THREE

CT. VALIDATION

PHASE   PF   EF   MEASUREMENT
  R     0    1    1111000000
  Y     0    0    0000010100
  B     0    0    0000010100

RESIDUAL COUNTER LESS THAN THREE

CT. VALIDATION

PHASE   PF   EF   MEASUREMENT
  R     0    1    1111000001
  Y     0    0    0000010100
  B     0    0    0000010100

RESIDUAL COUNTER LESS THAN THREE
```

Figure 16. Detection of doubtful data by positive residual value exceeding tolerance

```
CT. VALIDATION

PHASE   ZF   EF   MEASUREMENT
  R     0    0    1111001001
  Y     0    0    0000011001
  B     0    0    0000011001

RESIDUAL COUNTER LESS THAN THREE

CT. VALIDATION

PHASE   ZF   EF   MEASUREMENT
  R     0    0    1111001000
  Y     0    0    0000011001
  B     0    0    0000011001

RESIDUAL COUNTER LESS THAN THREE

CT. VALIDATION

PHASE   ZF   EF   MEASUREMENT
  R     0    0    1111000111
  Y     0    0    0000011001
  B     0    0    0000011001

RESIDUAL COUNTER LESS THAN THREE

CT. VALIDATION

PHASE   ZF   EF   MEASUREMENT
  R     0    0    1111000110
  Y     0    0    0000011001
  B     0    0    0000011001

RESIDUAL COUNTER LESS THAN THREE

CT. VALIDATION

PHASE   ZF   EF   MEASUREMENT
  R     0    1    1111000101
  Y     0    0    0000011001
  B     0    0    0000011001

RESIDUAL COUNTER LESS THAN THREE
```

Figure 17. Detection of doubtful data by negative residual exceeding tolerance

CLOSE	OPEN	FLAG
111111	000000	0
111111	000000	0
111111	000000	0
111111	000000	0

VOLTAGE "A"

PHASE	ZF	EF	MEASUREMENT
R	1	1	0000000000
Y	1	0	0000000000
B	1	0	0000000000

VOLTAGE "C"

PHASE	ZF	EF	MEASUREMENT
R	1	0	0000000000
Y	0	0	0001000001
B	0	0	1110111111

CURRENT "B"

PHASE	ZF	EF	MEASUREMENT
R	1	1	0000000000
Y	0	0	1110110000
B	0	0	0000101000

CURRENT "C"

PHASE	ZF	EF	MEASUREMENT
R	0	0	0000101000
Y	0	0	1110110000
B	0	0	0000101000

LINE VOLTAGE ESTIMATE

R	0	0	0000000000
Y	0	0	0001000001
B	0	0	1110111111

LINE CURRENT ESTIMATE

R	0	0	0000101000
Y	0	0	1110110000
B	0	0	0000101000

Figure 18. Transmission line estimates by duplicate measurement

CLOSE	OPEN	FLAG
111111	000000	0
111111	000000	0
111111	000000	0
111111	000000	0

VOLTAGE "A"

PHASE	PF	EF	MEASUREMENT
R	1	1	0000000000
Y	0	0	0001000001
B	1	0	0000000000

VOLTAGE "C"

PHASE	PF	EF	MEASUREMENT
R	1	0	0000000000
Y	0	0	0001000001
B	0	0	1110111111

CURRENT "S"

PHASE	PF	EF	MEASUREMENT
R	0	0	0000101000
Y	0	0	1110110000
B	0	0	0000101000

CURRENT "G"

PHASE	PF	EF	MEASUREMENT
R	0	1	0000101000
Y	1	0	0000000000
B	0	0	0000101000

LINE VOLTAGE ESTIMATE

R	0	0	0000000000
Y	0	0	0001000001
B	0	0	1110111111

LINE CURRENT ESTIMATE

R	0	0	0000101000
Y	0	0	1110110000
B	0	0	0000101000

Figure 19. Line voltage estimate by duplicate measurement and line current estimate taken directly from the line measurement

SWITCH STATUS VALIDATION

CLOSE	OPEN	FLAG
111111	000000	0
111111	000000	0
111111	000000	0
111111	000000	0

VOLTAGE "A"

PHASE	ZF	EF	MEASUREMENT
R	0	1	0001011010
Y	0	0	1111100101
B	0	0	1111110000

VOLTAGE "B" CTDL S TO START

PHASE	ZF	EF	MEASUREMENT
R	0	1	0001010000
Y	0	0	1111110000
B	0	0	1111110000

CURRENT "S"

PHASE	ZF	EF	MEASUREMENT
R	0	0	0000100010
Y	0	0	1110110010
B	0	0	0000101100

CURRENT "B"

PHASE	ZF	EF	MEASUREMENT
R	0	0	0000101100
Y	0	0	1110101100
B	0	0	0000100110

LINE VOLTAGE ESTIMATE

R	0	0	0001010101
Y	0	0	1111101010
B	0	0	1111110000

LINE CURRENT ESTIMATE

R	0	0	0000100111
Y	0	0	1110101111
B	0	0	0000101001

Figure 20. Transmission line estimates by average values

SWITCH STATUS VALIDATION

CLOSE	OPEN	FLAG
111111	000000	0
111111	000000	0
111111	000000	0
111111	000000	0

VOLTAGE "A"

PHASE	RF	EF	MEASUREMENT
R	0	1	0001000110
Y	0	0	1111100000
B	0	0	1111110000

VOLTAGE "C"

PHASE	RF	EF	MEASUREMENT
R	0	1	0001010000
Y	0	0	1111110000
B	0	0	1111001110

CURRENT "S"

PHASE	RF	EF	MEASUREMENT
R	0	0	0000100010
Y	0	0	1110110010
B	0	0	0000101100

CURRENT "C"

PHASE	RF	EF	MEASUREMENT
R	0	0	0000101000
Y	0	0	1110110000
B	0	0	0000101000

LINE VOLTAGE ESTIMATE

R	0	0	0001001011
Y	0	0	1111101000
B	0	0	1111011111

LINE CURRENT ESTIMATE

R	0	0	0000100101
Y	0	0	1110110001
B	0	0	0000101010

Figure 21. Transmission line estimates by average values

```
SWITCH STATUS VALIDATION

CLOSE     OPEN    FLAG
111111    000000    0
111111    000000    0
111111    000000    0
111111    000000    0


VOLTAGE "A"

PHASE  PF   EF   MEASUREMENT
  R     0    1    0001010000
  Y     0    0    1111110000
  B     0    0    1111101000


VOLTAGE "C"

PHASE  PF   EF   MEASUREMENT
  R     0    1    0001010000
  Y     0    0    1111011000
  B     0    0    1111110110


CURRENT "B"

PHASE  PF   EF   MEASUREMENT
  R     0    1    0100011000
  Y     0    0    1100111000
  B     1    0    0000000000


CURRENT "C"

PHASE  PF   EF   MEASUREMENT
  R     0    0    0100011110
  Y     0    0    1011100010
  B     1    0    0000000000


CURRENT "T"

PHASE  PF   EF   MEASUREMENT
  R     0    1    0010011111
  Y     0    0    1101110001
  B     1    0    0000000000


CURRENT "U"

PHASE  PF   EF   MEASUREMENT
  R     0    0    0010010000
  Y     0    0    1101101000
  B     1    0    0000000000

RESIDUAL COUNTER GREATER THAN THREE


LINE VOLTAGE ESTIMATE
  R     0    0    0001010000
  Y     0    0    1111100100
  B     0    0    1111101111


LINE CURRENT ESTIMATE
  R     0    0    0100011011
  Y     0    0    1100001101
  B     0    0    0000000000
```

Figure 22. Transmission line estimates by average values when residual counter has been exceeded

SWITCH STATUS VALIDATION

CLOSE	OPEN	FLAG
111111	000000	0
111111	000000	0
111111	000000	0
111111	000000	0

VOLTAGE "A"

PHASE	PF	EF	MEASUREMENT
R	0	0	0001011010
Y	0	0	1111001110
B	0	0	1111011000

VOLTAGE "C"

PHASE	PF	EF	MEASUREMENT
R	0	1	0001000110
Y	0	0	1111100010
B	0	0	1111110000

CURRENT "S"

PHASE	PF	EF	MEASUREMENT
R	0	1	1110110000
Y	1	0	0000000000
B	0	0	0000101000

CURRENT "C"

PHASE	PF	EF	MEASUREMENT
R	0	1	1110111010
Y	0	0	0000101100
B	0	0	0000100010

CURRENT "T"

PHASE	PF	EF	MEASUREMENT
R	0	0	1111100010
Y	0	0	0000010010
B	0	0	0000010000

CURRENT "U"

PHASE	PF	EF	MEASUREMENT
R	0	0	1111011000
Y	0	0	0000011001
B	0	0	0000010100

LINE VOLTAGE ESTIMATE

R	0	0	0001011010
Y	0	0	1111001110
B	0	0	1111011000

LINE CURRENT ESTIMATE

R	0	0	1110111010
Y	0	0	0000101011
B	0	0	0000100100

Figure 23. Line current estimate by network redundancy

2.9 On-line tests

On-line data validation tests were made using the substation computer model system described in section (2.6). In this section only the hardware set-up and the software which were used for the switch and analogue data validation are described. Some typical results of the on-line tests are presented.

2.9.1 Hardware arrangement

The system hardware configuration set up for data validation for on-line tests is illustrated schematically in Fig. 24. As indicated in section (2.6) the mesh substation in Fig. 4(a) is simulated in two parts, called for identification purposes, the primary and secondary circuits. The primary circuit refers to the mesh corner simulated in 3-phase form with transducers for current and voltage measurements and contactors acting as circuit breakers and isolators. Push button switches are available for manual closing and opening. The term secondary circuit refers to the rest of the substation and simulates switch positions in the adjacent mesh corners by single phase relays which are manually operated. Substation switching configuration for on-line tests are simulated using this manual operation facility available for each switch. An interface has been specially constructed[14] to provide 4-16 bit digital word inputs for switch status from the four corners of a mesh substation.

One primary circuit of the substation model[13] consisting of a contactor unit and a current and voltage transducers is connected to a 3-phase resistive load for analogue data validation on-line tests. The load current is limited to a maximum of 10 amp r.m.s. The current signals monitored by the computer are measured by air cored current

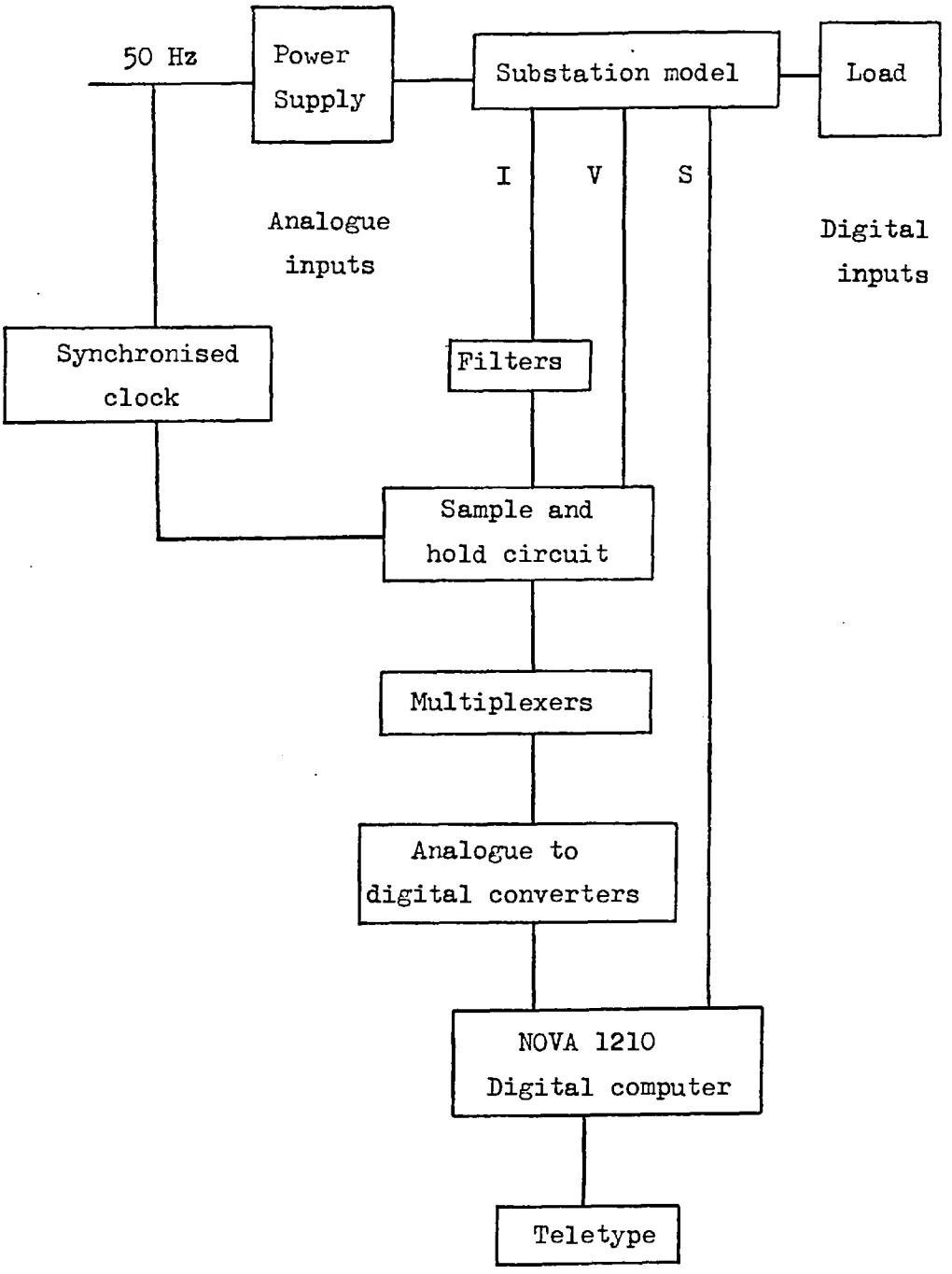


Figure 24. Hardware configuration for on-line data validation tests.

transformers (linear couplers). The linear coupler output is proportional to the rate of change of input current, such differentiation leads to the amplification of noise and high frequency components producing distortion on the output. Active low-pass filters with a 130 Hz cut-off frequency and a gain of about 50 are used for filtering and scaling the distorted waveforms of the linear coupler outputs during normal operation. The maximum voltage that can be applied to the A/D converter is $\pm$ 10 volt. No other precautions were taken to obtain more accurate measurements. The rest of the hardware configuration has been described in section (2.6.2).

2.9.2 Test program

Figure 25 shows the flowchart of the program which was used for on-line tests. The program starts by setting the synchronised clock to the sample frequency desired, the multiplexer addresses and mode of operation. Then it waits for the user to select either the switch-data validation test or the analogue data validation test. In both cases, once the proper code test is selected, the validation procedure starts reading the digital data-word for the switch status of the first mesh corner (primary circuit) and the validation is carried out using the procedure of section (2.5.1). Once validated the switch-word is stored and the process is repeated for all corners of the mesh substation. If the switch validation test was selected the program writes out on the teletype the validation results and returns to the start point.

In the case of analogue test, the program, after validating the switch-data words, continues reading, validating and storing current and voltage values up to the number of cycles previously pre-set. The hardware configuration set up for the on-line tests assigns the first

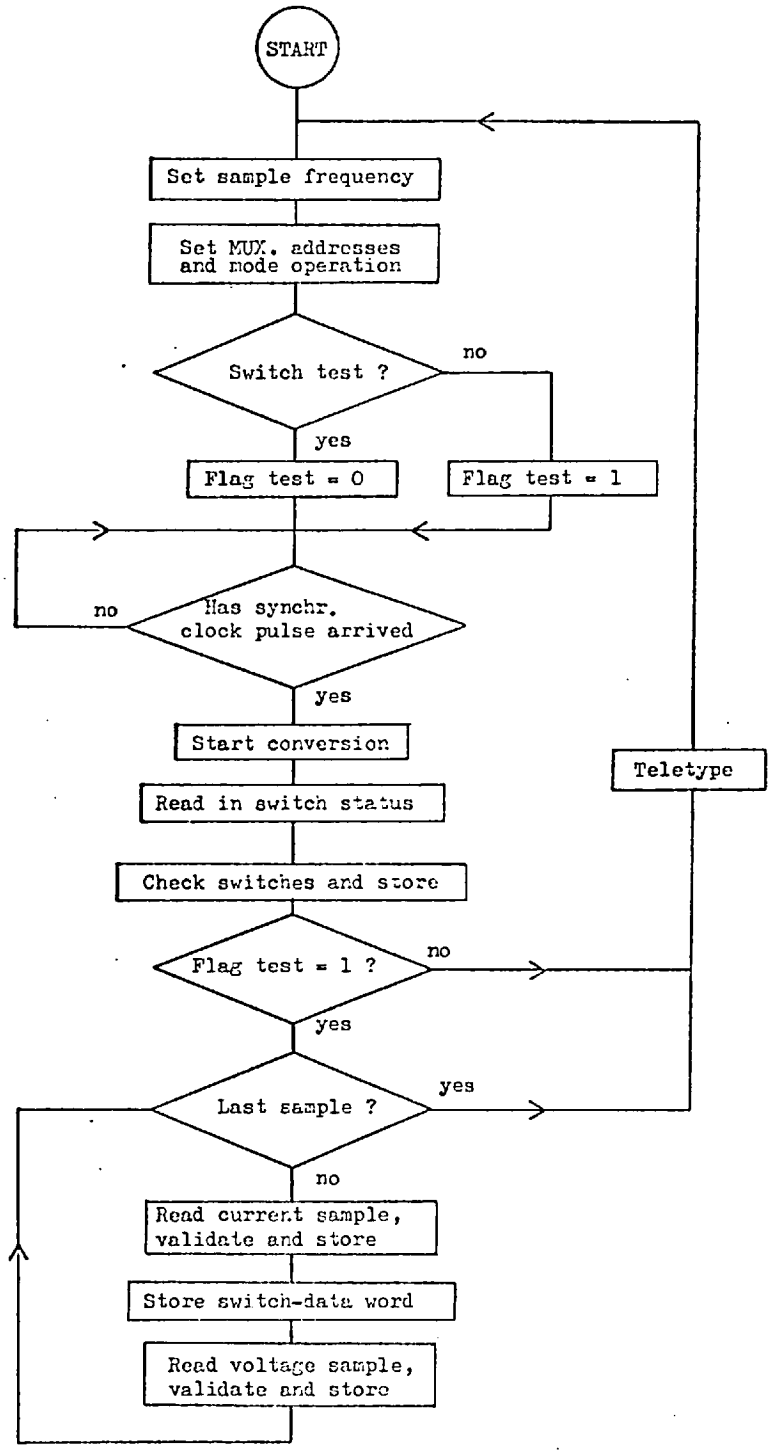


Figure 25. Flow diagram for on-line data validation

channel of the multiplexers to the monitored current values and the second one to the voltage transformer, so the analogue validation starts checking the first three-phase current sample, continues with the first three-phase voltage sample and repeats the process for the number of cycles desired. The validation is done using the procedure of section (2.5.2) so during the current sample validation the presence of a non-zero value is used to correct a possible doubtful indication of the associated switch of the mesh corner. Then after the storage of each current sample the program also stored the switch-data word used during that validation. Once the program has completed its number of cycles, the results are printed out on the teletype and returns to the clock setting.

2.9.3 On-line test results

The software described in section (2.9.2) was used to detect and identify doubtful indication of the switch status and to validate current and voltage samples taken through the hardware configuration described in section (2.9.1). Many tests for different substation switching configurations and tolerance limits of the residual value were carried out.

Figures 26 to 29 show some typical results for the digital and analogue data validation algorithms.

The large unbalance present in all three-phase current and voltage samples was mainly due to:

(i) the linear coupler outputs were different for equal inputs

(ii) inequality in the turns-ratios of the voltage transformers

(iii) different phase shifts produced by asymmetric assembling of the linear couplers in the model.

CLOSE	OPEN	FLAG
110101	110000	1
111111	000000	0
101110	100010	1
111111	000000	0

CLOSE	OPEN	FLAG
111111	000000	0
111111	000000	0
111111	000000	0
111111	000000	0

CLOSE	OPEN	FLAG
111111	000000	0
110101	100001	1
111111	000000	0
101110	000110	1

CLOSE	OPEN	FLAG
111111	010001	1
111111	000000	0
111111	000011	1
111111	000000	0

CLOSE	OPEN	FLAG
111111	000000	0
111111	000000	0
110110	001010	1
011101	000110	1

CLOSE	OPEN	FLAG
111111	000000	0
111111	100010	1
111111	000000	0
111111	010010	1

CLOSE	OPEN	FLAG
110101	100001	1
011110	100001	0
111001	001010	1
101110	000110	1

CLOSE	OPEN	FLAG
111111	001001	1
111111	100010	1
111111	010010	1
111111	010010	1

CLOSE	OPEN	FLAG
110101	100001	1
101110	100001	1
111001	001010	1
110101	000110	1

CLOSE	OPEN	FLAG
111111	101000	1
111111	100010	1
111111	000000	0
111111	000000	0

CLOSE	OPEN	FLAG
011111	000000	1
011101	010001	1
011111	000001	1
011110	001001	1

CLOSE	OPEN	FLAG
111111	000000	0
111111	000000	0
111111	010011	1
111111	000010	1

CLOSE	OPEN	FLAG
111111	000000	0
111111	000000	0
111111	000000	0
111111	000000	0

PHASE	ZF	EF	MEASUREMENT
R	0	1	0001001111
Y	1	0	0000000000
B	1	0	0000000000

CLOSE	OPEN	FLAG
111111	000000	0

PHASE	ZF	EF	MEASUREMENT
R	0	1	1111100001
Y	0	0	1111101000
B	1	0	1111111111

PHASE	ZF	EF	MEASUREMENT
R	0	1	0001011011
Y	0	0	1111110111
B	0	0	1111110110

CLOSE	OPEN	FLAG
111111	000000	0

PHASE	ZF	EF	MEASUREMENT
R	0	0	0000010100
Y	0	0	1111101111
B	0	0	0000001001

PHASE	ZF	EF	MEASUREMENT
R	0	1	0000111101
Y	1	0	1111111110
B	0	0	1110101010

CLOSE	OPEN	FLAG
111111	000000	0

PHASE	ZF	EF	MEASUREMENT
R	0	1	1111110101
Y	0	0	1111010001
B	0	0	1110101101

PHASE	ZF	EF	MEASUREMENT
R	0	1	1110110101
Y	0	0	1111000100
B	0	0	1110110110

CLOSE	OPEN	FLAG
111111	000000	0

PHASE	ZF	EF	MEASUREMENT
R	0	1	0000001101
Y	0	0	0000101101
B	0	0	1111101101

PHASE	ZF	EF	MEASUREMENT
R	0	1	1110101001
Y	0	0	0000001100
B	0	0	0000110111

CLOSE	OPEN	FLAG
111111	000000	0

PHASE	ZF	EF	MEASUREMENT
R	0	1	1111101100
Y	0	0	0000101001
B	0	0	0000010011

PHASE	ZF	EF	MEASUREMENT
R	0	1	0000010100
Y	0	0	0000011110
B	0	0	0101010101

CLOSE	OPEN	FLAG
111111	000000	0

PHASE	ZF	EF	MEASUREMENT
R	0	1	1111100001
Y	0	0	1111100001
B	0	0	1111111010

PHASE	ZF	EF	MEASUREMENT
R	0	1	0001011011
Y	0	0	1111110001
B	0	0	1111010000

CLOSE	OPEN	FLAG
111111	000000	0

PHASE	ZF	EF	MEASUREMENT
R	0	0	0000010110
Y	0	0	1111101111
B	0	0	0000001001

PHASE	ZF	EF	MEASUREMENT
R	0	1	0000110010
Y	1	0	1111111110
B	0	0	1110101111

CLOSE	OPEN	FLAG
111111	000000	0

PHASE	ZF	EF	MEASUREMENT
R	0	1	1111110001
Y	0	0	1111010010
B	0	0	1110101000

Figure 27. On-line analogue data validation with four bits tolerance and four samples/cycle

CLOSE	OPEN	FLAG
111111	000000	0
111111	000000	0
111111	000000	0
111111	000000	0

PHASE	ZF	EF	MEASUREMENT
R	0	1	1111100111
Y	0	0	0000001101
B	0	0	0001101100

CLOSE	OPEN	FLAG
111111	000000	0

PHASE	ZF	EF	MEASUREMENT
R	0	0	1111111000
Y	0	0	0000110001
B	1	0	0000000011

PHASE	ZF	EF	MEASUREMENT
R	0	1	1111101010
Y	0	0	0000001110
B	0	0	0001101100

CLOSE	OPEN	FLAG
111111	000000	0

PHASE	ZF	EF	MEASUREMENT
R	0	0	1111110100
Y	0	0	0000101010
B	0	0	0000001101

PHASE	ZF	EF	MEASUREMENT
R	0	1	0000000001
Y	2	0	0000001101
B	0	0	0001100100

CLOSE	OPEN	FLAG
111111	000000	0

PHASE	ZF	EF	MEASUREMENT
R	0	0	1111110011
Y	0	0	0000100000
B	0	0	0000011010

PHASE	ZF	EF	MEASUREMENT
R	0	1	0000100111
Y	0	0	0000001100
B	0	0	0001001111

CLOSE	OPEN	FLAG
111111	000000	0

PHASE	ZF	EF	MEASUREMENT
R	0	0	1111110100
Y	0	0	0000011000
B	0	0	0000100001

PHASE	ZF	EF	MEASUREMENT
R	0	1	0000000011
Y	0	0	0000001011
B	0	0	0000110101

CLOSE	OPEN	FLAG
111111	000000	0

PHASE	ZF	EF	MEASUREMENT
R	0	0	1111111000
Y	0	0	0000010000
B	0	0	0000100111

PHASE	ZF	EF	MEASUREMENT
R	0	1	0001110011
Y	0	0	0000001011
B	0	0	0000101001

CLOSE	OPEN	FLAG
111111	000000	0

PHASE	ZF	EF	MEASUREMENT
R	1	0	1111111111
Y	0	0	0000000100
B	0	0	0000100110

PHASE	ZF	EF	MEASUREMENT
R	0	1	0001100101
Y	0	0	0000000111
B	0	0	1111110101

CLOSE	OPEN	FLAG
111111	000000	0

PHASE	ZF	EF	MEASUREMENT
R	0	0	0000001001
Y	1	0	1111111100
B	0	0	0000100101

PHASE	ZF	EF	MEASUREMENT
R	0	1	0001101010
Y	0	0	0000000111
B	0	0	1111010111

CLOSE	OPEN	FLAG
111111	000000	0

PHASE	ZF	EF	MEASUREMENT
R	0	0	0000001111
Y	0	0	1111110100
B	0	0	0000100011

Figure 28. On-line analogue data validation with 6 bits tolerance and 16 samples/cycle

CLOSE	OPEN	FLAG
111111	100000	1
111111	000000	0
111111	000000	0
111111	000000	0

PHASE	RF	EF	MEASUREMENT
R	0	0	0001000000
Y	1	0	0000000000
B	0	0	1110100000

CLOSE	OPEN	FLAG
111111	000000	1

PHASE	RF	EF	MEASUREMENT
R	0	0	0000100010
Y	0	0	1111101101
B	0	0	0000001000

PHASE	RF	EF	MEASUREMENT
R	0	0	0000111000
Y	1	0	1111111111
B	0	0	1110011100

CLOSE	OPEN	FLAG
111111	000000	1

PHASE	RF	EF	MEASUREMENT
R	0	0	0000101100
Y	0	0	1111110110
B	0	0	1111110010

PHASE	RF	EF	MEASUREMENT
R	1	0	0000000001
Y	1	0	0000000000
B	0	0	1110100000

CLOSE	OPEN	FLAG
111111	000000	1

PHASE	RF	EF	MEASUREMENT
R	0	0	0000101100
Y	1	0	0000000011
B	0	0	1110011001

PHASE	RF	EF	MEASUREMENT
R	0	0	1110000111
Y	1	0	1111111101
B	0	0	1110001000

CLOSE	OPEN	FLAG
111111	000000	1

PHASE	RF	EF	MEASUREMENT
R	0	0	0000100100
Y	0	0	0000011101
B	0	0	1111100101

PHASE	RF	EF	MEASUREMENT
R	0	0	1110100010
Y	0	0	0000001000
B	0	0	0000010010

CLOSE	OPEN	FLAG
111111	000000	1

PHASE	RF	EF	MEASUREMENT
R	0	0	0000000000
Y	0	0	0000101000
B	0	0	0000010000

PHASE	RF	EF	MEASUREMENT
R	0	0	1111110100
Y	0	0	0000101011
B	0	0	0000001011

PHASE	RF	EF	MEASUREMENT
R	0	0	0000000111
Y	0	0	0000001110
B	0	0	0001100100

CLOSE	OPEN	FLAG
111111	000000	1

PHASE	RF	EF	MEASUREMENT
R	0	0	1111110100
Y	0	0	0000011001
B	0	0	0000011111

Figure 29. On-line analogue data validation with 7 bits tolerance, 16 samples/cycle illustrating the use of the sample/swith validation

2.10 Final comments

The use of two complementary bits for each switch position and a sample-switch routine provides a simple and reliable validation for any doubtful switch indication.

Zero data detection must be considered within the hardware design to allow validation routines to be carried out during the sampling interval. The hardware identification (zero flag) of zero data and the information of the switch adjacent to the value being checked, provide simple detection of missing data from all 3 phases. The use of the residual value and a residual counter allows the detection of missing data in one or two phases and the presence of errors bigger than a required tolerance. The self-checking criterion provided for analogue data validation gives great flexibility to the validation scheme.

The use of two data flags (zero and error) with any type of redundancy available for each measurement, enables use of a simple estimation routine to provide replacements for missing or erroneous data. The estimate values are transferred via a first-in, first-out buffer to assemble the relaying data base in the control and protection processor. The execution time obtained for the one corner mesh-substation model shows the feasibility to apply the data validation algorithm proposed in this chapter, in such model, with a sampling rate up to 16 samples per cycle. The additional memory requirements of the validation program for a computer relaying scheme are modest.

CHAPTER III

SYSTEM RELIABILITY ANALYSIS

3.1 Introduction

In system reliability analysis, the system is normally modelled by a probabilistic graph in which each node and each edge has an assigned weight which is the probability of being good. A basic measure in system reliability for relay networks[15] and process systems[23] is the terminal-pair reliability but for communication networks several other reliability measures based on node and edge connectivity[37] are also important.

This chapter will briefly review known procedures for the terminal-pair reliability calculation and the limitations that arise in their application. A comparison is made of the approximation techniques for the minimal cut-set approach. Only the case where the failure probabilities of the elements are statistically independent is considered.

3.2 Terminal-pair reliability problem

The terminal reliability with respect to a pair of nodes s and t, here called the system reliability, is the probability that there exists at least one path from s to t along which all nodes and branches are good. In performing such reliability analysis the logical approach is to decompose the system into functional entities composed of units, subsystems or components. The subdivision generates a reliability graph where the system components are replaced by single lines here called

elements, branches, edges or links and their terminals are called nodes. Models are then formulated to fit this logical structure and the calculus of probability is used to compute the system reliability in terms of component reliabilities.

When the reliability graph is series-parallel with respect to the terminal pair (s,t) and all nodes are perfectly reliable, their reliability can be evaluated very simply[17]. It is obvious that the structural nature of practical systems will generally be such that neither a pure series nor a pure parallel reliability model is appropriate. The reliability analysis when all components have identical reliability and all nodes are perfectly reliable or vice versa becomes a combinatorial problem for which some methods are available[15],[38]. For the general case of non-series parallel networks with unequal different reliabilities the problem is more complicated and more general techniques are needed. The following methods can be used for the terminal-pair reliability problem:

1. Event enumeration[18],[19],[20]
2. Factoring theorem[21],[22]
3. Tie-set enumeration[23],[24],[25],[26],[27]
4. Cut-set enumeration[30],[31],[32],[33],[34]

3.3 Event enumeration

In order to calculate the system reliability it is necessary to find all mutually exclusive events which result in successful system operation. The basic approach to apply the event enumeration method requires a table of all possible logical occurrences in the system, called the reliability table. The table is then separated into successful and unsuccessful events and these tables are called partial reliability

tables. If the tables are properly prepared all the events are mutually exclusive. The probability of success is then merely the sum of the occurrence probabilities of each successful event. The reliability could also be computed by first finding the probability of failure, which is given by the sum of the occurrence probabilities of the unsuccessful events.

The technique can be illustrated using the reliability graph of Fig. 1. The event-enumeration procedure consists of constructing a table[18] with k event-groups, k = 0,n where n is the number of elements in the system.

If elements are represented by their own designation when they work properly and by its complement if they fail, then group 0 of Table I represents no failures, and the number of terms in this group is given by ${}_0C_3$. Group 1 represents one failure and contains ${}_1C_3$ elements. The success of an event may be determined by inspecting the reliability graph as follows: the failed units in the event in question are deleted from the graph, and if the remaining structure has at least one continuous path connecting nodes s and t, the event is successful. All the circled events in Table I are unsuccessful ones, and the uncircled ones are all successful. Thus the probability of successful communication between nodes s and t is given by

$$P_S(s,t) = \Pr\{E_1 + E_2 + E_3\} \tag{3.1}$$

Since all these events are mutually exclusive, the probability of the union of these events is the sum of the probabilities of each event taken separately. If we assume that all units are identical, and that the units have a probability of success p and a probability of failure 1-p, Eq. (3.1) reduces to

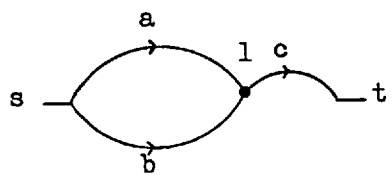


Figure 1. Reliability graph

TABLE I

Event-space for system of Fig. 1

Group 0	Group 1	Group 2	Group 3
$E_1 = a\ b\ c$	$E_2 = \bar{a}\ b\ c$	$E_5 = \bar{a}\ \bar{b}\ c$	$E_8 = \bar{a}\ \bar{b}\ \bar{c}$
	$E_3 = a\ \bar{b}\ c$	$E_6 = \bar{a}\ b\ \bar{c}$	
	$E_4 = a\ b\ \bar{c}$	$E_7 = a\ \bar{b}\ \bar{c}$	

$$P_S(s,t) = p^3 + 2p^2 (1-p) = 2p^2 - p^3$$

Dubes[19] presents a procedure to simplify the reliability expression when all the elements have unequal probabilities of success based on successive applications of the theorem

$$p_a = p_a \cdot (1-p_b) + p_a \cdot p_b$$

for which the partial reliability table of successful events is assumed to be available. Wing and Demetriov[20] report a method to check automatically if there exists at least one path from every node to every other for each of the events of the complete reliability table. This method also has the advantage that when all the events have been examined, all the terminal reliabilities are obtained.

Event enumeration methods are exhaustive algorithms where the total number of events is given by 2^n, n being the number of elements in the system. This technique has great computational difficulties for large systems. Systems with more than 20 elements are very difficult to solve even with a digital computer because they have over one million events.

3.4 Factoring theorem

Although the series-parallel technique[17],[21] is extremely useful for the reliability analysis of a large class of redundant networks, not all reliability diagrams can be reduced to the series-parallel model. For example, bridge-type networks cannot be decomposed into series-parallel elements.

For redundant networks the theorem[22] may be stated in the following form:

"A redundant network reliability function is equal to the reliability factor of any one single element of the network times the resulting

network reliability function with the terminals of the element short-circuited, plus the unreliability factor of the element times the resulting network reliability function with the terminals of the element open-circuited".

The probability of successful communication between a pair of node s and t can be expressed by the factoring theorem as follows:

$$P_S(s,t) = p_j \left\{ P_S(s,t) \right\}_{p_{j=0}} + (1-p_j) \left\{ P_S(s,t) \right\}_{p_{j=1}} \tag{3.2}$$

where p_j is the probability that the jth element of the network will not fail, $\left\{ P_S(s,t) \right\}_{p_{j=1}}$ and $\left\{ P_S(s,t) \right\}_{p_{j=0}}$ denote the probability of successful communication between nodes s and t assuming that element j is replaced by an open-circuit and a short-circuit, respectively. In complex reliability calculations it is often necessary to apply the factoring theorem in several steps to break up the system into smaller units of series and parallel networks for a systematic evaluation[21] of the network reliability function. Figure 2 shows an example of the application of the factoring theorem for a non-series parallel network with adjacent interconnecting branches. Misra[21] uses different combinations of the states of the interconnecting branches to define corresponding series-parallel networks reducing the total number of combinations in networks with non-adjacent interconnecting branches.

The algorithms based on an iterative application of the factoring theorem for all non series-parallel elements are very effective for networks containing few non series-parallel elements between any pair of nodes. However, for highly interconnected graphs (non series-parallel graphs) with b-branches and n-nodes, the required computation time can be shown to grow approximately as 2^{b-n}. The method is not applicable

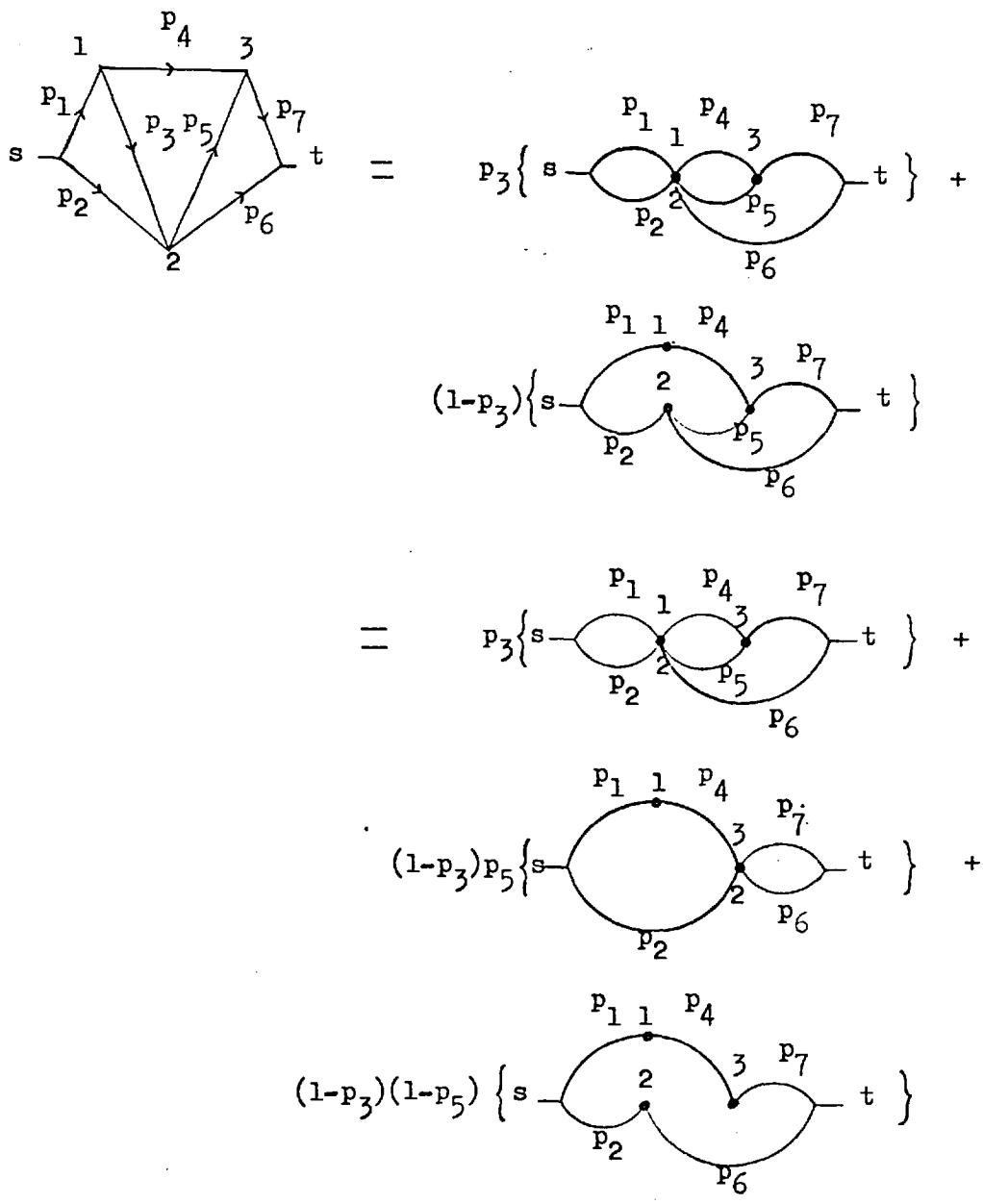


Figure 2. Application of the factoring theorem

when nodes as well as branches are unreliable.

3.5 Tie-set enumeration

One of the steps in the event enumeration technique suggests that a simplest way to find the reliability expression is to locate the successful paths at the outset. Then each successful path forms a favourable event and the union of these events gives the reliability expression.

3.5.1 Problem formulation

A path or tie-set is a group of branches which form a connection between input and output when traversed in a stated direction. A minimal path is a path which contains a minimum number of elements. If no node is traversed more than once in tracing a path, the path is minimal.

The probability of successful communication between nodes s and t is defined as the probability of successful operation of all elements in at least one tie-set and is given by

$$\begin{aligned} P_S(s,t) &= \Pr\left\{ \text{at least one path is good} \right\} \\ &= \Pr\left\{ \bigcup_{i=1}^{m} T_i\,(s,t) \right\} \end{aligned} \tag{3.3}$$

where m is the number of tie-sets between nodes s and t and $\bigcup$ denotes the union.

For the oriented graph in Fig. 3 there are 5 tie-sets between nodes s and t given by the following combinations of elements $T_1 = (e_1e_4e_7)$, $T_2 = (e_1e_3e_5e_7)$, $T_3 = (e_1e_3e_6)$, $T_4 = (e_2e_5e_7)$ and $T_5 = (e_2e_6)$. Thus $P_S(s,t)$ is given by Eqn. (3.4) where the + indicates the logic sum or union.

$$P_S(s,t) = Pr\{T_1 + T_2 + T_3 + T_4 + T_5\}$$

$$= Pr\{e_1e_4e_7 + e_1e_3e_5e_7 + e_1e_3e_6 + e_2e_5e_7 + e_2e_6\} \tag{3.4}$$

Alternatively, the probability $P_f(s,t)$ of failure between any pair of nodes (s,t) is given by

$$P_f(s,t) = Pr\{\text{ all paths have a failure }\}$$

$$= Pr\{\bigcap_{i=1}^{m} \overline{T}_i(s,t)\} \tag{3.5}$$

where $\bigcap$ indicates the intersection.

3.5.2 Network representation

As a path is an ordered sequence of branches between the source and sink terminals, each tie-set can be associated with a subnetwork of a series arrangement of the elements of the ith tie-set. Since the communication between s and t is successful if and only if all of the elements in at least one tie-set work the network may be represented as a parallel arrangement of the minimal tie-sets. An example is shown in Fig. 4 for the network in Fig. 3. The probability of a failure between nodes s and t of Fig. 3 is given by the probability that all 5 tie-sets have a failure, that is, for each set at least one element fails.

3.5.3 Methods of solution

Algorithms based on direct expansion of Eqn. (3.3) involve the probability of the union of m-paths which are generally not mutually

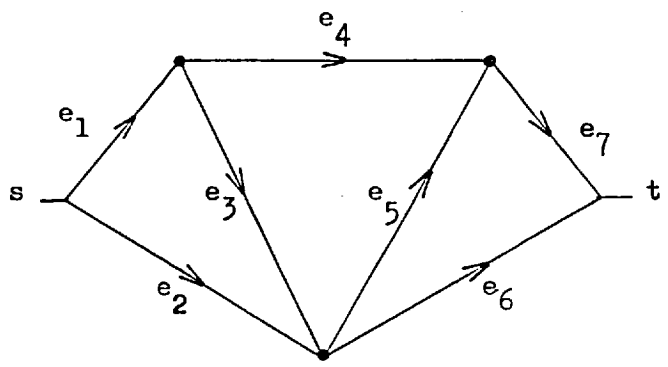


Figure 3. An oriented graph

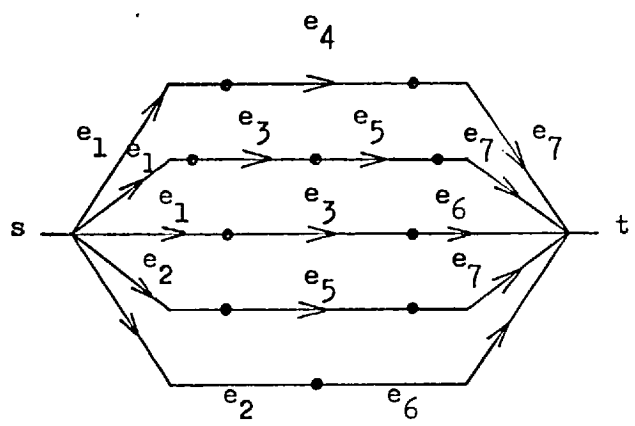


Figure 4. Minimal tie-sets between nodes s and t of Fig. 3

exclusive. Such evaluation[18] contains 2^m-1 terms. The probability of the intersection of m events as given by Eqn. (3.5) is expressed as the joint product of one independent probability and (m-1) dependent probabilities. Evaluation of dependent probabilities can become quite cumbersome; e.g., in the evaluation of Eqn. (3.3) there are $2^m-(m+1)$ dependent probabilities involved. Okada[29] has shown that the number of s-t minimal paths in a connected graph may be as large as 2^{b-n+1}. Systems of moderate complexity are very difficult to solve even with modern digital computers.

Improved methods have been developed which reduce the number of terms of the reliability expression $P_S(s,t)$ either using Boolean algebra[25],[26] to generate mutually exclusive events from the tie-sets or by a special term-reduction technique based on zero-valued terms of conditional probabilities[27]. A few of the more important results of these methods are now presented.

Brown[25] solves the problem by generating the reliability expression from the Boolean algebra transmission function constructed by enumerating all the success between the input and output terminals of the system. Instead of generating the $2^m - 1$ events required by the event-enumeration technique he generates directly only the successful events from the binary representation of the success paths. The basic principle that must not be violated when translating and combining the elementary Boolean events into the probability function is never to include a given elementary event more than once. The algorithm is easily programmed to obtain all mutually exclusive events but it has two drawbacks for large systems:

1) It assumes that the task of storage and enumeration of the various paths of system success is nominal compared to that of obtaining the reliability function. However, the 16-element redundant system

of Fig. 5 has 55 success paths which are increased to 418 if an element is added between each pair of connected nodes. The resulting system has just 26 elements.

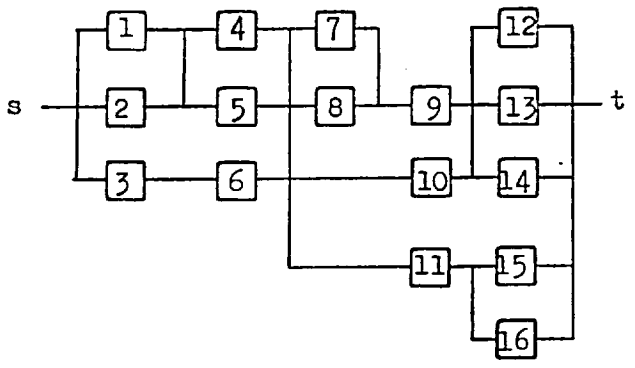


Figure 5. A 16 - element system

To solve the system of Fig. 5 with this algorithm requires the generation of 163840 binary numbers stored in a matrix (163840 x 16) which is extremely large. To overcome this problem it is advantageous before proceeding to find the success paths to combine all parallel elements between common nodes i and j using Boolean algebra rules and replace them by an equivalent link[21]. With this process the system of Fig.5 is reduced to the system in Fig. 6 which has only 10 elements and 6 success paths instead of 16 and 55 respectively.

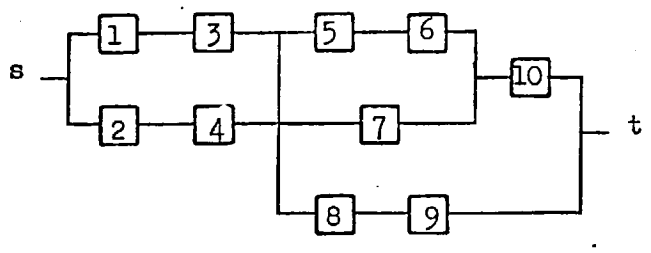


Figure 6. Equivalent to system in Fig. 5

To solve the system of Fig. 6 only 320 binary numbers are required stored in a matrix of dimension (320 x 10) compared with a matrix (163840 x 16) for Fig. 5. Having reduced the redundant elements the success paths can be found by any of the several path-finding algorithms now available in the literature[28].

2) A second drawback for large systems is that each transformed term is compared with each succeeding binary number up to 2^n-1 where n is the number of elements. The number of binary number comparison doubles with each additional component. For instance, a 10-element system would require 1024 binary number to be generated and evaluated by the computer, whereas an 11-element system would require 2048, and a 15-element system 32768. The problem becomes very serious for systems of 20 to 30 elements. The inefficient comparison digit by digit of each number with the succeeding numbers up to 2^n-1 is overcome by generating a restricted set of numbers directly from the binary representation of each success path taking advantage of the nature of binary numbers. This procedure reduces the 1024 binary numbers required for the 10-element system in Fig. 6 to 320 binary numbers. However an upper bound for the binary number comparisons is given by $(n_b-1)!$ where n_b is the total binary numbers required by the algorithm.

Reductions greater than 60% of the number of terms are obtained using a simplification procedure based on the theorem $p_A \cdot q_B + p_A \cdot p_B = p_A$. In general higher reductions (up to 80%) are obtained when an ascendent or descendent ordering is applied to the terms of the partial reliability table before applying the simplification procedure. The use of an ascendent or descendent ordering does not add significantly to the

execution time. Table II shows the results for several networks (see Appendix 1) with different number of elements and success paths. The times do not include success path generation.

Fratta and Montanari[26] use the same approach of reducing the Boolean algebra transmission function $F = \bigcup_{i=1}^{m} T_i$ to a sum of disjoint products by performing simple Boolean algebraic operations on the initial form obtained from the set of all tie-sets. For each term of F they perform the operation $F \leftarrow \overline{T}_i \cdot (F - T_i)$ and reduce $P_S(s,t)$ to a sum of mutually exclusive products. The approach uses the well-known identity $\overline{e_1 e_2} = \overline{e_1} + \overline{e_2}$ for computing F and then uses distributivity. The method has no space limitations for its implementation but is time limited even for systems having less than 20 paths due to the large number of fundamental products of F and the number of symbolic multiplications required. They recognised the deficiency mentioned in Brown's method by incorporating a series-parallel reduction procedure and a path-finding algorithm as the first and second steps in the implementation of their algorithm.

Lin et al[27] present a modification for the direct expansion of Eqn. (3.3) expressing the joint probabilities of success paths as conditional probabilities. For example, $P_S(s,t)$ in a probabilistic graph having only 2 paths (P_1, P_2) from s to t is given by

$$P_S(s,t) = p_1 + p_2 - p_{1,2}$$

$$= p_1 + q_{1(2)}\, p_2$$

where $q_{1(2)} = 1 - p_{1(2)}$, $p_1(2)$ being the probability that p_1 is good under the condition that P_2 is good and p_1 and p_2 are the success probabilities of paths P_1 and P_2 respectively. For the general case of m paths, the explicit expression for $P_S(s,t)$ by the present method has 2^{m-1} terms

TABLE II

COMPARISONS OF DIFFERENT ORDERINGS OF THE BINARY TERMS IN THE PARTIAL RELIABILITY TABLE

Network (App 1)	Number of Success Paths	Number of elements	PARTIAL RELIABILITY TABLE			ARBITRARY ORDERING			ASCENDENT ORDERING			DESCENDENT ORDERING		
			Terms	Multipli-cations	Time (sec)	Terms	Multipli-cations	Time (sec)	Terms	Multipl-ications	Time (sec)	Terms	Multipl-cations	Time (sec)
1	3	4	8	31	0.450	3	11	0.482	3	11	0.485	3	11	0.484
2	3	5	17	84	0.516	5	24	0.537	5	24	0.543	4	19	0.537
3	4	6	19	113	0.609	5	29	0.634	5	29	0.646	5	29	0.648
4	6	7	58	405	1.096	13	90	1.351	8	55	1.383	10	69	1.408
5	8	7	61	426	1.345	10	69	1.388	10	69	1.414	10	69	1.482
6	8	7	82	573	1.854	14	97	1.885	9	62	1.977	11	76	1.996

which is about half of the number from the direct expansion of Eqn. (3.3). Fortunately the q-factors usually produce zero-valued terms which do not need to be generated thus reducing the actual number of terms. The method for systems having fewer than 20 paths has shown[27] to reduce the number of terms of the reliability expression up to the order of two or three hundred instead of the $2^{20}-1$ terms given by direct expansion of the probability of a union.

3.6 Cut-set enumeration

One way to alleviate the computational difficulties of the event enumeration method is by using a technique which requires the enumeration of a much smaller set of system states. These states are called minimal cut-sets.

3.6.1 Problem formulation

A cut-set of a graph can be defined as a group of elements that, when they fail, the system fails regardless of the condition of the other elements in the system. A cut-set is minimal when there is no proper sub-set of its elements whose failure alone will cause the system to fail. The elimination of any element of a minimal cut no longer makes it a cut. This is because a nonminimal cut corresponds to more elements than are required to cause system failure. The minimal cut-sets are the group of distinct cut-sets of the system containing a minimal number of elements. All system failures can be represented by the removal of at least one minimal cut-set from the graph.

Consequently, a cut-set with respect to a specific pair of nodes s and t in a connected network, called an s-t cut is such that its removal destroys all paths between nodes s and t. Therefore, the probability of failure between any pair of nodes s and t is given by the probability

that at least one minimal cut-set fails. If C_j represents the jth minimal cutset and $\overline{C}_j$ the failure of the jth cut-set the preceding statement can be expressed as follows:

$$P_f(s,t) = \Pr\{\text{ at least one cut-set occurs }\}$$
$$= \Pr \bigcup_{j=1}^{m} \overline{C}_j\,(s,t) \qquad (3.6)$$

where m is the number of minimal cut-sets. Similar to the path-enumeration method, $P_f(s,t)$ is not readily calculated from Eqn. (3.6) because the $\overline{C}_j(s,t)$ are not mutually exclusive events. On the other hand, the terminal-pair reliability is given by

$$P_S(s,t) = 1 - P_f(s,t) \qquad (3.7)$$

or alternatively

$$P_S(s,t) = \Pr\left\{\begin{array}{l}\text{all cut-sets are good, i.e. at least}\\ \text{one element of the set is working}\end{array}\right\}$$
$$= \Pr\left\{\bigcap_{j=1}^{m} C_j\right\} \qquad (3.8)$$

3.6.2 Network representation

The system may have a large number of cuts and a particular element may be in more than one of them. An example is shown in Fig. 7. As long as any success path exists between terminals 1 and 4 of the network, the network is said to be successful. An element failure opens the path between the two terminals of the component. The cuts of this network are listed in Table III and the numbers of the elements describe the cuts. The failure of any one of these cuts will cause the network of Fig. 7 to fail. The minimal cut-sets between nodes 1 and 4 of Fig. 7 are listed in Table IV along with their probability of occurrence. q_i is the probability that the ith element fails. Note

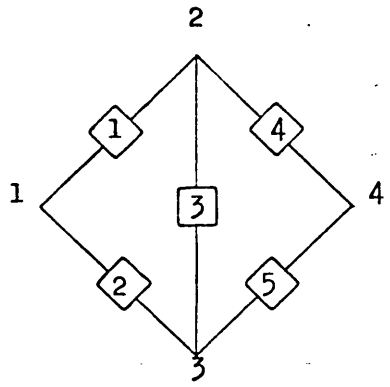


Figure 7. Bridge network

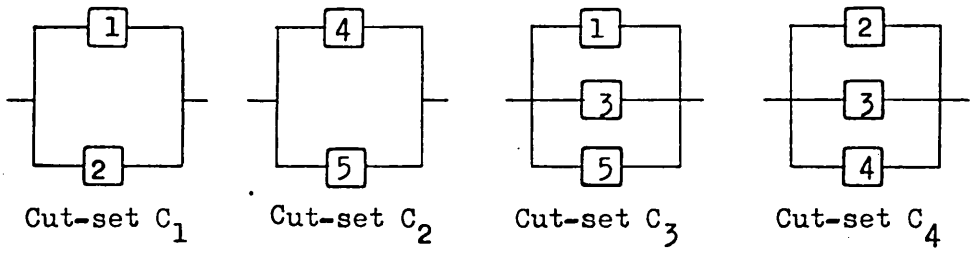


Figure 8. Parallel arrangement of elements in the minimal cut-sets of Fig. 7

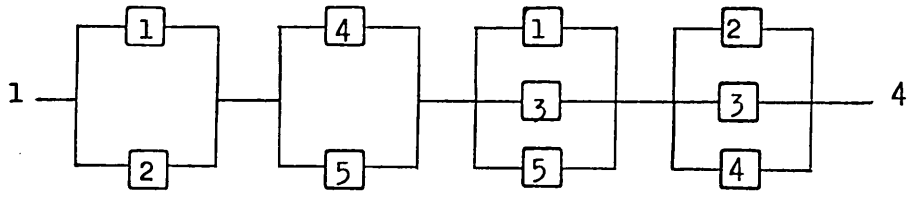


Figure 9. Series arrangement of the minimal cut-sets of Fig. 7

that cut 2 of Table III is not a minimal cut since it is contained in cut 1 which is a minimal cut.

Each minimal cut-set C_j can be associated with a subnetwork of a parallel arrangement of the elements of the jth cut-set. These subnetworks for the network of Fig. 7 are shown in Fig. 8. Since the network fails if, and only if, at least one of the minimal cut-set fails, the network of Fig. 7 may be represented as a series arrangement of the minimal cut subnetworks of Fig. 8 as is shown in Fig. 9.

3.6.3 Methods of solution

Direct expansion of Eqn. (3.6) involves, as in the path-enumeration method, the probability of a union which contains 2^m-1 terms. Bellmore and Jensen[33] have shown that the number of s-t minimal cut-sets in a n-node network is bounded by 2^{n-2}. The computation for systems of moderate complexity is clearly infeasible even with large computers. This situation has led to the development of techniques either to generate mutually exclusive sets of cutting states[34] or to provide lower and upper bounds approximation to system reliability. Shooman[18] and Barlow and Proschan[35],[36] provide good mathematical background material and give many references to previous work in the area of reliability approximation.

Nelson et al[30],[31] provides an algorithm for system reliability with unidirectional elements based on the concepts of success paths and cut-sets. Jensen and Bellmore[32],[33] provide an algorithm for determining the reliability of a complex network with bidirectional elements. A single application of the algorithms of both papers gives the minimal cut-sets between two nodes of a network. A few of the more important results of these papers will now be presented.

TABLE III

Cuts between nodes 1 and 4 of Fig. 7

Cut	Elements in Cut
1	1,2
2	1,2,3
3	1,2,4
4	1,2,5
5	1,2,3,4
6	1,2,3,5
7	1,2,4,5
8	1,2,3,4,5
9	4,5
10	3,4,5
11	2,4,5
12	1,4,5
13	2,3,4,5
14	1,3,4,5
15	1,3,5
16	2,3,4

TABLE IV

Minimal cuts between nodes 1 and 4 of Fig. 7

Minimal Cut	Elements in Cut	Probability of failure of minimal cut
C1	1,2	q_1q_2
C2	4,5	q_4q_5
C3	1,3,5	$q_1q_3q_5$
C4	2,3,4	$q_2q_3q_4$

The Nelson et al[30] approach provides bounds for system reliability based on the concepts of success paths and cut-sets. A listing of the elements of the system, their predecessors, and the probability of successful operation of each element are the inputs of the computer program[31]. The outputs are the success paths, the cut-sets, and a series of upper and lower reliability bounds; these bounds are based on the inclusion-exclusion method of section 3.9 The algorithm for determining the cuts from the success paths is based on Boolean logic which is simple to understand. The algorithm is easily implemented for computer analysis of system reliability but has two distinct drawbacks in practice. First, it needs to determine and store the success paths. As mentioned before if we add one element between each pair of connected nodes in Fig. 5, we have a system with 418 success paths instead of the original 55 by just increasing the network from 16 to 26 elements. The use of the success paths as they are determined by Nelson et al requires operation with a matrix of dimension (418 x 26). Second, to generate the cuts of order k ($k > 1$) in a network of n elements it is necessary to cover all the k logic sums or unions between the n column vectors of the path matrix i.e., ${}_kC_n$, and this is repeated until all "possible" cuts of order 1,2,...,n have been exhausted. Thus, it is necessary to calculate

$$S = \sum_{k=2}^{n} {}_kC_n = 2^n - (n + 1)$$

logic sums in this procedure. (e.g. $n = 10$ $S = 1013$; $n = 20$ $S = 1{,}048{,}555$). Besides, after a possible cut of order k is identified, it needs to be checked against all cuts of lower order using Boolean algebra for intersection (AND operation) i.e. for a network of m cuts, $m(m+1)/2$ checks must be done at the end of the minimal cut-set generation (e.g. $m = 100$, 5000 checks). This is a serious problem to handle if we consider that,

depending on the configuration of the network, the number of minimal cuts between two nodes (IN and OUT nodes) varies from n-1 to 2^{n-2}. There are n-1 cuts if the network is a simple chain and the IN and OUT nodes are the ends of the chain and 2^{n-2} cuts if there is a component between each pair of nodes. For instance a network of 12 nodes will have, in the second case, 66 elements and 1024 minimal cuts.

Jensen and Bellmore[32] provide an algorithm for determining the reliability of a complex system in which all components have two terminals and are bidirectional i.e. a path may traverse it in either direction. For a n-node network, source(s) and sink(t) nodes are numbered 1 and n respectively, the numbering of all other nodes is arbitrary. They solve the problem of generating the set of minimal cuts as the problem of generating the set of two-part partitions of a set N into two mutually exclusive subsets X and $\overline{X}$. The subset X must define a connected subnetwork that includes the source node s. The subset $\overline{X}$ must define a connected subnetwork that includes the sink node t. The minimal cut corresponding to a two-part partition is the set of components with one terminal in X and one terminal in $\overline{X}$. The absence of an unscanned node signals termination of the algorithm. They use a lower bound approximation[33] based on the disjoint property and their algorithm is highly efficient for networks of single input-output nodes (source and sink nodes) i.e. finding the minimal cut-sets between source and sink nodes in networks of two terminals.

Hansler et al[34] have developed a procedure to iteratively calculate a minimal set of mutually exclusive events containg all minimal cut-sets and summing the probabilities of these events produces the probability of service interruption $P_f(s,t)$ between the specified pair of nodes. The procedure starts with the minimal cut-set consisting of

the branches incident to node t. Subsequently, these branches are reconnected in all combinations and we then cut the minimal set of branches adjacent to those that lie on a path to node s. The branch replacements are repeated until the set of branches connected to node s are reached. The procedure has been found to be particularly effective for graphs of small diameter. Furthermore, a very small amount of storage is required since each event is generated from the previous one but due to the inherent computational complexity of the problem, the computation time grows exponentially with the system size.

3.7 Selection of the method

The event enumeration method can be used for all systems but considering that the number of events is equal to the total number of combinations of n units i.e., 2^n the method is impractical except for extremely simple systems.

Breaking down the network structure into a series of elementary ones by an iterative use of the factoring theorem is efficient and simple when the system has few bridging elements. For highly non series-parallel networks the labour involved increases rapidly. Further the technique is not applicable when nodes as well as branches are unreliable.

For the general case of complex networks (non series-parallel) with unreliable branches and nodes the path and cut-set methods are the most practical for system reliability evaluation. A simple set of comparative rules which can help to determine whether path or cut-set enumeration should be used can be formulated as follows:

1. In any network of b branches and n nodes, the order of the number of s-t cut-sets is 2^{n-2} whereas the order of the number of s-t paths is 2^{b-n+1}. For networks having nodes of

average degree (number of incident branches) greater than four, $b > 2n$ and $2^{b-n+1} > 2^{n-2}$. Consequently such networks have a larger number of paths than cut-sets and the computation is simpler using cut-set enumeration.

2. In both cases the terminal reliability measure is the probability of the union of the events corresponding to the existence either of the paths or of the cut-sets. These events are not disjoint and the complete expansion of the union becomes a hopeless task. Several approximations have been suggested (see section 3.8) to provide upper and lower bounds on system reliability using the path and cut-sets methods. The bounds based on the cut-sets are best in the high reliability region, and those based on the tie-sets are best in the low reliability region.

If the path method is used, computations can be simplified if parallel elements are reduced to a single equivalent element. Series branches need not be combined. If the cut-set method is used, series branch reduction simplifies the computation and parallel links need not be combined.

3.8 Approximations for reliability evaluation

Exact reliability analysis using the path or cut-sets approaches involve the complete expansion of the probability of a union which contains 2^m-1 terms. One way to alleviate this difficulty is to obtain approximate equations which provide bounds on the system reliability, making feasible the reliability evaluation of large networks. Two approximations which are useful in simplifying the computations of

Eqns. (3.3) and (3.6) are discussed in this section and a sample calculation is made for $P_S(1,4)$ in Fig. 7 assuming all components have equal reliability.

3.8.1 Mutually exclusive approximation

If the events are mutually exclusive, i.e., possess the disjoint property, the probability of a union is greatly simplified. Messinger and Shooman[39] have shown that for any set of events A, B, C the disjoint approximation gives an upper bound

$$\Pr(A \cup B \cup C) \leqslant \Pr(A) + \Pr(B) + \Pr(c)$$

Thus if the tie-sets are assumed to be mutually exclusive an upper bound R_U can be written for $P_S(s,t)$ as

$$P_S(s,t) \leqslant R_U = \Pr\left\{ \sum_{i=1}^{m} T_i(s,t) \right\}$$

$$= \Pr\left\{ \sum_{i=1}^{m} \left(\prod_{j \varepsilon T_i} p_j \right) \right\} \qquad (3.9)$$

Similarly from Eqns. (3.6) and (3.7) of the cut-set approach a lower bound R_L for $P_S(s,t)$ is given by

$$P_S(s,t) \geqslant R_L = 1 - \Pr\left\{ \sum_{j=1}^{m} \overline{C}_j(s,t) \right\}$$

$$= 1 - \Pr\left\{ \sum_{j=1}^{m} \left[\prod_{i \varepsilon C_j} (1 - p_i) \right] \right\} \qquad (3.10)$$

Table V shows these bounds for P_S between nodes 1 and 4 in Fig. 7. The upper bound from the tie-sets becomes a good approximation in the low reliability region and the lower bound from the cut-sets is good in the high reliability region.

3.8.2 Independence approximation

Esary and Proschan[40] have shown that a computation of reliability representing the system as a set of m independent minimal paths acting in parallel gives an upper bound on actual system reliability. This upper bound is given by

$$P_S(s,t) \leqslant R_U = 1 - \prod_{i=1}^{m} (1 - \prod_{j \varepsilon T_i} p_j) \qquad (3.11)$$

Similarly they proved that a reliability calculation considering the system as a set of m independently operating minimal cuts acting in series gives a lower bound on actual system reliability. This lower bound is obtained from

$$P_S(s,t) \geqslant R_L = \prod_{i=1}^{m} \left\{ 1 - \prod_{j \varepsilon C_i} (1 - p_j) \right\} \qquad (3.12)$$

Table V presents these bounds for the probability of successful communication between nodes 1 and 4 in Fig. 7. Similarly to the results given by the disjoint property the bound from the tie-sets is a good approximation in the low reliability region while the bound from the cut-sets becomes good in the high reliability region. Note that the bounds calculated from the independence approximation are sharper than the ones from the mutually exclusive approximation.

3.9 The inclusion-exclusion method

Let $\overline{C}_j$ describe the event that all components in minimal cut-set C_j fail. Then system failures correspond to the event $\bigcup_{j=1}^{m} \overline{C}_j$ if the system has m minimal cut-sets and the system reliability is given by Eqn. (3.7).

Let

$$S_k = \sum_{1 \leqslant i_1 < i_2 \ldots < i_k \leqslant m} \Pr \left\{ \overline{C}_{i_1} \cap C_{i_2} \cap \ldots \cap \overline{C}_{i_k} \right\} \qquad (3.13)$$

TABLE V

COMPARISON OF BOUNDS FROM THE DISJOINT AND INDEPENDENCE APPROXIMATIONS

P_S (1,4) IN FIG. 7

Component reliability p	Disjoint approximation		Independence approximation		Exact reliability $P_S(s,t)$
	Lower bound Cut-sets	Upper bound Tie-sets	Lower bound Cut-sets	Upper bound Tie-sets	
0.10	0.00000	0.02200	0.00265	0.02186	0.02152
0.20	0.00000	0.09600	0.03086	0.09309	0.08864
0.30	0.00000	0.23400	0.11227	0.21601	0.19836
0.40	0.00000	0.44800	0.25176	0.38183	0.34048
0.50	0.25000	0.75000	0.43066	0.56934	0.50000
0.60	0.55200	1.00000	0.61817	0.74824	0.65952
0.70	0.76600	1.00000	0.78399	0.88773	0.80164
0.80	0.90400	1.00000	0.90691	0.96914	0.91136
0.90	0.97800	1.00000	0.97814	0.99735	0.97848
0.92	0.98618	1.00000	0.98623	0.99884	0.98637
0.94	0.99237	1.00000	0.99238	0.99961	0.99243
0.96	0.99667	1.00000	0.99667	0.99992	0.99668
0.98	0.99918	1.00000	0.99918	0.99999	0.99919

By the inclusion-exclusion principle[41] Eqn. (3.7) can be written as

$$P_S(s,t) = 1 - \sum_{k=1}^{m} (-1)^{k-1} S_k \tag{3.14}$$

and bounds on the system reliability are expressed as

$$P_S(s,t) \geqslant R_{L1} = 1 - S_1 \tag{3.15a}$$

$$P_S(s,t) \leqslant R_{U1} = 1 - S_1 + S_2 \tag{3.15b}$$

$$P_S(s,t) \geqslant R_{L2} = 1 - S_1 + S_2 - S_3 \tag{3.15c}$$

$$P_S(s,t) \leqslant R_{U2} = 1 - S_1 + S_2 - S_3 + S_4 \tag{3.15d}$$

and so on. Thus using the minimal cut-set approach the inclusion-exclusion method provides successive lower and upper bounds on system reliability which converge to the exact system reliability. Table VI shows these bounds using four terms approximation for the probability of successful communication between nodes 1 and 4 of Fig. 7. In this simple example the bounds save no computation as a four term expansion gives the exact probability of terminal-pair success. In practice it may be necessary to calculate only a few S_k's to obtain a close approximation and the computations can be stopped when the margin between two successive bounds becomes negligible. The bounds from the minimal cut-sets as shown in Table VI are good in the high reliability region.

Let T_i describe the event that all components in minimal path T_i work. Then system success corresponds to the event $\bigcup_{i=1}^{m} T_i$ if the system has m minimal tie-sets and the system reliability is given by Eqn. (3.3).

Defining

$$S_k' = \sum_{1 \leqslant i_1 < i_2 < \cdots < i_k \leqslant m} \Pr\left\{ T_{i_1} \cap T_{i_2} \cap \cdots \cap T_{i_k} \right\} \tag{3.16}$$

the bounds on system reliability given by the inclusion-exclusion principle are

$$P_S(s,t) \leqslant R_{U1}' = S_1' \tag{3.17a}$$

$$P_S(s,t) \geqslant R_{L1}' = S_1' - S_2' \tag{3.17b}$$

$$P_S(s,t) \leqslant R_{U2}' = S_1' - S_2' + S_3' \tag{3.17c}$$

$$P_S(s,t) \geqslant R_{L2}' = S_1' - S_2' + S_3' - S_4' \tag{3.17d}$$

and so on. Thus using the minimal tie-set approach the inclusion-exclusion method provides successive upper and lower bounds on system reliability which also converge to the exact system reliability. Table VI presents these bounds for $P_S(1,4)$ in Fig. 7 using a four term expansion. These bounds are a good approximation in the low reliability region. The bounds on system reliability given by Eqns. (3.15) and (3.17) are calculated simply only if statistical independence is assumed between components. This assumption has been used for the sample calculation in Table VI.

3.10 System reliability indices

During the operating period of a system, components may be deemed non-repairable or repairable. The non-repairable case is much the easier, but for digital protection systems which are intended to operate for 20 years or so, it is evident that all detectable failures will be

TABLE VI

COMPARISON OF BOUNDS USING THE INCLUSION-EXCLUSION APPROXIMATION

P_S (1,4) IN FIG. 7

Component reliability p	Minimal cut-sets		Minimal tie-sets		Exact reliability P_S (s,t)
	Lower bound	Upper bound	Lower bound	Upper bound	
0.10	0.00000	0.02152	0.02152	0.02153	0.02152
0.20	0.00000	0.08864	0.08864	0.08896	0.08864
0.30	0.03029	0.19836	0.19836	0.20079	0.19836
0.40	0.26272	0.34048	0.34048	0.35072	0.34048
0.50	0.46875	0.50000	0.50000	0.53125	0.50000
0.60	0.64928	0.65952	0.65952	0.73728	0.65952
0.70	0.79921	0.80164	0.80164	0.96971	0.80164
0.80	0.91104	0.91136	0.91136	1.00000	0.91136
0.90	0.97847	0.97848	0.97848	1.00000	0.97848
0.92	0.98637	0.98637	0.98637	1.00000	0.98637
0.94	0.99243	0.99243	0.99243	1.00000	0.99243
0.96	0.99668	0.99668	0.99668	1.00000	0.99668
0.98	0.99919	0.99919	0.99919	1.00000	0.99919

repaired. Thus, for the reliability evaluation of digital protection systems we will assume that the system has reached a steady-state condition of failure and repair. The inherent high reliability achieved in digital hardware suggests that the use of the minimal cut-set approach will provide the best bounds on system reliability.

Two key indices in system reliability evaluation are the probability that the system has failed and the frequency of system failure. The former is obtained from Eqn. (3.6) and the frequency of failure[16] for an m-cuts system is given by

$$\begin{aligned} f_f = {} & \Pr\{\overline{C}_1\}\,\mu_1 + \Pr\{\overline{C}_2\}\,\mu_2 + \dots + \Pr\{\overline{C}_m\}\,\overline{\mu}_m \\ & - \Pr\{\overline{C}_1 \cap \overline{C}_2\}\,\overline{\mu}_{1+2} - \Pr\{\overline{C}_1 \cap \overline{C}_3\}\,\overline{\mu}_{1+3} - \dots \\ & \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ & + (-1)^{m-1} \Pr\{\overline{C}_1 \cap \overline{C}_2 \cap \dots \cap \overline{C}_m\}\,\overline{\mu}_{1+2+\dots+m} \end{aligned} \tag{3.18}$$

where $\overline{\mu}_{j+k+l+\dots}$ represents the sum of μ_i over all $i \in C_j \cup C_k \cup C_l \dots$, i.e., the sum of the repair rates of the components which belong to any or all the cut-sets C_j, C_k, C_l, ...

If λ_j and μ_j represent the failure and repair rates respectively of the jth component then the steady-state probability of the j-component being failed is

$$P_{jd} = \frac{\lambda_j}{\lambda_j + \mu_j} \tag{3.19}$$

and the probability of the system being in the state corresponding to the failure of the elements of the minimal cut-set C_i is

$$\Pr\{\overline{C}_i\} = \prod_{j \in C_i} P_{jd} \tag{3.20}$$

Once the probability P_f that the system is failed and the frequency of system failure f_f has been calculated other quantities of interest can be obtained as follows:

Availability	$A = 1 - P_f$	(3.21)
Mean cycle time	$T = 1/f_f$	(3.22)
Mean up time	$U = (1 - P_f)/f_f$	(3.23)
Mean down time	$D = P_f/f_f$	(3.24)

3.11 Final comments

Since each component is assumed to be in either of two states, working or failed, the task of determining the exact system reliability is computationally unfeasible for large systems because the number of possible states in 2^n for n-element networks.

Two approximations analysed suggest lower and upper bounds to system reliability that requires the enumeration of a much smaller set of system states. The analysis is based on the concepts of minimal cut-sets and minimal paths. These terms refer to the effects of element failure or success on operation of the network.

From the analysis and the sample calculations performed it can be concluded that the minimal cut-set approach is more attractive and efficient than the tie-set method for the reliability analysis of complex networks with highly reliable components. This is the case of digital protection systems.

CHAPTER IV

MINIMAL CUT-SET ALGORITHMS

4.1 Introduction

Digital hardware configurations are integrated by highly reliable elements. In chapter III it was shown that the minimal cut-set approximation gives bounds which are good in the high reliability region. The importance of any algorithm based on the minimal cut-set approximation is the procedure used to enumerate the set of minimal cuts. This chapter presents new algorithms for the minimal cut-set generation in networks with directed (1-way), undirected (2-way) or both types of elements. The case of both unreliable branches and nodes is considered.

4.2 Basic assumptions and definitions

To determine the reliability of a system, the system may be represented either by a reliability graph or by a reliability block diagram as illustrated in Figs. 1 and 2 respectively. A block in the reliability diagram will be called an element and the whole assembly will be called a network.

The reliability analysis treated in this chapter is based on the concepts of minimal cut-sets and coherent system. These terms refer to the effects of element failures on operation of the network.

A coherent system[40] is defined by the following four conditions:

1) When a group of elements in the system has failed, causing the system to be failed, the occurrence of any additional failure or failures will not return the system to a successful condition.

2) When a group of elements in the system is successful and the system is successful, the system will not fail if some of the failed components are returned to the successful condition.

3) When all the elements in the system are successful the network is successful.

4) When all the elements in the system have failed the system has failed.

Since much of the study of system reliability is based on graph theory[42] basic concepts and definitions which are relevant to the present study are summarized in the following paragraphs.

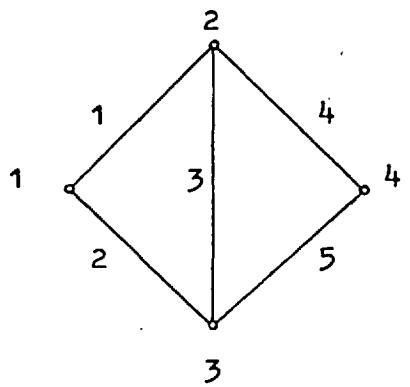


Figure 1. Reliability graph No. 1

A graph G(N,E) consists of a set N of n nodes and a set E of b branches or edges. The number of edges incident on a node is called the degree $d(n_i)$ of node n_i. A node having no incident edge is called an isolated node and a node of degree one is called a pendant node or an end node. In the definition of a graph G = (N,E), it is possible for the edge set E to be empty. Such a graph, without

any edges, is called a null graph. Although the set E may be empty, the node set N must not be empty i.e., by definition a graph must have at least one node. A graph in which there exists an edge between every pair of nodes is called a complete graph.

A graph g is called a subgraph of G if all nodes and edges of g are in G and all the edges in g have the same node terminals as in G. Two (or more) subgraphs g_1 and g_2 of a graph G are edge disjoint if g_1 and g_2 do not have any edges in common. The union of two graphs $G_1 = (N_1,E_1)$ and $G_2 = (N_2,E_2)$ is another graph $G_3 = (N_3,E_3)$ whose node set $N_3 = N_1 \cup N_2$ and the edge set $E_3 = E_1 \cup E_2$. The intersection $G_1 \cap G_2$ of graphs G_1 and G_2 is a graph consisting only of those nodes and edges that are common to G_1 and G_2. The ring sum of two graphs G_1 and G_2, $G_1 \oplus G_2$, is a graph consisting of the node set $N_1 \cup N_2$ and of edges that are either in G_1 or G_2, but not in both.

It is necessary to distinguish between two types of graphs, directed and undirected. In a directed graph some of the elements may not be able to allow flow in both directions. On the other hand, all elements in undirected graphs are two-way branches. An undirected graph is said to be connected if there is at least one path between every pair of nodes. Otherwise it is disconnected. A null graph of a single node is defined to be a connected graph. A digraph (di-rected graph) is said to be connected if its corresponding undirected graph is connected.

A cut-set in a connected graph G, is a set of edges whose removal from G leaves G disconnected. A cut-set is minimal provided no subset of the removed edges has the same property. For instance, in Fig. 1 the set of edges (1,2) is a minimal cut-set. The set of

edges (1,2,3), on the other hand, is not a minimal cut-set because one of its subset (1,2) is a cut-set. Removal of the edges of any cut-set in a graph G always separates the graph into two connected subgraphs producing a partition of all nodes into two mutually exclusive subsets Y and $\overline{Y}$. A proof of this last statement is given by Jensen and Bellmore[32]. This property of the cut-sets will be used later to introduce the concept of the cut-node incidence vector. Then a cut-set can be seen also as the minimal number of edges with one terminal in Y and the other in $\overline{Y}$ whose removal from G destroys all paths between these two sets of nodes.

Each cut-set of a connected graph G consists of a number of edges (order of the cut-set). The smallest order of the cut-sets is defined as the edge connectivity of G. Similarly, the node connectivity (or simply connectivity) of a graph G is defined as the minimum number of nodes whose removal from G leaves the remaining graph disconnected. A connected graph is said to be separable if its node connectivity is one. All other connected graphs are called non-separable. In a separable graph a node whose removal disconnects the graph is called an articulation point.

With each minimal cut-set MC_i, i = 1,2,...,NMC say, may be associated a binary representation MC_{ij}, j = 1,2,...,NE whose components take the value of 1 if the ith cut contains the jth element of the graph and is 0 otherwise.

The minimal cut-set matrix MCM_{ij} is defined as the array of all the binary numbers MC_{ij} where each row i shows the incidence of all jth elements of the network in the ith minimal cut-set of the network. Similarly each column j shows the incidence of the jth element in all cut-sets of the network. The minimal cut-set matrix

for the reliability graph of Fig. 1 between nodes 1 and 4 is

$$MCM = \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

The four cut-sets between nodes 1 and 4 are thus (1,2), (2,3,4), (1,3,5) and (4,5) as indicated by the 1's in the corresponding positions in rows 1 to 4.

This chapter makes the following assumptions:

1) the reliability networks are coherent networks.
2) each element has two terminals and can be in either of two states, working or failed. There is no standby or switched redundancy.
3) the states of all elements are independent i.e., the failure of one element does not affect the probability of failure of any other.

Other assumptions concerning the type of systems, repairable or nonrepairable, are introduced in the next section.

4.3 Modelling of complex systems by reliability graphs

A complex system will be modelled by a weighted linear graph where the weights of the elements are either the probability of success p_j for the non-repairable component or the steady-state probability of being failed p_{jd} for the repairable case defined through the failure and repair rates of the component. For the case of repairable systems it is also assumed that the system has reached

a steady state condition of failure and repair and that repaired components are as good as new.

Elements are interconnected at nodes of the network and all the inputs in the system are combined into a single node called the reference node. All other nodes are considered as output nodes. For example, in the representation of a power system, inputs correspond to electrical generators, elements to transmission lines, and nodes to busbars. Electric power flows from the generators via the network of transmission lines to the busbars. In the representation of a digital protection system components of the hardware structure are considered the elements of the graph, the electrical junctions represent the nodes and the signal transducers are the inputs. If data validation and estimation routines similar to those described in chapter II are available, missing data can be replaced or estimated from the redundant measurements. In that case it is assumed that only the simultaneous failure of all inputs could cause a system failure. Otherwise the system has to be analysed as an r-out-of-n configuration.

Unidirectional elements are given a direction from IN to OUT terminals. In graphs with only unidirectional elements the signal from one node will pass to other nodes just through those elements which have their IN terminals connected to that node. For graphs with both types of elements, unidirectional and bidirectional, the signal from one node can pass to the other nodes of the network either through any of the bidirectional elements connected to the node or through the unidirectional elements which have their IN terminals connected to that node. A signal from any of the inputs of the system will reach another node through a path which includes at least one of the undirected elements incident to that node or any directed element

with their OUT terminal incident to that node.

All the elements connected to the reference node are directed elements with their IN terminals joined by the reference node. In separable networks with an articulation point n_j all edges incident to n_j which are disconnected from the inputs when n_j is removed are considered directed elements with their IN terminals connected to n_j.

4.4 Cut-node incidence matrix

The minimal cut-sets of the system are defined here as the set of distinct minimal cuts such that when they occur one or more output nodes of the network are disconnected from the reference node. The minimal cut-sets of node j are a subset of the minimal cut-sets of the network such that when at least one cut occurs the node j is disconnected from the reference node. Each one of these cut-sets is called a minimal cut-node CN_j and the group is the minimal cut-node-set MCN_j. To illustrate these concepts tables I and II show the minimal cut-sets and the minimal cut-node-sets of Fig. 1.

Note in table II that the minimal cut-set (1,2) causes the disconnection of all the output nodes of the network. Such a minimal cut-set is called the basic minimal cut-set and contains all the elements connected to the reference node. Note also that the minimal cut-set (1,3,4), (2,3,5) and (4,5) cause the disconnection only of the nodes 2,3 and 4 respectively. Each one of these minimal cut-sets is called the proper minimal cut-node set and contains for each node all the elements incident to the node. The order of this cut-set is equal to the degree of the node. The basic minimal cut-set is a cut-set for all the output nodes of the network, all the other minimal cut-sets of the network will cause the disconnection of maximum of n-1 nodes where n is the total number of output nodes.

TABLE I

Minimal cut-sets of system in Fig. 1

Cut	Elements
1	1,2
2	1,3,5
3	1,3,4
4	2,3,4
5	2,3,5
6	4,5

TABLE II

Minimal cut-node sets of system in Fig. 1

2	3	4
1,2	1,2	1,2
1,3,5	2,3,4	4,5
1,3,4	2,3,5	1,3,5
		2,3,4

Based on these concepts, with each minimal cut-set MC_i $i = 1,2,..,NMC$ of the system, may be associated a binary number CNV_{ij}, $j = 2,3,...,n$ called the cut-node incidence vector whose components take the value of 1 if the ith cut-set causes the disconnection of the jth node and is 0 otherwise.

The cut-node incidence matrix CNM_{ij} is defined as the array of all the binary numbers CNV_{ij} where each row i indicates the incidence of the ith minimal cut in all the output nodes of the network, $j = 2,3,...n$ because reference node is numbered 1. The cut-node incidence matrix for the reliability graph of Fig. 1 is shown below.

$$CNM = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \begin{array}{c} \text{Minimal cut-sets} \\ 1,2 \\ 1,3,5 \\ 1,3,4 \\ 2,3,4 \\ 2,3,5 \\ 4,5 \end{array}$$

The problem of generating the minimal cut-sets for each node in a n output-node network is then the same as the problem of finding the minimal cut-sets between the reference node and each one of the output nodes and defines the incidence of each new minimal cut on all other output nodes. In other words, when we are considering the generation of the cut-sets between the reference node and node k we look also for the incidence or not of each cut-set over all other output nodes in the network. The removal of the components in a minimal cut-set for node k separates the network into exactly two connected subnetworks. One subnetwork includes reference node and j output nodes of the network, the other subnetwork includes node k and $n-(j + 1)$ output nodes. A minimal cut-set for node k is also minimal

cut-set for the n-(j + 1) output nodes of the same connected sub-network.

The generation of the cut-node incidence relationship simultaneously with the minimal cut-set enumeration provides an efficient and systematic method of generating the minimal cut-node sets for the (n -1) output nodes simultaneously in a graph of n nodes when the edge reliabilities are known a priori.

4.5 Nonseparable graphs

The contribution of this section is an algorithmic procedure that generates the set of minimal cuts for nonseparable graphs. The advantage of the algorithm compared with others described in the literature[33],[34] is that when the generation process stops, the minimal cut-sets between the reference node and each output node have been generated all at once.

Each element or edge has two terminals identified as SB and EB and the numbering of nodes is arbitrary except that reference node is numbered 1.

4.5.1 Description of the algorithm[43]

The algorithm is a node-scan procedure which generates the minimal cut-sets between all output nodes and the reference node when (NN - 1) nodes are scanned, NN being the total number of nodes in the network. Then the algorithm provides a node-counter NK and a node-flag NF(j) to count and identify the scanned j-nodes.

Each time a minimal cut-set is generated a binary vector MC_{ij} is constructed to indicate the j-elements that constitute the ith cut-set. Similarly a cut-node vector CNV_{ij} is generated to mark whether or not the ith cut-set is a cut-set for each one of the j output nodes.

As soon as the procedure scans a new node, the proper cut-set of the node is generated with all the edges incident to the new node. The ith proper cut-set is identified by a cut-flag CF(i) equal to 1. All other cut-sets generated when scanning node k are produced by the ring sum of the proper cut-set of node k and each one of the i cut-sets previously generated that have at least one edge in common with the proper cut-set. The number of edges in common is called the index $CI_k(i)$ of the ith cut-set when scanning node k. Each time a ring sum is made the resulting set of edges are checked to see if they are contained in one of the minimal cut-sets generated from the ring sums of the same proper cut-set. If they are contained then this set of edges is not a minimal cut-set. The ring sums of any proper cut-set and each non-zero index cut-set are made in an ordered sequence from the highest to the lowest indices.

Each time a cut-set is found the appropriate cut-node vector is generated using the cut-node vector and the cut-flag of the cut-set which has been combined with the proper cut-set. If the other cut-set is a proper cut-set or has been generated from a proper cut, the new cut-node vector is equal to the old one but will also include the node being scanned. Otherwise the cut-node vector is equal to the old one but will not include the node being scanned.

The algorithmic procedure can be described in 5 steps:

Step 1 : Generate the basic minimal cut MC_{1m} with the set of edges incident to the reference node. Construct a cut-node vector CNV_{1j}, j = 2,3,..NN with all components equal to 1 and mark reference node as scanned, FN(1) = 1. Set a node-counter NK equal to 1 and a cut-flag CF(1) equal to zero. Calculate lower and upper reliability bounds for each node of the network. Go to step 2.

Step 2 : Check the node-counter. If it is less than NN - 1 choose the unscanned node with the lowest number and go to step 3. If it is equal to NN - 1 the algorithm terminates and all minimal cut-node sets have been generated and bounds for each node reliability are available.

Step 3 : Let the node chosen to be denoted as node i ($i > 1$). Find the [k] elements incident to node i. Identify from the minimal cut-sets generated which contain one or more of the k elements and give to each an index $CI_i(p)$ equal to the number of k elements in the p-cut. If at least one of the cut-sets already generated has a non-zero index, go to step 4. If not, advance to next unscanned node in increasing order from node i and repeat step 3.

Step 4 : Generate the proper minimal cut-set of node i with the k elements found in step 3 and a cut-node vector with all elements equal to 0 except for the element corresponding to node i which is equal to 1. Mark the cut as a proper cut by means of a cut-flag equal to 1. From the new minimal cut-set calculate new lower and upper reliability bounds for node i. Go to step 5.

Step 5 : Make the ring sum between the proper cut-set of step 4 and each one of the already generated cuts with a non-zero index identified in step 3. The new set of edges is a minimal cut-set if and only if it is not contained in any of the cuts already generated when scanning node i. This procedure is applied for all cuts with non-zero index, beginning with the cuts of highest indices and continuing with cuts of decreasing index. Construct for each minimal

cut a cut-node vector equal to that of the cut-set combined with the proper cut-set except for the term of the node being scanned which must be equal to 0 or 1 if the old cut-flag is 0 or 1 respectively. Set the new cut-flag equal to the old cut-flag. Each time a minimal cut and its cut-node vector are generated, calculate lower and upper reliability bounds of the affected nodes. Mark node i as scanned FN(i) = 1, increment node-counter and go to step 2.

To illustrate this procedure, a very simple example (Fig. 1) is used to determine the minimal cut-sets for output nodes 2, 3 and 4. Reference node is numbered 1. In the example we reserve m as subindex of the 5 elements (m = 1,2,3,4,5) and j as subindex of the three output nodes (j = 2,3,4).

Step 1 : MC_{1m} = 1 1 0 0 0 (first minimal cut-set)

CNV_{1j} = 1 1 1

FN(1) = 1

CF(1) = 0

NK = 1

Step 2 : NK = 1 go to step 3

Step 3 : i = 2

[k] = 1 0 1 1 0

$CI_2(1)$ = 1

Step 4 : MC_{2m} = 1 0 1 1 0 (second minimal cut-set)

CNV_{2j} = 1 0 0

CF(2) = 1

Step 5 : MC_{3m} = 0 1 1 1 0 (third minimal cut-set)

CNV_{3j} = 0 1 1

CF(3) = 0

FN(2) = 1

NK = 2

Step 2 : NK = 2 go to step 3

Step 3 : i = 3

[k] = 0 1 1 0 1

$CI_3(1)$ = 1

$CI_3(2)$ = 1

$CI_3(3)$ = 2

Step 4 : MC_{4m} = 0 1 1 0 1 (fourth minimal cut-set)

CNV_{4j} = 0 1 0

CF(4) = 1

Step 5 : MC_{5m} = 0 0 0 1 1 (fifth minimal cut-set)

CNV_{5j} = 0 0 1

CF(5) = 0

MC_{6m} = 1 0 1 0 1 (sixth minimal cut-set)

CNV_{6j} = 1 0 1

CF(6) = 0

MC_{7m} = 1 1 0 1 1 ε MC_{5m}

FN(3) = 1

NK = 3

Step 2 : NK = 3 = NN - 1 stop.

Termination of the procedure

A detailed discussion and proof of the algorithm steps is included in section A2.2 of Appendix 2.

4.5.2 Programming considerations

The algorithm is designed to generate the entire minimal cut-sets and cut-node incidence matrices and maintain them in the core memory of the computer. In the computer program implementation core memory space must be set aside to keep all the information concerning both matrices.

A very simple and efficient storage scheme is possible taking advantage of the binary nature of both matrices. The minimal cut-set matrix will be kept as a column vector whose components indicate the number of the columns with value equal to 1 for each cut-set i.e., the number of the edges of the graph that contribute to that cut-set. A second vector will be used to keep the information about the location of the cut-set information within the first vector; its dimension will be (1 x NMC), where NMC is the number of minimal cut-sets.

Example: Let A be the minimal cut-set matrix of the cut-sets between nodes 1 and 8 of the graph in Fig. 3. The number of minimal cut-sets is 10.

$$A = \begin{bmatrix} 1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&1&0&1&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&1&1&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0 \\ 0&0&0&0&0&0&1&1&0&1&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&1&1&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&0&1&1&1&0&1&1 \end{bmatrix}$$

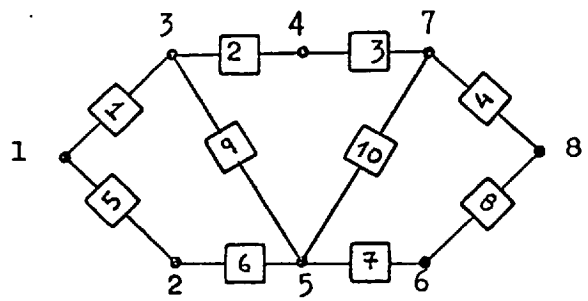


Figure 2. Reliability block diagram No. 1 [32]

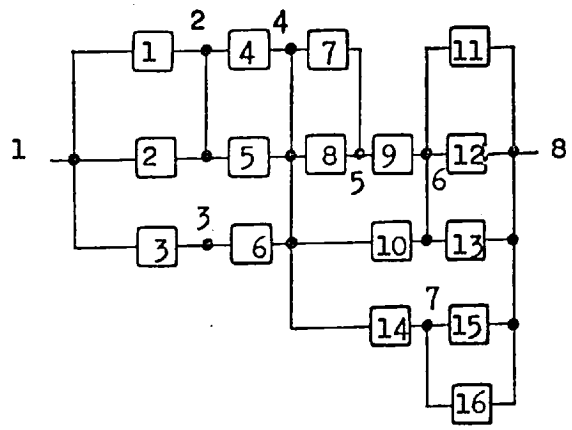


Figure 3. Reliability block diagram No. 2 [30]

Using the storage scheme as described above the information contained in A is kept in a column vector of dimension (37 x 1) as follows:

$$B^T = (1,2,3/3,4,5/1,2,6/4,5,6/7,8,10,14/9,10,14/$$
$$11,12,13,14/7,8,10,15,16/9,10,15,16/11,12,13,15,16)$$

where the cut-sets are shown separated by slashes and the elements in each cut-set separated by commas for clarity. A second column vector C will keep the information of the start point of each cut-set within vector B as follows:

$$C^T = (1,4,7,10,13,17,20,24,29,33)$$

This example shows how matrix A of dimension (10 x 16) is stored in two vectors B(37 x 1) and C(10 x 1) which represents a 70.6% reduction of space requirement.

A similar scheme is used to keep the cut-node incidence matrix but the equivalent vector will identify the nodes affected by each cut-set. Table III shows the percentage space-saving for storage of the minimal cut-sets and cut-node incidence matrices of 4 graphs described in Appendix 1. Graphs 5 and 6 are complete graphs with an edge between every pair of nodes. There are $(n - 1).2^{n-2}$ minimal cut-sets, between the reference node (node 1) and all other nodes, for these graphs. Graphs 1 and 4 correspond to Figs. 2 and 3 taken from references [32] and [30] respectively.

TABLE III

CUT-SETS SPACE SAVING

Graph	Cut-set matrix dimension	Column vectors	Cut-node matrix dimension	Column vectors	% Space saving
1	28 x 10	1 x 114	28 x 7	1 x 112	52.5
4	22 x 16	1 x 164	22 x 7	1 x 89	50.0
5	63 x 21	1 x 735	63 x 6	1 x 255	41.8
6	127 x 28	1 x 1874	127 x 7	1 x 575	44.9

With this simple modification the algorithm is limited practically only by the computer time available, while without this change the procedure is limited by the core size of the computer.

4.5.3 Tests

The algorithm for the minimal cut-set generation of non-separable graphs described in section (4.5.1) has been programmed for the CDC 6400 computer using Fortran IV.

Table IV shows the cut-node generation time for several randomly generated reliability graphs described in Appendix 1. The graph configuration index presented for each graph is defined by the quotient between number of nodes and number of elements. The times presented do not include the time required to calculate all node reliability bounds each time a new cut is generated. The time required by the program cannot be measured entirely in terms of the number of nodes but a rough indicator is the graph configuration index. Graphs with many edges in series (high index) will be solved more quickly than those with more grid-like configuration (low index).

Both sparse and non-sparse approaches were implemented and a comparison of the computation times is presented in table V. The

TABLE IV

COMPUTATION TIME FOR CUT-NODE GENERATION
USING SPACE SAVING MODIFICATION

Graph	Nodes	Elements	Configuration index	Number of cut-node	Total time (secs.)	Average cut-node time (msec.)
1	8	10	0.80	84	0.395	4.70
2	20	33	0.61	1514	1.65	1.09
3	15	25	0.60	875	1.12	1.28
4	8	16	0.50	61	0.409	6.70
5	7	21	0.33	192	0.672	3.50
6	8	28	0.29	448	1.78	3.97
7	9	36	0.25	1024	7.27	7.10
8	28	47	0.60	4514	4.88	1.08
9	14	21	0.67	856	1.06	1.24

TABLE V

COMPARISON OF COMPUTATION TABLE

Graph	Sparse Average cut-node time (msec)	Non-sparse Average cut-node time (msec)
1	4.70	5.4
2	1.09	5.2
3	1.28	4.7
4	6.70	7.8
5	3.50	6.5
6	3.97	8.8
7	7.10	20.5

TABLE VI

COMPUTATION TIME FOR DIFFERENT ELEMENT

REDUNDANCY IN FIGURE 3

Number of minimal cut-node sets = 61

Case	Elements	Configuration index	Average cut-node time (msec.)
1	16	0.50	7.8
2	26	0.31	8.7
3	36	0.22	9.4
4	46	0.17	9.9

space saving modification increases the efficiency of the algorithm reducing both the memory and computation-time requirements of the computer program implementation of the algorithm.

Computation time is almost independent of the redundancy between each pair of connected nodes in the graph, it increases the space requirement of the program but the number of cut-node sets is the same. This is an advantage of the cut-set approach compared with methods based on success paths which need to reduce parallel edges to an equivalent edge before defining the success paths in order to decrease running time. Table VI shows the cut-node generation time for the graph of Fig. 3 for different redundancy of the element. Case 1 corresponds to Fig. 3 and cases 2, 3 and 4 represent an increase of 1, 2 or 3 elements between each pair of connected nodes.

The input for the algorithm is the graph represented by its edge listing which identify each edge as a pair of unsigned nodes. Figures 4 to 6 list the minimal cut-sets and the cut-node incidence of Fig. 2 as they are output by the computer program and are examples of the node numbering independence of the algorithm.

4.6 Separable graphs

Deo[42] has proved that the node connectivity of any graph G can never exceed the edge connectivity of G which at the same time cannot exceed the degree of the node with the smallest degree in G. Then any graph with one or more pendant nodes (degree one) is always a separable graph but the converse is not also true, i.e., not all separable graphs have pendant nodes. This is illustrated by Figs. 7 and 8 taken from references [42] and [44] respectively. Node 6 in both figures is an articulation point but only Fig. 8 has a pendant node identified as node 7.

Edge listing

E	SB	E
1	1	3
2	3	4
3	4	7
4	7	8
5	1	2
6	2	5
7	5	6
8	6	8
9	3	5
10	5	7

Cut	Edges in the cut	Cut-node incidence
1	1,5	2,3,4,5,6,7,8
2	5,6	2
3	1,6	3,4,5,6,7,8
4	1,2,9	3
5	2,5,9	2,4,5,6,7,8
6	2,6,9	4,5,6,7,8
7	2,3	4
8	1,3,9	4,5
9	3,5,9	2,5,6,7,8
10	3,6,9	5,6,7,8
11	6,7,9,10	5
12	2,7,10	4,6,7,8
13	3,7,10	6,7,8
14	5,7,9,10	2,5
15	1,7,9,10	3,4,6,7,8
16	7,8	6
17	6,8,9,10	5,6
18	2,8,10	4,7,8
19	3,8,10	7,8
20	5,8,9,10	2,5,6
21	1,8,9,10	3,4,7,8
22	3,4,10	7
23	4,7	6,8
24	4,8	8
25	2,4,10	4,7
26	1,4,9,10	3,4,7
27	4,5,9,10	2,5,6,8
28	4,6,9,10	5,6,8

Figure 4. Minimal cut-sets and cut-node incidence for Fig. 2 First numbering of nodes

Edge listing

E	SB	E
1	1	2
2	2	3
3	3	4
4	4	5
5	1	6
6	6	7
7	7	8
8	8	5
9	2	7
10	7	4

Cut	Edges in the cut	Cut-node incidence
1	1,5	2,3,4,5,6,7,8
2	1,2,9	2
3	2,5,9	3,4,5,6,7,8
4	2,3	3
5	1,3,9	2,3
6	3,5,9	4,5,6,7,8
7	3,4,10	4
8	2,4,10	4,5
9	1,4,9,10	2,3,4
10	4,5,9,10	5,6,7,8
11	4,8	5
12	3,8,10	5,6
13	2,8,10	3,4,5
14	1,8,9,10	2,3,4,5
15	5,8,9,10	6,7,8
16	5,6	6
17	1,6	2,3,4,5,7,8
18	2,6,9	3,4,5,7,8
19	3,6,9	4,5,7,8
20	4,6,9,10	5,7,8
21	6,8,9,10	7,8
22	6,7,9,10	7
23	4,7	5,8
24	7,8	8
25	2,7,10	3,4,5,8
26	3,7,10	4,5,8
27	5,7,9,10	6,7
28	1,7,9,10	2,3,4,5,8

Figure 5. Minimal cut-sets and cut-node incidence for Fig. 2 Second numbering of nodes

Edge listing

E	SB	E
1	1	7
2	7	8
3	8	6
4	6	4
5	1	2
6	2	5
7	5	3
8	3	4
9	7	5
10	5	6

Cut	Edges in the cut	Cut-node incidence
1	1,5	2,3,4,5,6,7,8
2	5,6	2
3	1,6	3,4,5,6,7,8
4	6,7,9,10	5
5	5,7,9,10	2,5
6	1,7,9,10	3,4,6,7,8
7	7,8	3
8	6,8,9,10	3,5
9	5,8,9,10	2,3,5
10	1,8,9,10	4,6,7,8
11	4,8	4
12	4,7	3,4
13	4,6,9,10	3,4,5
14	4,5,9,10	2,3,4,5
15	1,4,9,10	6,7,8
16	3,4,10	6
17	3,6,9	3,4,5,6
18	3,5,9	2,3,4,5,6
19	1,3,9	7,8
20	3,8,10	4,6
21	3,7,10	3,4,6
22	1,2,9	7
23	2,7,10	3,4,6,8
24	2,8,10	4,6,8
25	2,4,10	6,8
26	2,3	8
27	2,5,9	2,3,4,5,6,8
28	2,6,9	3,4,5,6,8

Figure 6. Minimal cut-sets and cut-node incidence for Fig. 2 Third numbering of nodes

If the algorithm of section (4.5) is applied to the graphs of Figs. 7 and 8 the proper cut-set of node 6 will be defined by the set of edges (1,6) (2,6) (3,6) (6,8) (6,4) (6,5) (6,7) and (4,6) (8,6) (6,7) respectively, where each edge has been represented by its terminal nodes. However this cut-set is not a minimal cut-set between nodes 1 and 6 because the subsets (1,6) (2,6) (3,6) and (4,6) (8,6) break all paths between these pairs of nodes respectively. Then the algorithm of section 4.5 cannot be applied directly for the case of separable graphs.

A slight modification to the algorithm of section 4.5 based on the concept of directed edges allow to present in this section an algorithmic procedure for the generation of the minimal cut-sets between the reference node and all other nodes of separable graphs. It can be applied in the reliability analysis of transmission and distribution systems of electric power and is the most efficient cut-set enumeration method known to the author for such application.

4.6.1 Definitions and basic considerations

An undirected edge is a two-way branch represented by an unordered pair of nodes (n_i, n_j) and represented by a line segment between n_i and n_j. An undirected edge is said to be incident on nodes n_i and n_j.

A directed edge is a one-way branch represented by an ordered pair of nodes (n_i, n_j) and represented by a line segment between n_i and n_j with an arrow directed from n_i to n_j. This orientation indicates that a signal passing through the link (n_i, n_j) must be in at node n_i and out at node n_j. A directed edge is said to be incident out of the initial node n_i and incident into the terminal node n_j.

To generate the minimal cut-sets between node 1 and all other nodes of systems with configurations similar to the ones shown in Figs. 7 and 8,

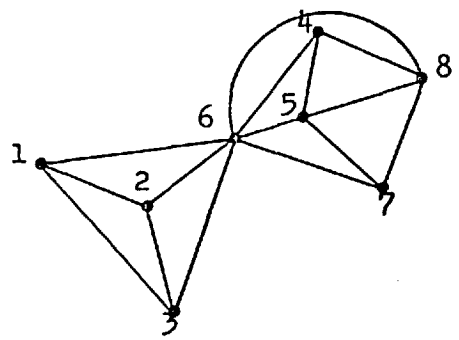


Figure 7. Separable graph without pendant nodes [42]

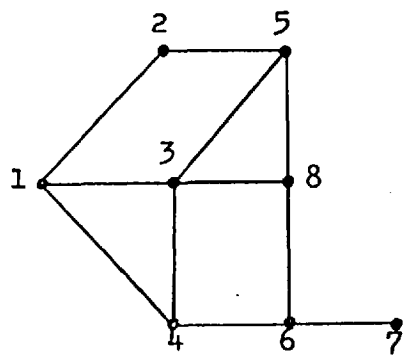


Figure 8. Separable graph with one pendant node [44]

both types of edges are used to represent the components. From Fig. 8 it can be seen that every node adjacent to a pendant node is an articulation point. However the algorithm proposed in this section is not designed to test the separability of a graph therefore all the articulation points are assumed to be known. The algorithm has been designed considering that all subsystems separated from the reference node by an articulation point n_k have tree configuration and can be represented by subgraphs with directed edges whose initial nodes are the nearest to n_k. This limitation was imposed because separable graphs with two or more nonseparable subgraphs of mesh structure are represented by undirected edges and can be efficiently analysed by successive applications of the algorithm of section (4.5) for each nonseparable subgraph. In such an application it should be noted that the set of minimal cuts between the reference node and any articulation point are also cut-sets for all nodes of the nonseparable subgraph connected to that articulation point. When enumerating the set of minimal cuts of the subgraph the articulation point represents the reference node and all edges incident out of that node are the inputs of the subsystem.

4.6.2 Description of the algorithm

The 5-step algorithm for nonseparable graphs is modified to solve the minimal cut-set enumeration of separable graphs by considering the orientation of the edges in each subgraph defined by the articulation points.

With this modification the proper cut-set generated when scanning each node is formed with the undirected edges incident on the node and the directed edges incident into that node. For example the proper cut-set of node 6 in Figs. 7 and 8 will be generated as the set of edges

(1,6) (2,6) (3,6) and (4,6) (8,6) respectively. In both figures 7 and 8 the set of edges (6,8) (6,4) (6,5) (6.7) and (6,7) respectively will be oriented incident out of node 6. All other cut-sets are generated in a similar way of section (4.5) by the ring sums of the proper cut-set of the node being scanned and all cut-sets from previous nodes which have at least one edge in common with this proper cut-set.

When scanning an articulation point the cut-node vector of the proper cut-set will indicate the incidence of such cut also in all nodes dependent of the articulation point. A node is defined to be dependent of an articulation point if all paths from the reference node to that node pass through the articulation point. It should be noted that all subgraphs dependent of an articulation point n_k have been assumed to be separable networks with tree configuration. A nonpendant node of this tree will be called an internal tree node. For all other cut-sets generated when scanning an articulation point the cut-node vector is generated as in the separable case but taking into account that a cut-set of the articulation point is also a cut-set for each node of the tree connected to that point. Similarly a cut-set which does not affect the articulation point it neither will affect any tree node connected to that point. As mentioned before the procedure assumes as known all the articulation points in the graph and also the internal tree nodes connected to a particular articulation point. The total number of nodes of the graph is designed by NN.

The algorithmic procedure can be described by 5 steps :

Step 1 : Generate the basic minimal cut-set MC_{1m} with the set of edges incident to the reference node. Construct a cut-node vector CNV_{1j}, j = 2,3,...,NN with all components equal to 1 and mark reference node as scanned FN(1) = 1. Set a node counter

NK equal to 1 and a cut-flag CF(1) equal to zero. Calculate lower and upper reliability bounds for each node of the network.

Step 2 : Check the node-counter. If it is less than NN choose the unscanned node with the lowest number and go to step 3. If it is equal to NN the algorithm terminates and all minimal cut-node sets have been generated and bounds for each node reliability are available.

Step 3 : Let the node chosen to be denoted as node i ($i>1$). Find the k elements incident on (undirected edges) and/or incident into (directed edges) node i. If node i is a tree-node (internal or pendant) go to step 4. If not, identify from the minimal cut-sets generated which ones contain one or more of the k elements and give to each one an index $CI_i(p)$ equal to the number of k elements in the pth cut-set. If at least one of the cut-sets already generated has a non-zero index, go to step 4. If not, advance to next unscanned node in increasing order from node i and repeat step 3.

Step 4 : Generate the proper minimal cut-set of node i with the k elements found in step 3. Compare this cut with all cuts already generated, if the node is not a tree node, and discard the other cut if it is identical to the proper cut-set. The discarded cut will not be used any more in the process. Construct a cut-node vector as follows: If node i is an articulation point or a tree-node the cut-node vector is equal to node i and other nodes dependent of node i. For all other types of nodes the cut-node vector is just equal to node i. Mark the cut as a proper cut-set setting a cut-flag equal to 1. If node i is a

tree-node or a pendant node, mark node i as scanned, increment node-counter and go to step 2. If not, go to step 5.

Step 5 : Make the ring sum between the proper cut-set of step 4 and each one of the cut-sets with a non-zero index identified in step 3. The new set of edges is a minimal cut-set if and only if it is not contained in any of the complete set of cuts previously generated. This procedure is applied for all cuts with non-zero index, beginning with the cuts of highest indices and continuing with cuts of decreasing index. Construct for each minimal cut-set the cut-node vector equal to the one of the cut-set combined with the proper cut-set except for the terms of node i and any node dependent of node i which must be equal to 0 or 1 if the flag of the cut combined with the proper cut-set is 0 or 1, respectively. Set the new cut-flag equal to the flag of the cut combined with the proper cut-set, mark node i as scanned, increment node-counter and go to step 2.

To illustrate the procedure the block reliability diagram shown in Fig. 9 is used and all cut-sets between node 1 and all other nodes of the diagram are determined. As defined in this section nodes 4 and 7 are articulation points, nodes 5 and 3 are tree-nodes and nodes 6, 8 and 10 are pendant nodes. It can be seen that each tree-node is also an articulation point. Table VII presents the results as they are produced when applying the algorithm step-by-step to Fig. 9. A detailed discussion and proof of the algorithm steps is included in section A2.3 of Appendix 2.

TABLE VII

CUT-SET GENERATION FOR FIGURE 9

Node	Cut	Elements in the cut	Cut-node incidence	Cut flag	Cut index				Cut generator
					2	3	4	7	
1	1	1,2	2,3,4,5,6,7,8,9,10	0	1	1			-
2	2	1,3,4	2	1			1	1	-
	3	2,3,4	3,4,5,6,7,8,9,10	0		1	1	1	1
3	4	2,5	3	1				1	-
	5	1,5	2,4,5,6,7,8,9,10	0				1	1
	6	3,4,5	4,5,6,7,8,9,10	0			1	2	3
4	7	3,6	4	1				1	-
	8	1,4,6	2,4,5,6,10	1				2	2
	9	2,4,6	3,7,8,9	0				2	3
	10	4,5,6	7,8,9	0				3	6
5	11	9	5,6	1					-
6	12	8	6	1					-
7	13	4,5,6	7	1	discard cut 10				
	14	3,6			discarded by cut 7				
	15	1,5			discarded by cut 5				
	16	2,5			discarded by cut 4				
	17	1,3,5,6			disjoint union of cuts 5 and 7				
	18	2,3,5,6			disjoint union of cuts 4 and 7				
	19	2,4,6			discarded by cut 9				
	20	1,4,6			discarded by cut 8				
	21	3,4,5			discarded by cut 6				
8	22	11	8	1					-
9	23	7	8,9	1					-
10	24	10	10	1					-

4.6.3 Tests

The computer program implementation of the algorithm uses the same storage scheme of section (4.5.2) for the information concerning the minimal cut-set and cut-node incidence matrices. The algorithm has been programmed for the CDC 6400 computer using Fortran IV.

Table VIII shows the cut-node generation time for three graphs described in Appendix 1. Graph 3 corresponds to the 1973 West-Venezuelan 115 kV electric power system. The time presented do not include the time required for calculation of reliability bounds for each node and are comparable to those obtained for nonseparable graphs. Graphs with tree configuration have high index configuration as shown in table VIII.

The inputs of the algorithm are the edge listing and the articulation points and pendant nodes of the graph. Each undirected edge is represented as a pair of unsigned nodes and each directed edge by a pair of signed nodes. The initial vertex of a directed edge is expressed by a negative number and the terminal node as a positive one. Each articulation point is defined with all the nodes of the tree connected to that point.

4.7 Acyclic directed graphs

The nonseparable graphs considered in this chapter so far have been undirected graphs. In this section we consider again the case of nonseparable graphs but now all edges are one-way branches. The concept of directed edges of section (4.6.1) is used to represent the edges of a directed graph. Then a directed graph G consists of a set of elements called nodes and a set of ordered pairs of nodes called directed edges. A directed graph that has no directed circuits is called acyclic. All directed graphs considered in this section are assumed to be acyclic

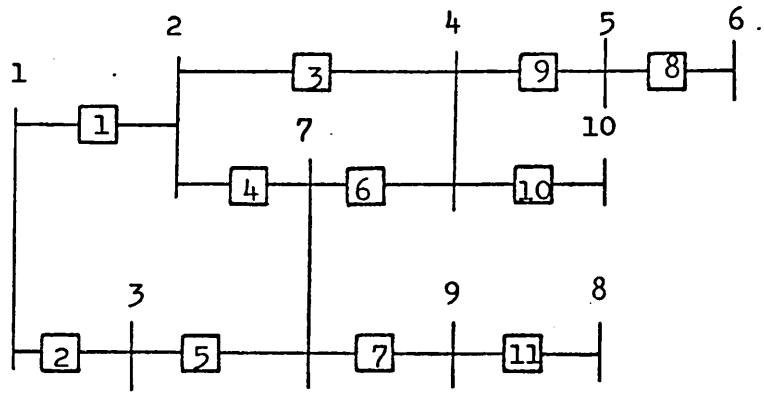


Figure 9. Reliability block diagram No. 3

TABLE VIII

COMPUTATION TIME FOR CUT-NODE GENERATION IN SEPARABLE GRAPHS

Graph	Nodes	Elements	Configuration index	Number of cut-node	Total time (sec.)	Average cut-node time (msec.)
10	10	12	0.83	57	0.557	9.77
11	11	14	0.79	106	0.650	6.13
12	15	18	0.83	449	1.660	3.70

and will be called simply directed graphs or digraphs for short.

Many systems can be modelled by directed graphs. In particular digital systems with one-way components are represented by directed graphs. In digital protection systems, analogue data (current and voltages) are sampled from the transducers in the substation through sample-hold circuits. These values are converted to digital form by A/D converters before they are fed into the processor responsible for the tripping criteria. Directed graphs can be employed for the representation of such systems when analysing the terminal reliability between the input and output of the system. Acyclic digraphs have been also extensively used for the representation of the activity networks[45] defined in the Critical Path Method (CPM) or in the Program Evaluation and Review Technique (PERT).

This section presents an algorithm for the minimal cut-set enumeration between the source node (s) and the sink node (t) of acyclic digraphs.

4.7.1 Basic definitions

A directed path from node s to t is an alternating sequence of nodes and edges beginning with s and ending with t such that all nodes and edges are distinct. Each edge in the sequence is oriented from the node preceding it to the node following it. We refer to such path as s-t directed path or an s-t path for brevity. Node s have only edges incident out of it and node t have only edges incident into it. Then any s-t directed path will always include one edge incident out of s and one incident into t.

A set of edges in a directed graph G is an s-t directed minimal cut-set if its removal from G breaks all directed paths from s to t, and no proper subset holds the same property. Since all graphs in this section

are directed graphs we shall refer to directed paths and directed minimal cut-sets simply as paths and cut-sets. As mentioned before a digraph G is connected if its corresponding undirected graph is connected. The same concept applies for any directed subgraph of G.

4.7.2 Description of the algorithm[43]

The 4-step algorithm presented in this section is concerned with the enumeration of the minimal cut-sets between the origin node s (or reference node) and the terminal node t (output of the system) of acyclic digraphs.

The algorithm is a node-oriented procedure which, starting from the origin, continues the search with the nearest nodes to the origin and ends when (NN - 1) nodes have been scanned.

Each time the algorithm scans a node the set of edges incident into and out of that node are identified. Since the origin has only edges out of it, the first cut-set produced by the process is that set of edges and the algorithm jumps to one of the nearest unscanned nodes. Each time a new node w is scanned, an index is created for each one of the cut-sets already generated. This index is equal to the number of edges incident into w that are contained in each cut-set. Each cut-set with an index not equal to zero will form a possible cut-set replacing the edges leading into w that are contained in the cut by the edges going out of w. A comparison is made with the previous cut-sets generated when scanning w to see if the new cut is minimal or not. This input-output substitution and the comparisons between cut-sets is used iteratively by the algorithm to enumerate all s-t cut-sets.

The algorithmic procedure can be described in 4 steps :

Step 1 : Generate the basic minimal cut-set MC_{1m} with the set of edges incident out of the reference node. Mark reference node as scanned FN(1) = 1 and set a node-counter NK equal to 1. Calculate lower and upper bounds on system reliability. Go to step 2.

Step 2 : Check the node-counter. If it is less than NN - 1 (NN number of nodes) choose the unscanned node with the lowest number and go to step 3. If it is equal to NN - 1 the algorithm terminates and all minimal cut-sets have been generated and bounds on system reliability have been calculated.

Step 3 : Let the node chosen to be denoted as node i ($i > 1$). Find the [k] and [p] elements which are incident into or going out of node i respectively. Identify from the minimal cut-sets generated which contain one or more of the k elements and give to each an index $CI_{i(j)}$ equal to the number of k elements in the jth cut. If at least one of the cuts already generated has index equal to k go to step 4. If not, advance to next unscanned node in increasing order from node i and repeat step 3.

Step 4 : A minimal cut with index x, x = 1,2...,k generates a new cut-set substituting its x element for the p elements of node i, where x are the k elements of node i contained in that cut. The new cut is a minimal cut if and only if it is not contained in anyone of the cuts already generated for the node i. This procedure is applied for all the cuts

with index, beginning with the cuts of highest indices and continuing with cuts of decreasing index. For each new minimal cut-set, calculate new lower and upper bounds for the system reliability. Mark node i as scanned, increment nodes-counter and go to step 2.

To illustrate the procedure we will use the reliability graph of Fig. 3 used by Nelson et al[30] for illustration of their uni-directional algorithm.

Step 1 : MC_{1m} = 1,2,3 (first cut-set)

$FN(1)$ = 1

NK = 1

Step 2 : NK = 1 (< 7)

Choose node 2, go to step 3

Step 3 : i = 2

[k] = 1,2

[p] = 4,5

$CI_2(1)$ = 2

Step 4 : MC_{2m} = 3,4,5 (second cut-set)

$FN(2)$ = 1

NK = 2

Step 2 : NK = 2 (< 7)

Choose node 3, go to step 3

Step 3 : $i = 3$

$[k] = 4,5,6$

$[p] = 7,8,10,14$

$CI_3(1) = 0$

$CI_3(2) = 2$

Discard node 3 and choose node 4

Step 3 : $i = 4$

$[k] = 7,8$

$[p] = 9$

$CI_4(1) = CI_4(2) = 0$

Discard node 4 and choose node 5

Step 4 : $i = 5$

$[k] = 3$

$[p] = 6$

$CI_5(1) = CI_5(2) = 1$

Step 4 : $MC_{3m} = 1,2,6$ (third cut-set)

$MC_{4m} = 4,5,6$ (fourth cut-set)

$FN(5) = 1$

$NK = 3$

Step 2 : $NK = 3 \quad (< 7)$

Choose node 3, go to step 3

Step 3 : $i = 3$

$[k] = 4,5,6$

$[p] = 7,8,10,14$

$CI_5(1) = 0$

$CI_5(2) = 2$

$CI_5(3) = 1$

$CI_5(4) = 3$

Step 4 : $MC_{5m} = 7,8,10,14$ (fifth cut-set)

$MC_{6m} = 3,7,8,10,14$ non minimal

$MC_{7m} = 1,2,7,8,10,14$ non minimal

$FN(3) = 1$

$NK = 4$

Step 2 : $NK = 4 \quad (< 7)$

Choose node 4, go to step 3

Step 3 : $i = 4$

$[k] = 7,8$

$[p] = 9$

$CI_4(1) = CI_4(2) = CI_4(3) = CI_4(4) = 0$

$CI_4(5) = 2$

Step 4 : $MC_{6m} = 9,10,14$ (sixth cut-set)

$FN(4) = 1$

$NK = 5$

Step 2 : NK = 5 (< 7)

Choose node 6, go to step 3

Step 3 : i = 6

[k] = 9,10

[p] = 11,12,13

$CI_6(1) = CI_6(2) = CI_6(3) = CI_6(4) = CI_6(5) = 0$

$CI_6(6) = 2$

Step 4 : MC_{7m} = 11,12,13,14 (seventh cut-set)

FN(6) = 1

NK = 6

Step 2 : NK = 6 (< 7)

Choose node 7, go to step 3

Step 3 : i = 7

[k] = 14

[p] = 15,16

$CI_7(1) = CI_7(2) = CI_7(3) = CI_7(4) = 0$

$CI_7(5) = CI_7(6) = CI_7(7) = 1$

Step 4 : MC_{8m} = 7,8,10,15,16 (eighth cut-set)

MC_{9m} = 9,10,15,16 (ninth cut-set)

MC_{10m} = 11,12,13,15,16 (tenth cut-set)

FN(7) = 1

NK = 7

Step 2 = NK = 7 stop.

A detailed discussion and proof of the algorithm is included in section A2.4 of Appendix 2.

4.7.3 Tests

The minimal cut-set generation algorithm for acyclic directed graphs have been programmed for the CDC 6400 computer in Fortran IV. The space saving modification presented in section (4.5.2) is not incorporated in this algorithm in order that a more realistic comparison with the approach presented by Nelson et al[30],[31] can be made.

Figure 10 shows the listing of the minimal cut-sets and system reliability bounds for the system in Fig. 3 as they are output by the computer program. Figure 3 corresponds to the system example used by Nelson et al[30]. The algorithm presented here listed the 10 minimal cut-sets between IN and OUT terminals and calculated the reliability bounds using the inclusion-exclusion method (see chapter 3, section 3.9) with all cut-sets in 1.45 seconds. The program uses only 0.455 seconds to list the 10 cut-sets without calculating reliability bounds. A probability of success P is given to each element of Fig. 3 as shown in Fig. 10.

Table IX presents the reliability bounds and computing times using the program reported by Batts[31] for 3 different cases. When all cut-sets are generated and used to calculate the bounds the computing time is 11.22 seconds. Even in the case of generating only the cut-sets up to third order the computing time greatly exceeds the computing time of the present algorithm in generating all the cut-sets.

INPUT DATA

E	SB	EB	P	Q
1	1	2	0.80	0.20
2	1	2	0.80	0.20
3	1	3	0.90	0.10
4	2	4	0.85	0.15
5	2	4	0.75	0.25
6	3	4	0.87	0.13
7	4	5	0.82	0.18
8	4	5	0.82	0.18
9	5	6	0.89	0.11
10	4	6	0.88	0.12
11	6	8	0.85	0.15
12	6	8	0.85	0.15
13	6	8	0.85	0.15
14	4	7	0.75	0.25
15	7	8	0.70	0.30
16	7	8	0.70	0.30

MINIMAL CUT MATRIX

CUT	ELEMENTS
1	1110000000000000
2	0011100000000000
3	1100010000000000
4	0001110000000000
5	0000001101000100
6	0000000011000100
7	0000000000111100
8	0000001101000011
9	0000000011000011
10	0000000000111011

SYSTEM RELIABILITY BOUNDS

CUT	LOWER	UPPER
1	.99600	.99600
2	.99240	.99240
3	.98772	.98772
4	.98351	.98351
5	.98255	.98255
6	.97941	.97941
7	.97860	.97860
8	.97834	.97834
9	.97749	.97749
10	.97727	.97727

Figure 10. Minimal cut-sets and system reliability bounds (6th order approx.)for system in Fig. 3

TABLE IX

COMPUTATION TIME USING NELSON ET AL APPROACH[30],[31]

Cut-sets generated	System reliability	Computation time (secs.)
All	0.97726	11.22
Up to 4th order	0.97726	9.40
Up to 3rd order	0.98006	2.48

The previous results show that the present algorithm is simpler than the Nelson et al[30] approach which needs to find all the 55 success paths between nodes 1 and 8 of Fig. 3 before starting to generate the minimal cut-sets. Afterwards they need to perform all k logic sums between the 16 column vectors of that path matrix to generate a k-order cut. Besides, when they generate a cut it needs to be compared with all cuts of lower order to establish if it is minimal or not.

Table X shows the computation times for the minimal cut-set enumeration of the same network of Fig. 3 but varying the redundancy of elements between each pair of connected nodes. Case 1 corresponds to Fig. 3 and cases 2, 3 and 4 represent an increase of 1, 2 or 3 elements between each pair of connected nodes. This is another advantage of the new algorithm compared with Nelson et al[30] because greater redundancy increases the number of success paths. Times in table X do not include the time to calculate the system reliability bounds.

TABLE X

COMPUTATION TIMES FOR DIFFERENT ELEMENT REDUNDANCIES IN FIG. 3 USING THE ALGORITHM OF SECTION 4.7

Case	Elements	Configuration index	Computing time (sec.)
1	16	0.50	0.455
2	26	0.31	0.466
3	36	0.22	0.539
4	46	0.17	0.595

4.8 Node failure problem

In many practical applications, the nodes of a graph may be also unreliable elements. For example, in an electric power system, the nodes represent busbars and the branches represent transmission lines. In a power system the failure of a branch (transmission line) occurs when the branch suddenly becomes incapable of carrying power and the failure of a node (busbar) occurs when electric power cannot flow through that electrical junction.

In general a node is outaged if the node itself has failed e.g., a busbar fault, or if no path exists from the node to any of the inputs of the system through branches and nodes in the up state. Components (branches or nodes) which have failed are said to be down. After a component has been repaired and is operating normally, it is said to be up. Up and down are referred to as the states of the components. The distinction between a node failure, which would usually represent a busbar fault, and a node outage, which is caused by the failure of one or

more branch(es) and/or node(s), should be noticed. Whether or not a node is outaged is, in general, a function of the states of all branches and nodes in the system.

A sample network of an electric power system is shown in Fig. 11. In this system there are three sources, one at each of busbars 1, 7 and 8. Figure 12 presents the reliability block diagram of the same system where the inputs have been combined into a single node called the reference node. The reference can never fail and is not included when considering node failures.

The cut-set approach by Billinton and Grover[79] assumes that generation busbars are 100% reliable. The influence of busbar faults on node outage rates is also neglected in [79] by considering them as 100% reliable when they are in the paths for other busbars. The equations provided in [79] can be used to determine the outage rates and durations associated with first, second and third order cut-set but they are considered only for single, double or triple line outages. It was considered that cut-sets higher than third order contribute very little to the load point reliability indices and were not included in the reliability evaluation.

Considering the third-order cut-set as the highest contingency to be evaluated, there are several combinations which can produce a nodal outage :

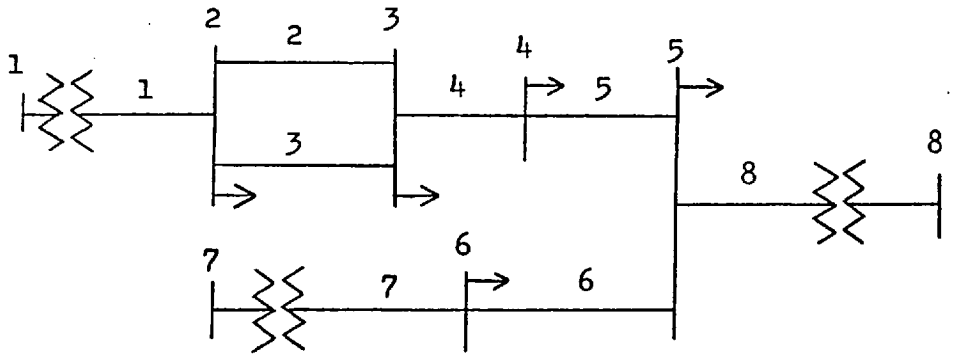


Figure 11. Distribution system example [79]

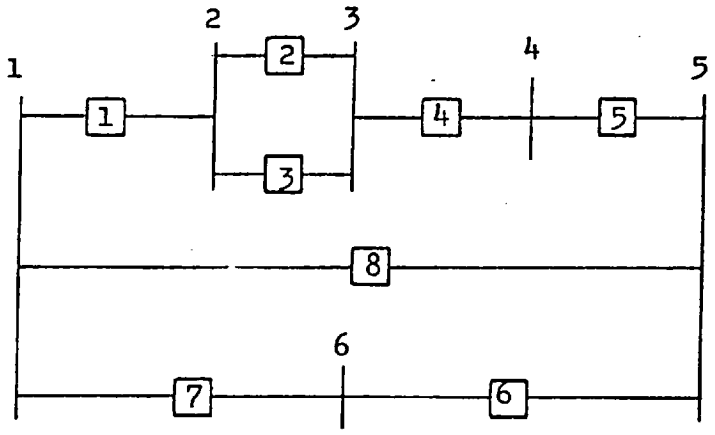


Figure 12. Reliability block diagram of Fig. 11

	Event	Cut-set order
1	One line outage	First
2	One node outage	First
3	Two line outages	Second
4	Node and line outages	Second
5	Two node outages	Second
6	Three line outages	Third
7	Node and two line outages	Third
8	Two nodes and one line outage	Third
9	Three node outages	Third

In order to evaluate the influence of busbar faults on the reliability indices of each node, an evaluation was done of the values of the previous nine events using the equations of [79] and considering that :

(i) all the lines have the same failure rates (λ_{line}) and outage duration

(ii) all the nodes have the same failure rates (λ_{node}) and outage duration with values between 1% to 100% of the line outage values.

(iii) line parameters are as follows :

normal weather	- permanent outage rate:	0.6	f/y
	- temporary outage rate:	1.0	f/y
adverse weather	- permanent outage rate:	30.0	f/y
	- temporary outage rate:	10.0	f/y
outage duration:		3.0	h
maintenance	- outage rate:	2.0	m/y
	- duration:	6.0	h

The minimal cut-sets used for such evaluation is

Event	Lines	Nodes
1	1 0 0	0 0 0
2	0 0 0	1 0 0
3	1 1 0	0 0 0
4	1 0 0	1 0 0
5	0 0 0	1 1 0
6	1 1 1	0 0 0
7	1 1 0	1 0 0
8	1 0 0	1 1 0
9	0 0 0	1 1 1

Table XI presents the permanent outage rates for these 9 events. It is clear from this table that a first order cut set produced by a node failure has a greater contribution to the node reliability index than a second order cut-set due to a double line outage even at the minimum node failure rate considered, i.e., 0.06 f/y. A second order cut-set produced by simultaneous failures of one line and one node also makes a greater contribution than the third order cut-set of a triple line contingency which is also lower than a double node failure for a node failure rate equal to 15% of the line failure rate i.e., one failure every 10 years. The third order cut-set caused by a double line contingency and one node failure is also comparable to the cut-set of the same order constituted by a triple line contingency. The third order cut-sets produced by three node failures or two nodes and one line failures can be neglected due to the small values even for high node failure rate. Table XII presents the temporary outage rates for the same 9 events. Equivalent comparisons can be done in this case as for table XI.

Tables XI and XII show that if meaningful reliability indices are to be obtained in a cut-set approach to failure modes and effects analysis in electric power systems, then node failures must be considered in such analyses.

TABLE XI

PERMANENT OUTAGE RATES FOR FIRST, SECOND AND THIRD ORDER CUT-SETS[1]

$\frac{\lambda_{node}}{\lambda_{line}}$	FIRST-ORDER CUT		SECOND-ORDER CUT			THIRD-ORDER CUT			
	1 Line Outage	1 Node Outage	2 Lines Outage	1 Line and 1 Node Out.	2 nodes Outage	3 Lines Outage	2 Lines and 1 Node Out.	1 Line and 2 Node Out.	3 Nodes Outage
.01	819	8.2	2.7	.027	.0003	.038	.0004	.0000	.0000
.05	819	40.9	2.7	.136	.0068	.038	.0019	.0001	.0000
.10	819	81.9	2.7	.272	.0272	.038	.0038	.0004	.0000
.15	819	122.8	2.7	.408	.0613	.038	.0056	.0008	.0001
.20	819	163.8	2.7	.545	.1089	.038	.0075	.0015	.0003
.25	819	204.7	2.7	.681	.1702	.038	.0094	.0024	.0006
.30	819	245.7	2.7	.817	.2450	.038	.0113	.0034	.0010
.35	819	286.6	2.7	.953	.3335	.038	.0132	.0046	.0016
.40	819	327.5	2.7	1.09	.4356	.038	.0151	.0060	.0024
.45	819	368.5	2.7	1.23	.5514	.038	.0169	.0076	.0034
.50	819	409.4	2.7	1.36	.6807	.038	.0188	.0094	.0047
.55	819	450.4	2.7	1.50	.8236	.038	.0207	.0114	.0063
.60	819	491.3	2.7	1.63	.9802	.038	.0226	.0136	.0081
.75	819	532.3	2.7	1.77	1.15	.038	.0245	.0159	.0103
.80	819	655.1	2.7	2.18	1.74	.038	.0301	.0241	.0193
.85	819	696.0	2.7	2.31	1.97	.038	.0320	.0272	.0231
.90	819	737.0	2.7	2.45	2.21	.038	.0339	.0305	.0274
.95	819	777.9	2.7	2.59	2.46	.038	.0358	.0340	.0323
1.00	819	819	2.7	2.7	2.7	.038	.038	.038	.038

[1] All values are multiplied by 10^{-3}

TABLE XII

TEMPORARY OUTAGE RATES FOR FIRST, SECOND AND THIRD-ORDER CUT-SETS[1]

$\frac{\lambda_{node}}{\lambda_{line}}$	FIRST-ORDER CUT		SECOND-ORDER CUT			THIRD-ORDER CUT			
	1 Line Outage	1 Node Outage	2 Lines Outage	1 Line and 1 Node Out.	2 Nodes Outage	3 Lines Outage	2 Lines and 1 Node Out.	1 Line and 2 Node Out.	3 Nodes Outage
.01	1067	10.7	1.36	.0136	.0001	.0137	.0001	.0000	.0000
.05	1067	53.4	1.36	.0678	.0034	.0137	.0007	.0000	.0000
.10	1067	106.7	1.36	.1356	.0136	.0137	.0014	.0001	.0000
.15	1067	160.0	1.36	.2034	.0305	.0137	.0021	.0003	.0000
.20	1067	213.4	1.36	.2713	.0543	.0137	.0027	.0005	.0001
.25	1067	266.8	1.36	.3391	.0848	.0137	.0034	.0009	.0002
.30	1067	320.1	1.36	.4069	.1221	.0137	.0041	.0012	.0004
.35	1067	373.4	1.36	.4747	.1661	.0137	.0048	.0017	.0006
.40	1067	426.8	1.36	.5425	.2170	.0137	.0055	.0022	.0009
.45	1067	480.1	1.36	.6103	.2746	.0137	.0062	.0028	.0012
.50	1067	533.5	1.36	.6781	.3391	.0137	.0068	.0034	.0017
.55	1067	586.8	1.36	.7459	.4103	.0137	.0075	.0041	.0023
.60	1067	640.2	1.36	.8138	.4883	.0137	.0082	.0049	.0030
.65	1067	693.5	1.36	.8816	.5730	.0137	.0089	.0058	.0038
.70	1067	746.9	1.36	.9494	.6646	.0137	.0096	.0067	.0047
.75	1067	800.2	1.36	1.02	.7629	.0137	.0103	.0077	.0058
.80	1067	853.6	1.36	1.09	.8680	.0137	.0109	.0087	.0070
.85	1067	906.9	1.36	1.15	.9799	.0137	.0116	.0099	.0084
.90	1067	960.3	1.36	1.22	1.10	.0137	.0123	.0111	.0100
.95	1067	1013.6	1.36	1.29	1.22	.0137	.0130	.0123	.0117
1.00	1067	1067	1.36	1.36	1.36	.0137	.0137	.0137	.0137

[1] All values are multiplied by 10^{-3}

4.9 A branch-node cut-set algorithm[46]

In section (4.6) is presented an algorithm for separable graphs which was an extension of the algorithm for nonseparable graphs but introducing the concept of articulation points. The algorithm of section (4.6) can be used for both cases except that for the case of nonseparable graphs it is necessary to specify that the graph has no articulation point. The algorithm of section (4.6) is employed to develop a new procedure to list the feasible branch-node cut-sets as well as branch(es) and node (s) cut-sets.

4.9.1 Basic definitions and assumptions

Previous algorithms give the minimal cut-set and the cut-node incidence matrices between the reference node and all other nodes of the system considering only edge failures. This information is used to develop a procedure which lists the minimal cut-set for a system with unreliable nodes. The concepts introduced in previous sections are still valid but new ones will be introduced for the branch-node cut-set generation algorithm.

A busbar-flag vector is defined for each minimal cut-set determined using the algorithm of section (4.6). This vector contains the information about the terminal nodes of the edges in each cut-set. Each non-proper minimal cut-set defined by that algorithm has constructed for it a busbar-flag vector BF_j, $j = 2,..,NN$. Table XIII shows the minimal cut-set matrix, the cut-node incidence matrix and the busbar-flag vectors for the reliability block diagram shown in Fig. 1.

By definition a proper minimal cut-set is a set of edges incident on the same node that, when they fail, only the common node fails. The algorithm in section (4.6) identifies such cuts by a cut-flag equal to 1. Each proper cut-set listed when considering only edge failures will also

correspond to a cut-set including only the common node when nodes are assumed to be unreliable. This cut-set will be a first order cut-set representing a node failure. When considering separable graphs the failure of an articulation point will also disconnect all the tree nodes connected to that point.

TABLE XIII

MINIMAL CUT-SETS, CUT-NODE INCIDENCE AND BUSBAR-FLAG VECTORS OF FIG. 1

Minimal cut-sets	Cut-node incidence	Busbar-flag vectors
1,2	2,3,4	110
1,3,4	2	311
2,3,4	3,4	121
2,3,5	3	131
4,5	4	112
1,3,5	2,4	221

Considering the results of section (4.8) for the contribution of nodes, branches or branch-node failures on the temporary and permanent outage rates of each busbar in an electric power system, the algorithm for the minimal cut-set enumeration in such systems with unreliable nodes considers cut-sets up to the third-order and includes the following events:

First order cut-sets : one line failure
one node failure

Second order cut-sets : two line failures
one line - one node failures

Third order cut-sets : three line failures
two line - one node failures

From the possible 9 events mentioned in section (4.8) only those including single or multiple line contingency and/or single node failure will be considered. The others due to their small influence on the values of the reliability indices, will be excluded. The branch-node cut-set algorithm as presented in this section has been specially designed for the reliability evaluation of electric power systems.

4.9.2 Description of the algorithm

The information provided by the busbar-flag vector and the minimal cut-set and cut-node incidence matrices listed by the algorithm of section (4.6) allows us to extend such algorithm to include cut-sets produced by single node failure or branch-node failures.

Let us consider the results shown in table XIII to explain how this information is used to list node cut-sets and branch-node cut-sets. Once the algorithm of section (4.6) has been applied for the graph of Fig. 1 the minimal cut-set and cut-node incidence matrices listed in table XIII are available. Then the new algorithm uses each cut-set already listed to produce new combinations as follows: First the busbar-flag vector is constructed for the cut-set under scan. Then all combinations are made replacing element(s) by the equivalent node using the busbar-flag vector and the edges contained in the cut-set. For example the cut-set (1,2) in table XIII has a busbar-flag vector BF_{ij} = 1,1,0, j = 2,3,4 that indicates that the cut-set has one edge incident on node 2 (edge 1) and the other incident on node 3 (edge 2). The reference node (numbered 1) assumed to be 100% reliable is therefore not included in this approach. In the new cut-sets, nodes will be

numbered in increasing order starting from NE + 1 where NE is the number of edges in the graph. The number of components NC (edges and nodes) of the graph will be NC = NE + NN. With such conventions the cut-set (1,2) will generate two new cut-sets given by the sets (1,7) and (2,6). As indicated in table XIII cut-set (1,2) affects nodes 2,3 and 4. Then when an edge is replaced for its node to form a branch-node cut-set, the cut-node vector is also modified excluding the node that is now in the cut-set. This procedure gives the cut-node incidence vectors (2,4) and (3,4) for the branch-node cut-sets (1,7) and (2,6) respectively. The same process is repeated for each minimal cut-sets listed by the algorithm of section (4.6).

The algorithmic procedure can be described in 5 steps:

Step 1 : Generate the minimal cut-set and cut-node incidence matrices using the 5-setps algorithm of section (4,6). Let i be the variable identifying the number of a minimal cut-set, i = 1, NMC where NMC is the total number of minimal cut-sets already generated. Initialize a cut-counter.

Step 2 : Check the cut-counter. If it is less than or equal to NMC, take the next cut from the cut-set matrix of step 1, and go to step 3. If it is greater than NMC the algorithm terminates and all minimal node and branch-node cut-sets have been generated.

Step 3 : Check the cut-flag produced in step 1 for the ith cut-set. If it is 0 the cut-set is not proper, then go to step 4. If it is 1, the i-cut is proper and must generate only one cut-set corresponding to a single failure of the common node

of all the edges in the i-cut. Construct a cut-node incidence vector equal to the same vector of the ith cut-set. Increment the cut-counter and go to step 2.

Step 4 : Construct the busbar-flag vector for the ith cut. Generate all possible combinations of events having single node failure and/or branch(es)-node failures. Go to step 5.

Step 5 : Compare each possible new cut-set with all cut-sets already generated that have the same node failure. Only if the possible cut-set has not been already generated and is a minimal one, consider it is a branch-node cut-set. Construct its cut-node incidence vector equal to that of the ith cut-set except for the term of the node having failed and any other tree node fed from this node. Repeat step 5 for all the combinations generated in step 4. Increment cut-counter and go to step 2.

To illustrate the procedure, table XIV presents the results step-by-step when the node and branch-node cut-set of Fig. 1 are generated by the algorithm. The busbar-flag vector and steps 4 and 5 are made only for non-proper minimal cut-sets. If the same procedure is applied for the proper cut-sets of Fig. 1 the results obtained are presented in table XV. This table shows that each proper minimal cut-set only generates one minimal cut-set, different from those generated by non-proper minimal cut-sets, which corresponds to the failure of the common nodes of the edges in the cut-set. The incidence of the nodes and branch-node cut-sets on the nodes of the graph is contained in a cut-node incidence matrix

TABLE XIV

GENERATION OF MINIMAL CUT-SET INCLUDING
SINGLE NODE OUTAGE OF FIG. 1

Old cut		Busbar-flag			Possible new cut		New cut		Cut-node incidence	Observations
No.	Elements	2	3	4	Elements	Nodes	No.	Components		
1	1,2	1	1	0	2	2	7	2,6	3,4	
					1	3	8	1,7	2,4	
2	1,3,4				-	2	9	6	2	Proper cut
3	2,3,4	2	2	1	2	2				See cut 7
					4	3	10	4,7	4	
					2,3	4	11	2,3,8	3	
4	2,3,5				-	3	12	7	3	Proper cut
5	1,3,5	2	2	1	5	2	13	5,6	4	
					1	3				See cut 8
					1,3	4	14	1,3,8	2	
6	4,5				-	4	15	8	4	Proper cut

TABLE XV

GENERATION OF CUT-SET FROM THE PROPER CUT-SET OF FIG. 1

Old cut		Busbar-flag			Possible new cut		New cut		Cut-node incidence	Observations[1]
No.	Elements	2	3	4	Elements	Nodes	No.	Components		
2	1,3,4	3	1	1	-	2	16	6	2	See cut 9
					1,4	3	17	1,4,7	2,4	Non-minimum, see cut 8
					1,3	4	18	1,3,8	2	See cut 14
4	2,3,5	1	3	1	2,5	2	19	2,5,6	3,4	Non-minimum, see cut 7
					-	3	20	7	3	See cut 12
					2,3	4	21	2,3,8	3	See cut 11
6	4,5	1	1	2	5	2	22	5,6	4	See cut 13
					4	3	23	4,7	4	See cut 10
					-	4	24	8	4	See cut 15

[1] All cuts mentioned are the new cuts in table XIV

similar to that used for the algorithm of section (4.6). It can be seen in table XIV that different cut-sets can produce similar combinations. For example the first combination produced with the old cut No. 3 is similar to the one produced by the first combination of the old cut No. 1. Similarly a new combination can be minimal with respect to others previously generated. Therefore the previous combination must be discarded from the minimal cut-set list. Then each new combination produced by the algorithm must be compared with the previous ones to see that a cut-set will not be generated more than once or if it is minimal. However the number of comparisons is kept to a minimum because only cut-sets with the same node failure are used.

The approach presented for the node and branch-node cut-set enumeration is a combinational procedure that uses the results from the algorithm of section (4.6) to list all possible combinations of branches and nodes that can result in minimal cut-sets. It is an exhaustive procedure that generates all node and branch-node minimal cut-sets.

4.9.3 Tests

The computer program implementation of the algorithm in section (4.6) has been extended to include steps 2 to 5 of the new algorithm. The extension also uses the space-saving modification of section (4.5.2) for efficient storage of the minimal cut-set and cut-node incidence matrices.

Table XVI shows the computing times for two randomly generated graphs and for the West Venezuelan 115 kV system (graph 12). The same graphs were used when testing the algorithm of section (4.6) as shown in table VIII. Table XVI indicates that the inclusion of node and branch-node cut-sets does not increase the average cut-node generation time. Figures 13 to 15 list the minimal cut-set and cut-node incidence

INPUT DATA

E	SB	EB
1	1	2
2	1	3
3	2	4
4	2	7
5	3	7
6	7	4
7	-7	9
8	-7	9
9	-4	5
10	-4	10
11	-9	8
12	-5	6

MINIMAL CUT MATRIX

CUT	ELEMENTS		
1	1	2	
2	1	3	4
3	2	3	4
4	2	5	
5	1	5	
6	3	4	5
7	3	6	
8	1	4	6
9	2	4	6
10	9		
11	12		
12	4	5	6
13	11		
14	7	8	
15	10		
16	2	13	
17	1	14	
18	13		
19	3	4	14
20	2	4	15
21	14		
22	5	13	
23	1	18	
24	4	5	15
25	3	18	
26	15		
27	6	13	
28	1	4	15
29	4	6	14
30	2	18	
31	16		
32	17		
33	18		
34	19		
35	20		
36	21		

CUT-NODE INCIDENCE MATRIX

CUT	NODES								
1	2	3	4	5	6	7	8	9	10
2	2								
3	3	4	5	6	7	8	9	10	
4	3								
5	2	4	5	6	7	8	9	10	
6	4	5	6	7	8	9	10		
7	4	5	6	10					
8	2	4	5	6	10				
9	3	7	8	9					
10	5	6							
11	6								
12	7	8	9						
13	8								
14	9	8							
15	10								
16	3	4	5	6	7	8	9	10	
17	2	4	5	6	7	8	9	10	
18	2								
19	4	5	6	7	8	9	10		
20	3	7	8	9					
21	3								
22	4	5	6	7	8	9	10		
23	2	4	5	6	10				
24	7	8	9						
25	4	5	6	10					
26	4	5	6	10					
27	4	5	6	10					
28	2								
29	7	8	9						
30	3								
31	5	6							
32	6								
33	7	8	9						
34	8								
35	9	8							
36	10								

Figure 13. Output of the branch-node algorithm
First numbering of nodes of graph 10.

INPUT DATA

E	SB	EB
1	1	10
2	1	5
3	10	7
4	10	2
5	5	2
6	2	7
7	-2	8
8	-2	8
9	-7	9
10	-7	3
11	-8	6
12	-9	4

MINIMAL CUT MATRIX

CUT	ELEMENTS		
1	1	2	
2	10		
3	12		
4	2	5	
5	1	5	
6	4	5	6
7	2	4	6
8	1	4	6
9	11		
10	3	6	
11	3	4	5
12	2	3	4
13	7	8	
14	9		
15	1	3	4
16	1	16	
17	2	21	
18	14		
19	15		
20	16		
21	1	13	
22	5	21	
23	13		
24	2	13	
25	4	6	16
26	2	4	18
27	1	4	18
28	6	21	
29	17		
30	18		
31	3	13	
32	3	4	16
33	4	5	18
34	19		
35	20		
36	21		

CUT-NODE INCIDENCE MATRIX

CUT	NODES								
1	2	3	4	5	6	7	8	9	10
2	3								
3	4								
4	5								
5	2	3	4	6	7	8	9	10	
6	2	6	8						
7	5	2	6	8					
8	3	4	7	9	10				
9	6								
10	7	3	4	9					
11	2	6	8	7	3	4	9		
12	5	2	6	8	7	3	4	9	
13	8	6							
14	9	4							
15	10								
16	2	3	4	6	7	8	9	10	
17	2	3	4	5	6	7	8	9	
18	3								
19	4								
20	5								
21	3	4	7	9	10				
22	2	3	4	6	7	8	9		
23	2	6	8						
24	5								
25	2	6	8						
26	5	2	6	8					
27	10								
28	3	4	7	9					
29	6								
30	7	3	4	9					
31	7	3	4	9					
32	2	6	8	7	3	4	9		
33	2	6	8						
34	8	6							
35	9	4							
36	10								

Figure 14. Output of the branch-node algorithm
Second numbering of nodes of graph 10.

INPUT DATA

E	SB	EB
1	1	6
2	1	8
3	6	10
4	6	4
5	8	4
6	4	10
7	-4	3
8	-4	3
9	-10	2
10	-10	5
11	-3	9
12	-2	7

MINIMAL CUT MATRIX

CUT	ELEMENTS		
1	1	2	
2	9		
3	7	8	
4	10		
5	1	3	4
6	2	3	4
7	4	5	6
8	12		
9	2	5	
10	1	5	
11	3	4	5
12	2	4	6
13	11		
14	3	6	
15	1	4	6
16	2	17	
17	1	19	
18	13		
19	14		
20	16		
21	17		
22	3	4	19
23	2	4	21
24	15		
25	18		
26	19		
27	1	15	
28	5	17	
29	3	15	
30	4	5	21
31	2	15	
32	4	6	19
33	20		
34	21		
35	6	17	
36	1	4	21

CUT-NODE INCIDENCE MATRIX

CUT	NODES								
1	2	3	4	5	6	7	8	9	10
2	2	7							
3	3	9							
4	5								
5	6								
6	2	3	4	5	7	8	9	10	
7	4	3	9						
8	7								
9	8								
10	2	3	4	5	6	7	9	10	
11	2	3	4	5	7	9	10		
12	4	3	9	8					
13	9								
14	10	2	7	5					
15	6	10	2	7	5				
16	2	3	4	5	7	8	9	10	
17	2	3	4	5	6	7	9	10	
18	2	7							
19	3	9							
20	5								
21	6								
22	2	3	4	5	7	9	10		
23	3	4	8	9					
24	4	3	9						
25	7								
26	8								
27	2	5	6	7	10				
28	2	3	4	5	7	9	10		
29	2	5	7	10					
30	3	4	9						
31	8								
32	4	3	9						
33	9								
34	10	2	7	5					
35	10	2	7	5					
36	6								

Figure 15. Output of the branch-node algorithm
Third numbering of nodes of graph 10.

matrices of graph 10 as they are output by the computer program. These figures show the node-numbering independence of the algorithm.

TABLE XVI

COMPUTATION TIMES FOR THE BRANCH-NODE CUT-SET ALGORITHM

Graph	Nodes	Elements	Configuration index	Number of cut-nodes	Total time (sec.)	Average cut-node time (msec.)
10	10	12	0.83	128	0.618	4.82
11	11	14	0.79	252	0.831	3.29
12	15	18	0.83	836	1.85	2.21

4.9.4 A practical application

The branch-node algorithm was used to list all the minimal cut-sets and the cut-node incidence of the electric distribution system of Fig. 11 taken from [79] . The results are listed in Fig. 16 which include cut-sets of all orders. Using the minimal cut-set approach to failure mode and effects analysis reported by Billinton and Grover(79), a complete analysis of the node failure effect on the reliability indices was made for the system configuration in Fig. 11.

To evaluate the reliability of a given electric distribution system, two variables are calculated for each node in the given system: the expected frequency of service interruptions and the expected duration of an interruption. It has been assume that loss of continuity at a node is the only mode of failure for the nodes. This condition can result from component permanent or temporary outages, overlapping permanent outages, temporary outages overlapping permanent outages and temporary and permanent outages overlapping preventive-maintenance outages.

MINIMAL CUTS					CUT-NODE INCIDENCE				
CUT	ELEMENTS				NODES				
1	1	7	8		2	3	4	5	6
2	1	2	3		2				
3	2	3	7	8	3	4	5	6	
4	2	3	4		3				
5	1	4			2	3			
6	4	7	8		4	5	6		
7	4	5			4				
8	2	3	5		3	4			
9	1	5			2	3	4		
10	5	7	8		5	6			
11	5	6	8		5				
12	6	7			6				
13	4	6	8		4	5			
14	2	3	6	8	3	4	5		
15	1	6	8		2	3	4	5	
16	7	8	9		3	4	5	6	
17	1	8	13		2	3	4	5	
18	9				2				
19	7	8	10		4	5	6		
20	2	3	8	13	3	4	5		
21	10				3				
22	4	9			3				
23	1	10			2				
24	1	11			2	3			
25	7	8	11		5	6			
26	4	8	13		4	5			
27	11				4				
28	5	9			3	4			
29	5	10			4				
30	2	3	11		3				
31	2	3	12		3	4			
32	1	12			2	3	4		
33	7	12			6				
34	5	8	13		5				
35	12				5				
36	13				6				
37	6	8	10		4	5			
38	6	8	11		5				
39	4	12			4				
40	6	8	9		3	4	5		

Figure 16. Minimal cut-set and cut-node for Fig. 11

The equations[79] used incorporate a 2-state weather model and consider the following situations:

(i) no repair during adverse weather

(ii) maintenance is not started if it could not be completed before a storm struck

The analysis assumes that all lines have the same reliability parameters indicated in section (4.8). The permanent and temporary failure rates of the nodes were varied from 0.01 to 1.0 times the line failure rates in steps of 0.05. The nodes outage durations were given the same values as line durations. All nodes were assumed to have the same reliability parameters. Figure 17 plots the resultant values of permanent and temporary node outage rates and permanent node outage duration expressed in per-unit of the respective values for perfectly reliable nodes against the node failure rate expressed as percentage of the line failure rate. These curves indicate that the error in the node reliability indices can be appreciable even for highly reliable nodes and is most serious if the node failure rates are a sizeable percentage of the line failure rates. The errors in node 5 are so high due to the fact that when assuming perfectly reliable nodes, node 5 is outaged only from occurrence of cut-sets of third order, therefore the results considering node failure have a high degree of nodal dominance.

4.10 Conclusion

The introduction of the concept of cut-node incidence allows the development of a systematic and efficient procedure for the enumeration of the minimal cut-sets between an origin and all other nodes in separable and nonseparable graphs. Each time a cut-set is generated we compute new reliability bounds for each node affected for that cut-set. When all cut-sets have been generated all node reliability bounds are available.

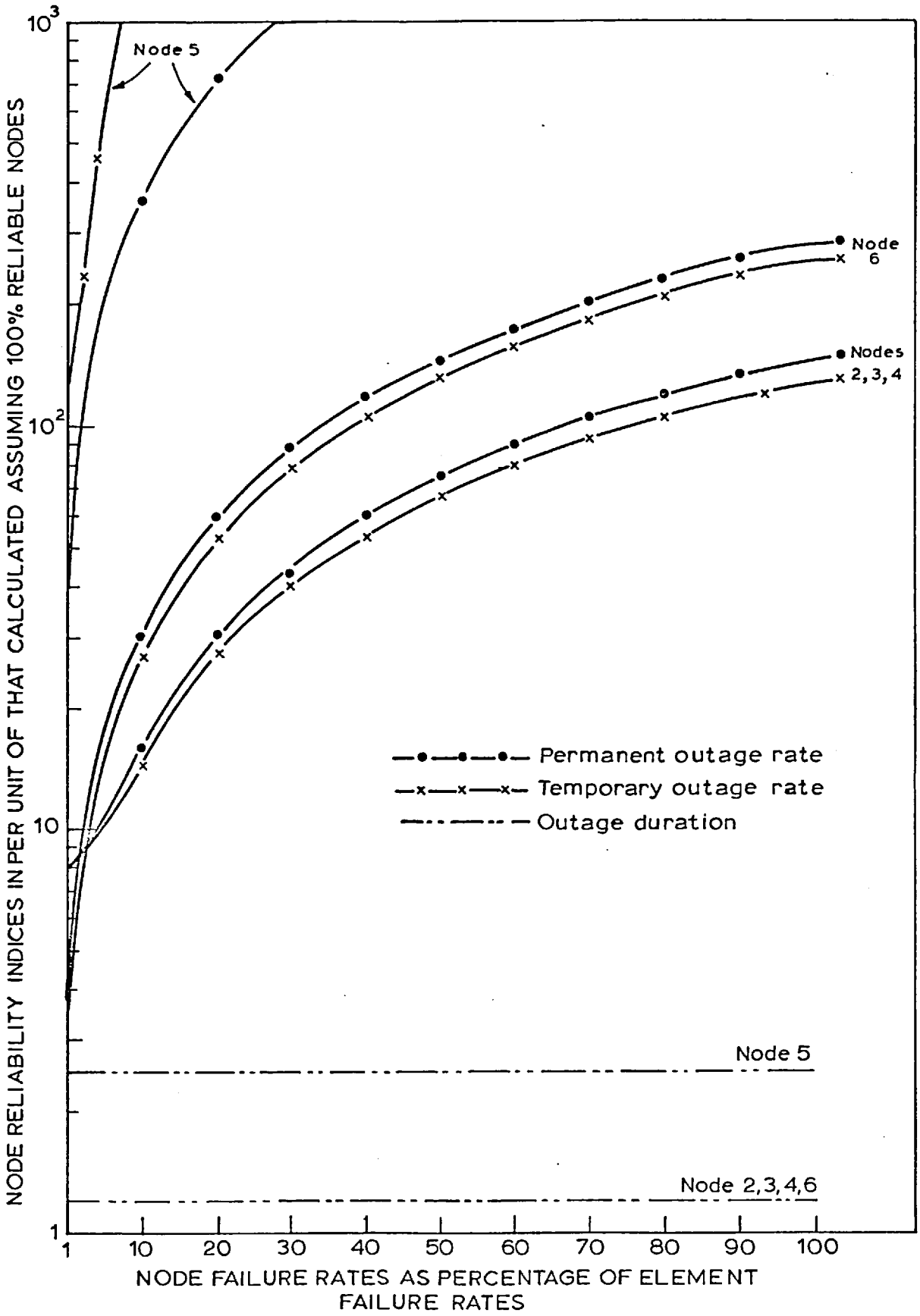


Fig. 17. Analysis of the influence of node outages on the reliability indices of the system in Fig. 11.

The set of proper cuts defined for the set of edges incident on each node proves to be a fundamental set from which all other cut-sets can be obtained by ring sum operations. These operations are reduced to a minimum if only those cuts with at least one edge in common are combined. In this sense the introduction of the index cut is an efficient criterion.

The cascade nature of acyclic directed graphs and the use of a breadth-first search method to fan out all edges at each node permits the listing of cut-sets in these graphs by a combinational approach that proves to be very efficient, compared with a similar technique reported in the literature.

Any physical system where the nodes are also subject to failures requires the enumeration of the feasible branch-node cut-sets for its reliability evaluation. Otherwise the results of such evaluation are meaningless. Unreliable nodes can be easily handled considering that an s-t branch-node cut-set is a minimal set of components (branches and nodes) that break all possible s-t paths. Once all branch cut-sets have been defined the use of a busbar-flag vector to indicate the terminal nodes of the branches in each cut-set expedites the generation of all possible branch-node combinations.

The efficiency of the minimal cut-set algorithms presented in this chapter cannot be measured entirely in terms of the number of nodes in the graph but rather in terms of the graph configuration index (number of nodes/number of branches). Graphs with high indices will be solved faster than those with low indices. The space requirement of the computer program implementation of the algorithms is determined by the number of cuts they may generate. A simple storage scheme based on the binary nature of the minimal cut-set and cut-node incidence matrices overcomes such restrictions.

CHAPTER V

VARIANCE AND APPROXIMATE CONFIDENCE LIMITS FOR THE PROBABILITY AND FREQUENCY OF SYSTEM FAILURE

5.1 Introduction

Reliability component data is generally obtained from two sources, the failure times of various items in a population placed on a life test[46] or field reports[47] listing operating hours of replaced components in equipment already in use. A very good way to present these data is to compute and plot either the failure density function or the failure rate as a function of time. Since both functions are continuous variables we first compute a piecewise-continuous failure density function and a failure rate from the data. Study of these piecewise-continuous functions is followed by the choice of a continuous model which fits the data satisfactorily and allows us to draw conclusions from test data on the behaviour of other similar components. Fitting a model to the data can be done either on the basis of experience, intuition and inspection of the appropriate graphs or through statistical techniques which can be used to efficiently process data and obtain "best" repeatable values for model parameters.

A point estimate of a parameter X is a single number which is our "best" estimate of the parameter. A measure of the variability of such a value is called the variance. If X varies relatively little or by a large amount, the variance will be relatively small or large respectively. If we are somewhat more realistic we may wish to quote an interval i.e., an interval estimate, into which the parameter probably falls. To be more precise we measure our sureness by the probability that the parameter falls within the interval. This probability is called the

"confidence"or "confidence coefficient". For a single component it is accepted that a bare statement of its estimated probability of failure during the period of operation is insufficient. A standard used for component procurement, the British Standard BS 4200 or the United States MIL STD 785, makes it very clear than an accuracy measure for the failure probability estimate is also necessary. This is usually in the form of a confidence limit.

Actually in many cases the interval estimate of the component parameters is only an intermediate step in solving the real problem of formulating an interval estimate for the system reliability. This chapter will focus on the development of computational methods for the variance and approximate confidence intervals for the probability and frequency of system failure taking into account the uncertainty of the component parameters. It is assumed that early testing has been performed on components and estimates and variances of component reliabilities are available i.e., this chapter presents methods for constructing approximate confidence limits on the system reliability indices from component test results.

5.2 Distribution of the failure and repair models

There exists a general agreement that the exponential distribution adequately describes the time-to-failure of most complex systems of solid state design but it is less relevant for electromechanical equipment where wear-out effects need to be considered. Hence, for a more satisfactory analysis it is necessary to consider the use of time-dependent distributions such as the Weibull distribution, whose density function is

$$f(t) = K\, t^{m}\, e^{-K\, t^{m+1} / m + 1} \qquad (5.1)$$

The parameter m determines the shape of the distribution and parameter K is a scale-change. When $m = 0$, the distribution becomes exponential. However, the exponential distribution, if not known to be the case, is usually a safe assumption for highly complex equipment[48]. It is also pertinent to remark that if routine maintenance is carried out sufficiently frequently then it can be assumed that defects due to time-dependent causes do not occur.

There is considerable controversy over the distribution of time-to-repair. The most commonly used are the exponential, the log-normal and the Erlang distributions. The log-normal has the serious limitation that it is very difficult to work with analytically in conjunction with another distribution. Since in many cases it is difficult, if not impossible, to distinguish between a sample from a log-normal distribution and a sample from an exponential distribution it would seem preferable to assume the exponential because it lends itself readily to an analytic approach. The Erlang family of service time distribution is defined by

$$f(t;\mu,k) = \frac{\mu(\mu t)^{k-1}}{(k-1)!} e^{-\mu t} \qquad (5.2)$$

where k is the number of stages. The limiting cases for this family occur when $k = 1$ and $k = \infty$. When $k = 1$, $f(t;\mu,k)$ reduces to the exponential and when $k = \infty$, $f(t;\mu,k)$ is a straight line intersecting the t-axis at $1/\mu$. The Gamma-distribution has a density function defined by

$$f(x) = \begin{cases} 0 & \text{for } x \leqslant 0 \\ \dfrac{\beta^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} & \text{for } x > 0 \end{cases} \qquad (5.3)$$

therefore, the exponential distribution (with $\alpha = 1$) and the so-called Erlang-distribution (in the case of integer α) are particular cases of a Γ-distribution.

Both the nonrepairable and the repairable approaches described in this chapter assume the exponential distribution for the time-to-failure and for the time-to-repair. However the repairable case is also applicable if the Weibull or Gamma distributions are used for any one of both times.

5.3 The exponential distribution in reliability

The exponential-distribution plays a key role in reliability and calculations because it describes so well the behaviour of components and systems in their useful life period i.e., when they display approximately constant failure rates. Its great advantage over other any statistical distributions is the single parameter λ, or its reciprocal m, which fully and completely describes a given exponential function.

The exponential density function is given by

$$f(t) = \lambda e^{-\lambda t} \qquad t \geqslant 0, \quad \lambda > 0 \tag{5.4}$$

where λ is the constant failure rate. A further advantage of the exponential distribution is that it is independent of the age of a component or system as long as they are operated during the period of constant failure rate.

5.3.1 Estimate and variance of the parameter

The fact that the exponential distribution is a single parameter distribution makes reliability testing of exponential equipment comparatively simple because all that is needed is to determine the value of λ by a test.

Suppose we have a sample $X = (x_1, x_2, .., x_n)$ of n independent observations drawn randomly from an exponential distribution with the density function of Eqn. (5.4), where λ is an unknown parameter and we

wish to estimate it using the sample data. Several criteria of estimation[49] have led to estimates of λ of the form d_n/S, where $S = \sum_1^n x_i$ is a sufficient statistic for λ and d_n is a function of n. The whole topic of estimation and testing schemes are discussed in many places, e.g. [50] or [51] . The most frequently used estimators are the maximum likelihood estimator (MLE) and the minimum variance unbiased estimator (MVUE).

Lloyd and Lipow[52] summarized the methods for estimating the parameters of the gamma distribution from complete samples by the method of maximum likelihood. Epstein and Sobel[46] derived MLE for the parameter of the exponential distribution based on censored samples. Sathe and Varde[53] derived the MVUE for the same distributions, giving the expected value and the variance for the parameter in each case.

5.3.2 Reliability estimate and variance

Let $t_1, t_2, \ldots, t_n$ be identically distributed independent random variables having an exponential distribution. These random variables represent the life-lengths of n identical systems or components. Let θ denote the expected life-length, or mean time between failures (MTBF) of the systems or components under consideration; $0 < \theta < \infty$. The reliability function of the system or component is

$$R(t,\theta) = \exp(-t/\theta) \quad 0 \leq t < \infty \tag{5.5}$$

The commonly used estimate of reliability in the exponential case is the maximum likelihood estimate $\hat{R}$, that for a given mission time is

$$\hat{R} = \exp(-t_m/\hat{\theta}) \tag{5.6}$$

where $\hat{\theta}$ is the sample mean given by

$$\hat{\theta} = \sum_{i=1}^{n} t_i/n \tag{5.7}$$

Though the estimate $\hat{R}$ has all the properties of MLE it has one serious drawback for application in reliability: it is biased. Pugh[54] has shown that the best estimate of R is not $\hat{R}$ but

$$\hat{R}^* = (1 - \frac{t_m}{n\hat{\theta}})^{n-1} \tag{5.8}$$

where $\hat{R}^* = 0$ if $\hat{\theta} \leqslant t_m/n$.

The probability density function of the complete sufficient statistic $S = \sum t_i$ is gamma-distributed $g(1/\theta,n)$. Accordingly

$$E\{\hat{R}^{*2}(t_m,S)\} = \frac{1}{(n-1)!} \int_{t_m/\theta}^{\infty} (1 - \frac{t_m}{\theta x})^{2n-2} e^{-x} x^{n-1} dx \tag{5.9}$$

and the variance of the MVU estimator $\hat{R}^*(t_m,S)$ is obtained by subtracting $e^{-2t_m/\theta}$ from Eqn. (5.9). For numerical computations it is more convenient to apply the Gauss-Laguerre quadrature method[55] for determining the integral of Eqn. (5.9).

5.4 Minimal cut-set basic formulae

In the minimal cut-set approach to system reliability the probability of system failure[30] is expressed as

$$P_f = \Pr\{\overline{C}_1 + \overline{C}_2 + \ldots + \overline{C}_n\} \tag{5.10}$$

where the + indicates the logic sum or union. The mutually exclusive approximation provides an upper bound to the probability of system failure given by

$$P_f \leqslant \sum_{i=1}^{NMC} \Pr\{\overline{C}_i\} \tag{5.11}$$

where NMC is the total number of minimal cut-set.

Similarly, the frequency of system failure[16] is expressed as

$$f_f = f \left\{ S_1 + S_2 + \dots + S_n \right\} \tag{5.12}$$

where each S_i represents a subset of the state space S formed by all $\overline{C}_i$ and

$$f(S_i) = Pr \left\{ \overline{C}_i \right\} \cdot \sum_{j \varepsilon C_i} \mu_j$$

The mutually exclusive approximation yields an upper found for f_f given by

$$f_f \leqslant \sum_{i=1}^{NMC} \left\{ Pr \quad (\overline{C}_i) \cdot \mu_i \right\} \tag{5.13}$$

where $\overline{\mu}_i$ is the sum of μ_j's over all $j \varepsilon C_i$. The disjoint approximation is preferred here because the simple relationships provided by Eqns. (5.11) and (5.13) with respect to $Pr \left\{ \overline{C}_i \right\}$ and $f \left\{ S_i \right\}$ respectively, ease the computation of the expected value and variance of P_f and f_f.

If λ_j and μ_j represent the failure and repair rate of the jth element the steady state probabilities[54] of that element being in the up state P_{u_j} or in the down state P_{d_j} are

$$P_{u_j} = \frac{\mu_j}{\lambda_j + \mu_j} \tag{5.14}$$

and

$$P_{d_j} = \frac{\lambda_j}{\lambda_j + \mu_j} \tag{5.15}$$

P_{u_j} and P_{d_j} can be interpreted as follows: As the operating time of the jth element approaches infinity, the fraction of time in the up state approaches P_{u_j} and the fraction of time in the down state approaches P_{d_j}.

For the nonrepairable condition of an element with the exponential failure model, the probability of failure[17] in the time interval from 0 to t_m is given by

$$Q_j(t_m) = 1 - e^{-\lambda_j t_m} \tag{5.16}$$

and the probability of system failure for the same interval using the minimal cut-set approach can be obtained from

$$P_f = \sum_{i=1}^{NMC} \left\{ \prod_{j \varepsilon i} Q_j(t_m) \right\} \tag{5.17}$$

For the repairable case, the system is assumed to have reached an equilibrium (steady state) condition of failure and repair. The steady state probability and frequency of system failure given by the minimal cut-set approach are obtained from

$$P_f = \sum_{i=1}^{NMC} \left\{ \prod_{j \varepsilon i} P_{d_j} \right\} \tag{5.18}$$

$$f_f = \sum_{i=1}^{NMC} \left\{ \prod_{j \varepsilon i} P_{d_j} \cdot \sum_{j \varepsilon i} \mu_j \right\} \tag{5.19}$$

For both repairable and nonrepairable conditions the highest contingency to be evaluated will be the third order cut-sets. Higher orders are assumed to produce negligible effects on the reliability indices.

5.5 Problem statement

We are concerned with the general problem of estimating the probability and frequency of failure that a complex system has during its operational life. The system will consist of a number of components,

logically interconnected in some fashion, with particular patterns of component failure causing system failure.

Component failure and repair rates obtained from life-tests or field reports are subject to uncertainty and may most properly be thought of as random variables. It follows from Eqns. (5.17) to (5.19) that P_f and f_f which are functions of these parameters are also subject to uncertainty and must be considered as random variables too. Presently, the expected or average values of the component parameters are used to compute the expected values of the P_f and f_f random variables. The expected values of P_f and f_f provide no information as to the range of values over which P_f and f_f might vary due to variations in component failure and repair rates. Such analysis requires a method for computing not only a point-estimate of both reliability indices but also interval estimates. The interval estimate of P_f and f_f will give upper and lower bounds such that these random variables lie between these limits with a specified probability.

The exact solution of this problem requires complete information about the internal variables λ_j, μ_j. This means that the distribution of each λ_j and μ_j must be known and how changes occur as a function of time. The amount of information required for the exact solution generally will not be available. Also as Eqns. (5.17) to (5.19) involve nonlinear operations, a solution in closed form is generally not possible. These are the reasons why the exact solution is not discussed in this thesis.

Considering that the parallel-series arrangement of components in the minimal cut-set approach adequately represent the system with statistically independent components, we now show how these interval estimates for the probability and frequency of system failure can be calculated by approximate methods.

A deterministic (nonprobabilistic) approach to this problem is first discussed. Two different types of approximation will be presented. The first technique involves the orthodox statistical procedure of characterizing distributions by their low order moments and uses the conservative Chebyshev inequality to bound the probabilities that the random variables lie within a certain range. The second technique is to apply Monte Carlo simulation based on common probability models, with which component test data may be translated into approximate system reliability limits at any confidence level.

5.6 Review of the literature

Methods of obtaining confidence bounds on the reliability of non-maintained systems are surveyed and numerical comparisons are given by Mann[56]. It is assumed that failure data have been collected from life tests performed on prototypes of the various components which make up the system, but that the system has not been tested as a whole for economic reasons or simply that it is virtually impossible to test the entire system without destroying it. This paper represents a very good summary of the state of the art and discusses methods applicable to series and/or parallel as well as more logically complex systems.

Rosenblatt[57] treats the problem of estimation of a probability R interpreted as system reliability when the estimate is to be based on data obtained from component tests. The problem is formulated in a general way leading to a widely applicable method for distribution-free estimation of R and calculation of approximate nonparametric confidence intervals. It discusses comparisons with exact and alternative approximate methods. The method is also well discussed by Mann[56].

Murchland and Weber[58] describe a method for the variance of the probability of failure of a complex system represented by a fault-tree with independent components using a moment method when the components are not repairable. In the repairable case, a method for approximately determining the variance of the expected number of failures is indicated, for which the authors suggest the use of Chebyshev inequality rather than picking any particular assumptions concerning the distribution of the unbiased simulation estimator of system reliability. Their bounds involve the expression for the exact rather than the asymptotic variance of the estimator of the system reliability. Their procedure presents no particular computational difficulties as long as the system remains logically simple. Calculation of their variance estimate for the reliability estimator becomes very complicated, however, as the system increases in complexity.

Burnett and Wales[59] present a method for constructing statistical confidence limits for the system reliability from component test results. They suggest Monte Carlo simulation of the fiducial distribution of system reliability given the subsystem failure for the series-system model and also for other models which are logically more complex. The method developed is only for the case of components with exponentially distributed failures and assumes: (i) individual components fail independently, (ii) a mathematical expression relating system reliability and component reliabilities can be written, (iii) estimates of component reliabilities are available from earlier testing and (iv) these estimates of omponent reliabilities have known sampling distributions. The method generates the numerical sampling distributions of component reliability estimates from which is selected a value for each component that is substituted into the mathematical relationship between system-component

reliabilities to give one value for the system reliability. This process is repeated until the numerical sampling distribution of the system reliability estimate is generated. Confidence limits are then computed on this numerical distribution.

Levy and Moore[60] extended the previous Monte Carlo technique to systems whose component failures follow the normal, log-normal, gamma and Weibull probability distribution. They also assume that individual components fail independently and that failure data is available from component tests. The parameter estimates are assumed to be obtained by the method of maximum likelihood based on life tests from a complete sample or from a censored sample where the distribution of the estimator is known.

Both techniques[59],[60] based in the fiducial method are optimum for one component and approach the optimum confidence bounds as the number of failures increases for all components. They are conservative in general but provide a method for obtaining confidence limits of systems composed of components whose failure patterns are represented by different families of mathematical models. Both are designed specially for series and/or parallel combinations but complex configurations must be broken down, by application of Bayes theorem, into combinations of these types.

Kamat and Riley[61] present a Monte Carlo simulation procedure for non-repairable systems represented by its minimal tie-sets when the components are assumed to be statistically independent. The method is a synthetic one, in that the failure times of individual components are generated and then used to determine the success or failure of the system. Each of these component failure times is converted to the Boolean state representation of success or failure by comparing it with the required operation time. The success or failure of each component

along any given minimal tie-set is checked from its state representation. If all the components along the tie-set are a success, the system is a success. If at least one of these components is a failure, another minimal tie-set is checked. This procedure is continued until either the system success has been identified or all the minimal tie-sets have been checked with each tie-set having at least one failed component. Replications of this procedure will yield n_s success and n_f failures of the system and the reliability estimate for a mission time t_m is given by $R(t_m) = n_s/(n_s + n_f)$. The confidence limits are obtained using the statistical normal approximation to the binomial distribution. The confidence interval obtained is narrowed by increasing the number of replications.

5.7 Deterministic approach

The simple first approach to the problem of section (5.5) is to ignore the probabilistic nature of the problem and attempt a deterministic solution. One technique is to assume that component parameters have distributions entirely concentrated between minimum and maximum values. This technique, called here the worst-case approach, makes an honest attempt to determine how much change there will be in the reliability indices if all the components are at their extremes and combine in the worst possible manner.

We restrict our analysis to the contribution to the reliability indices of minimal cut-sets up to the third order.

Let us consider the case of repairable components where the steady-state probability of the i-th component being in the down state is given by

$$P_{d_i} = \frac{\lambda_i}{\lambda_i + \mu_i}$$

If σ_λ and σ_μ represent the standard deviations of the failure and repair rates respectively, it seems reasonable to use $\lambda_i \pm \sigma_{\lambda_i}$ for $\lambda_{i_{Max}}$ and $\lambda_{i_{min}}$, $\mu_i \pm \sigma_{\mu_i}$ for $\mu_{i_{Max}}$ and $\mu_{i_{min}}$, respectively. The minimal cut-set approach to the probability and frequency of system failure are given by Eqns. (5.17) to (5.19). Table I summarises the resulting expressions for minimal cut-sets up to third order of the steady-state probability of failure. Capital letters refer to maximum values and minor ones to minimum values. Symbol ($\wedge$) refers to the expected value of the parameter.

TABLE I

MINIMUM AND MAXIMUM VALUES FOR THE STEADY STATE PROBABILITY OF FAILURE

Cut-set order	Point-estimate	minimum	Maximum
1	$\frac{1}{1 + \hat{\mu}/\hat{\lambda}}$	$\frac{1}{1 + \hat{\mu}_M/\hat{\lambda}_m}$	$\frac{1}{1 + \hat{\mu}_m/\hat{\lambda}_M}$
2	$\prod_{j=1}^{2} \frac{1}{1 + \hat{\mu}_j/\hat{\lambda}_j}$	$\prod_{j=1}^{2} \frac{1}{1 + \hat{\mu}_{j_M}/\hat{\lambda}_{j_m}}$	$\prod_{j=1}^{2} \frac{1}{1 + \hat{\mu}_{j_m}/\hat{\lambda}_{j_M}}$
3	$\prod_{j=1}^{3} \frac{1}{1 + \hat{\mu}_j/\hat{\lambda}_j}$	$\prod_{j=1}^{3} \frac{1}{1 + \hat{\mu}_{j_M}/\hat{\lambda}_{j_m}}$	$\prod_{j=1}^{3} \frac{1}{1 + \hat{\mu}_{j_m}/\hat{\lambda}_{j_M}}$

Referring to the probability of failure of the system the idea is to combine the minimum and maximum values of the minimal cut-sets in the worst possible manner. A little thought shows that these values are given by

$$P_{f_M} = \sum_{i=1}^{NMC} Pr \left\{ C_i \right\}_M \tag{5.20}$$

$$P_{f_m} = \sum_{i=1}^{NMC} Pr \left\{ C_i \right\}_m \tag{5.21}$$

Similarly for the probability of failure of the system in the time interval from 0 to t_m, it seems reasonable to use, for each component, $Q_j(t_m) \pm \sigma_{Q_j}$ for $Q_j(t_m)_{Max}$ and $Q_j(t_m)_{min}$ respectively. In such cases general expressions for minimal cut-sets will be

$$P_{f_M}(t_m) = \sum_{i=1}^{NMC} \prod_{j \epsilon i} Q_j(t_m)_M \tag{5.22}$$

$$P_{f_m}(t_m) = \sum_{i=1}^{NMC} \prod_{j \epsilon i} Q_j(t_m)_m \tag{5.23}$$

However, all one can say is that P_f probably lies between the values P_{f_M} and P_{f_m}. Since the statistics are unknown no confidence coefficient can be stated for this reliability index.

If the reliability index is a rather complicated function involving many parameters, as it is in the case for the frequency of system failure, it is hard to decide by inspection which combination of values gives a minimum and a maximum value of the index. In such a case, approximating the reliability index by the first few terms of a Taylor series expansion simplifies the problem. For the purposes of the deterministic approach it should be sufficient to retain the constant and the first-order terms and for a function of n variables these terms are given by

$$f(\Delta x_1, \Delta x_2, \ldots, \Delta x_n) = f_{bv} + \sum_{i=1}^{n} (\partial f / \partial x_i)_{bv} \Delta x_i$$

For the minimal cut-set approach we need to evaluate the partial derivatives for cuts up to the third-order. Both the function and its partial derivatives are to be evaluated at the best value (bv) of the variables.

First-order cut-set

A point-estimate is given by

$$f_{C1} = \frac{\hat{\lambda}_1}{\hat{\lambda}_1 + \hat{\mu}_1} \cdot \hat{\mu}_1 \tag{5.24}$$

The coefficients from the first partial derivatives designated by b_{f_1} and b_{f_2} are

$$b_{f_1} = (\partial f_{C1}/\partial \lambda_1)_{bv} = (\hat{\mu}_1/\hat{\lambda}_1 + \hat{\mu}_1)^2 \tag{5.25}$$

$$b_{f_2} = (\partial f_{C1}/\partial \mu_1)_{bv} = (\hat{\lambda}_1/\hat{\lambda}_1 + \hat{\mu}_1)^2 \tag{5.26}$$

Using worst-case techniques the minimum and maximum values for the frequency of the first-order cut-set will be

$$f_{C1_m} = \hat{f}_{C1} - b_{f_1} \Delta\lambda_1 - b_{f_2} \Delta\mu_1 \tag{5.27}$$

$$f_{C1_M} = \hat{f}_{C1} + b_{f_1} \Delta\lambda_1 + b_{f_2} \Delta\mu_1 \tag{5.28}$$

where $\Delta\lambda_1$ and $\Delta\mu_1$ are substituted as positive quantities, made equal in this case to the standard deviation of the respective parameter.

Second-order cut-set

A single-point estimate is

$$\hat{f}_{C2} = \frac{\hat{\lambda}_1}{\hat{\lambda}_1 + \hat{\mu}_1} \cdot \frac{\hat{\lambda}_2}{\hat{\lambda}_2 + \hat{\mu}_2} \cdot (\hat{\mu}_1 + \hat{\mu}_2) \tag{5.29}$$

The coefficients from the first partial derivatives are

$$b_{f1} = (\partial f_{C2}/\partial \lambda_1)_{bv} = \frac{\hat{\lambda}_2}{\hat{\lambda}_2 + \hat{\mu}_2} \cdot \frac{\hat{\mu}_1}{(\hat{\lambda}_1 + \hat{\mu}_1)^2} \cdot (\hat{\mu}_1 + \hat{\mu}_2) \tag{5.30}$$

$$b_{f2} = (\partial f_{C2}/\partial \mu_1)_{bv} = \frac{\hat{\lambda}_2}{\hat{\lambda}_2 + \hat{\mu}_2} \cdot \frac{\hat{\lambda}_1}{(\hat{\lambda}_1 + \hat{\mu}_1)^2} \cdot (\hat{\lambda}_1 - \hat{\mu}_2) \tag{5.31}$$

$$b_{f3} = (\partial f_{C2}/\partial \lambda_2)_{bv} = \frac{\hat{\lambda}_1}{\hat{\lambda}_1 + \hat{\mu}_1} \cdot \frac{\hat{\mu}_2}{(\hat{\lambda}_2 + \hat{\mu}_2)^2} \cdot (\hat{\mu}_1 + \hat{\mu}_2) \tag{5.32}$$

$$b_{f4} = (\partial f_{C2}/\partial \mu_2)_{bv} = \frac{\hat{\lambda}_1}{\hat{\lambda}_1 + \hat{\mu}_1} \cdot \frac{\hat{\lambda}_2}{(\hat{\lambda}_2 + \hat{\mu}_2)^2} \cdot (\hat{\lambda}_2 - \hat{\mu}_1) \tag{5.33}$$

Using worst-case techniques the minimum and maximum values for the frequency of the second-order cut-set are

$$f_{C2_m} = \hat{f}_{C2} - \Delta f_{C2} \tag{5.34}$$

$$f_{C2_M} = \hat{f}_{C2} + \Delta f_{C2} \tag{5.35}$$

where Δf_{C2} is given by

$$\Delta f_{C2} = b_{f1}\Delta\lambda_1 + b_{f2}\Delta\mu_1 + b_{f3}\Delta\lambda_2 + b_{f4}\Delta\mu_2 \tag{5.36}$$

The basic assumption that component repair rates are much larger than failure rates would give a negative sign to coefficients b_{f2} and b_{f4}. Therefore, $\Delta\lambda_j$ and $\Delta\mu_j$, $j = 1,2$ are substituted equal

to the respective standard deviation of the parameter with the sign to combine in the worst possible manner.

Third-order cut-set

A single-point estimate is

$$f_{C3} = \frac{\hat{\lambda}_1}{\hat{\lambda}_1 + \hat{\mu}_1} \cdot \frac{\hat{\lambda}_2}{\hat{\lambda}_2 + \hat{\mu}_2} \cdot \frac{\hat{\lambda}_3}{\hat{\lambda}_3 + \hat{\mu}_3} \cdot (\hat{\mu}_1 + \hat{\mu}_2 + \hat{\mu}_3) \tag{5.37}$$

The coefficients from the first partial derivatives with respect to λ_i and μ_i, i=1,2,3 designated by b_{fj} and b_{fk}, j = 1,3,5; k = 2,4,6 respectively, are

$$b_{f1} = \frac{\hat{\lambda}_2}{\hat{\lambda}_2 + \hat{\mu}_2} \cdot \frac{\hat{\lambda}_3}{\hat{\lambda}_3 + \hat{\mu}_3} \cdot \frac{\hat{\mu}_1}{(\hat{\lambda}_1 + \hat{\mu}_1)^2} \cdot (\hat{\mu}_1 + \hat{\mu}_2 + \hat{\mu}_3) \tag{5.38}$$

$$b_{f2} = \frac{\hat{\lambda}_2}{\hat{\lambda}_2 + \hat{\mu}_2} \cdot \frac{\hat{\lambda}_3}{\hat{\lambda}_3 + \hat{\mu}_3} \cdot \frac{\hat{\lambda}_1}{(\hat{\lambda}_1 + \hat{\mu}_1)^2} \cdot (\hat{\lambda}_1 - \hat{\mu}_2 - \hat{\mu}_3) \tag{5.39}$$

$$b_{f3} = \frac{\hat{\lambda}_1}{\hat{\lambda}_1 + \hat{\mu}_1} \cdot \frac{\hat{\lambda}_3}{\hat{\lambda}_3 + \hat{\mu}_3} \cdot \frac{\hat{\mu}_2}{(\hat{\lambda}_2 + \hat{\mu}_2)^2} \cdot (\hat{\mu}_1 + \hat{\mu}_2 + \hat{\mu}_3) \tag{5.40}$$

$$b_{f4} = \frac{\hat{\lambda}_1}{\hat{\lambda}_1 + \hat{\mu}_1} \cdot \frac{\hat{\lambda}_3}{\hat{\lambda}_3 + \hat{\mu}_3} \cdot \frac{\hat{\lambda}_2}{(\hat{\lambda}_2 + \hat{\mu}_2)^2} \cdot (\hat{\lambda}_2 - \hat{\mu}_1 - \hat{\mu}_3) \tag{5.41}$$

$$b_{f5} = \frac{\hat{\lambda}_1}{\hat{\lambda}_1 + \hat{\mu}_1} \cdot \frac{\hat{\lambda}_2}{\hat{\lambda}_2 + \hat{\mu}_2} \cdot \frac{\hat{\mu}_3}{(\hat{\lambda}_3 + \hat{\mu}_3)^2} \cdot (\hat{\mu}_1 + \hat{\mu}_2 + \hat{\mu}_3) \tag{5.42}$$

$$b_{f6} = \frac{\hat{\lambda}_1}{\hat{\lambda}_1 + \hat{\mu}_1} \cdot \frac{\hat{\lambda}_2}{\hat{\lambda}_2 + \hat{\mu}_2} \cdot \frac{\hat{\lambda}_3}{(\hat{\lambda}_3 + \hat{\mu}_3)^2} \cdot (\hat{\lambda}_3 - \hat{\mu}_1 - \hat{\mu}_2) \tag{5.43}$$

The minimum and maximum values for the frequency of the third-order minimal cut-set are

$$f_{C3_m} = f_{C3} - \Delta f_{C3} \tag{5.44}$$

$$f_{C3_M} = f_{C3} + \Delta f_{C3} \tag{5.45}$$

where

$$\Delta f_{C3} = \sum_{i=1}^{3} b_{fj}\Delta\lambda_i + \sum_{i=1}^{3} b_{fk}\Delta\mu_i \tag{5.46}$$

Again $\Delta\lambda_i$ and $\Delta\mu_i$ are combined to give the worst possible values.

The minimum and maximum values for the frequency of system failure are

$$f_{f_m} = \sum_{i=1}^{NMC} f_{Ci_m} \tag{5.47}$$

$$f_{f_M} = \sum_{i=1}^{NMC} f_{Ci_M} \tag{5.48}$$

5.8 Approximate statistical approach - Moment method

To determine $(1 - \alpha)$ symmetrical confidence limits on a random variable x, the $\alpha/2$ and $(1 - \alpha/2)$ percentiles of the cumulative distribution F(x) must be found by solving the equations

$$F(x_0) - F(0) = \alpha/2$$

$$F(x_1) - F(0) = 1 - \alpha/2$$

where

α is the level of significance

x_0 is the $\alpha/2$ percentile of the x distribution

x_1 is the $(1 - \alpha/2)$ percentile of the x distribution

Thus, a knowledge of $F(x)$ contains all the information necessary to compute confidence limits for a random variable. One way to alleviate the difficulties of the exact solution is by using the statistical procedure of characterizing a distribution function by their low order moments.

Since the probability and frequency of system failure given by Eqns. (5.17) to (5.19) are non-linear functions in each component parameter, approximate formulae can be given for the function moments in terms of the component moments by the Taylor series approximation technique.

Considering a general function $f(x_1,x_2,..,x_n)$ one needs to expand the function in a Taylor series about the mean values $(\gamma_1, \gamma_2,.., \gamma_n)$, form the appropriate powers of $x_i - \gamma_i$ by the multinomial theorem and to take expectations hoping that as many terms as possible will be zero because they contain a first power of $x_i - \gamma_i$. From such an expansion we compute the first and second moment of the function which gives us the expected value and the variance for the same function. Since the distribution is unknown we bound the probabilities that the function lies within a certain range using Chebyshev's inequality[62].

5.8.1 Low order moments formulae

The first- and second-order terms of the Taylor series expansion of a function $f(x_1,x_2,..,x_n)$ yield

$$f(\Delta x_1,\Delta x_2,..,\Delta x_n) = a + \sum_{i=1}^{n} b_i \Delta x_i + 1/2 \sum_{i=1}^{n} c_i \Delta x_i^2 + \sum_{\substack{i=1 \\ i \neq j}}^{n} \sum_{j=1}^{n} d_{ij} \Delta x_i \Delta x_j \qquad (5.49)$$

where $\Delta x_i^n = (x_i - \gamma_i)^n$, γ_i is the expected value $E(x_i)$ and the constant term a is the function evaluated at the best value of the variables. The coefficients b_i, c_i and d_{ij} are the partial derivatives $(\partial f/\partial x_i)$, $(\partial^2 f/\partial x_i^2)$ and $(\partial^2 f/\partial x_i \partial x_j)$ respectively evaluated at the best values, $x_i = \gamma_i$.

Computing the expected value or first moment E(f) from Eqn. (5.49) yields

$$E(f) = a + \frac{1}{2} \sum_{i=1}^{n} c_i \sigma_i^2 + \sum_{\substack{i=1 \\ i \neq j}}^{n} \sum_{j=1}^{n} d_{ij} \operatorname{cov}(x_i, x_j) \qquad (5.50)$$

The computation of the variance is much more involved and is made by computing $E(f^2)-E^2(f)$. If for any reason we agree to neglect higher than second-order moments, Var(f) can be written in general as

$$\operatorname{Var}(f) = \sum_{i=1}^{n} b_i^2 \sigma_1^2 + 2 \sum_{\substack{i=1 \\ i \neq j}}^{n} \sum_{j=1}^{n} b_i b_j \operatorname{cov}(x_i, x_j) \qquad (5.51)$$

Proper simplifications for the case under study will allow the use of Eqns. (5.50) and (5.51) efficiently to compute the mean and variance of a function of n random variables.

5.8.2 Expected value and variance

The probability and frequency of system failure given by Eqns. (5.17) to (5.19) are non-linear functions of the failure and repair rates of the components of the system. Therefore, to compute the expected value and the variance of both indices we need to make a linearization of those equations by expanding them in a truncated Taylor series.

The assumptions made in this reliability case are :

(i) component data is available in terms of their expected values and variances,

(ii) components are statistically independent. Using the same notation of section 5.8.1 we have

$$\mathrm{cov}(x_1, x_j) = 0$$
$$E(\Delta x_i \Delta x_j^n) = E(\Delta x_i)\, E(\Delta x_j^n) = 0$$
$$E(\Delta x_1^2 \Delta x_j^2) = \sigma_1^2\, \sigma_j^2$$

(iii) component repair rates are much larger than their failure rates, then the c and d coefficients used in the Taylor series expansion are smaller than the b coefficients.

Under such assumptions the expected values and variances of the steady-state probability and frequency of system failure are given by

$$E(P_f) = (P_f)_{bv} + \frac{1}{2} \sum_{i=1}^{n} c_{p\lambda_i}\, \sigma_{\lambda_i}^2 + \frac{1}{2} \sum_{i=1}^{n} c_{p\mu_i}\, \sigma_{\mu_i}^2 \tag{5.52}$$

$$\mathrm{Var}(P_f) = \sum_{i=1}^{n} b_{p\lambda_i}\, \sigma_{\lambda_i}^2 + \sum_{i=1}^{n} b_{p\mu_i}\, \sigma_{\mu_i}^2 \tag{5.53}$$

$$E(f_f) = (f_f)_{bv} + \frac{1}{2} \sum_{i=1}^{n} c_{f_{\lambda i}}\, \sigma_{\lambda i}^2 + \frac{1}{2} \sum_{i=1}^{n} c_{f_{\mu i}}\, \sigma_{\mu i}^2 \tag{5.54}$$

$$\mathrm{Var}(f_f) = \sum_{i=1}^{n} b_{f_{\lambda i}}\, \sigma_{\lambda_i}^2 + \sum_{i=1}^{n} b_{f_{\mu i}}\, \sigma_{\mu_i}^2 \tag{5.55}$$

The expected value and the variance of the probability of system failure in the interval time from 0 to t_m are given by

$$E\{P_f(t_m)\} = P_{f_{bv}}(t_m) + \frac{1}{2} \sum_{i=1}^{n} c_{Q_i}(t_m)\, \sigma_{Q_i}^2(t_m) \tag{5.56}$$

$$\mathrm{Var}\{P_f(t_m)\} = \sum_{i=1}^{n} b_{Q_i}(t_m)\, \sigma_{Q_i}^2(t_m) \tag{5.57}$$

The linearity of both reliability indices with respect to the minimal cut-set values $Pr\{\overline{C}_i\}$ and $f\{S_i\}$ in Eqns. (5.11) and (5.13) suggest a cut-oriented procedure for the computations of the b and c coefficients of Eqns. (5.52) to (5.57).

Non-repairable components (t_m mission time)

First-order cut-set

Let i be the component contained in the cut set

$$b_{Q_i}(t_m) = 1$$

$$c_{Q_i}(t_m) = 0$$

$$E\{P_{fC1}\} = \hat{Q}_i(t_m) \tag{5.58}$$

$$Var\{P_{fC1}\} = Var\{\hat{Q}_i(t_m)\} \tag{5.59}$$

Second-order cut-set

Let i and j be the components of the cut-set

$$b_{Q_i}(t_m) = \hat{Q}_j(t_m)$$

$$b_{Q_j}(t_m) = \hat{Q}_i(t_m)$$

$$c_{Q_i}(t_m) = c_{Q_j}(t_m) = 0$$

$$E\{P_{fC2}\} = \hat{Q}_i(t_m)\,\hat{Q}_j(t_m) \tag{5.60}$$

$$Var\{P_{fC2}\} = \sum_{\substack{k=i,j \\ l=j,i}} \hat{Q}_k(t_m)\, Var\{\hat{Q}_l(t_m)\} \tag{5.61}$$

Third-order cut-set

Let i, j and k be the components in the cut set

$$b_{Q_i}(t_m) = \hat{Q}_j(t_m)\,\hat{Q}_k(t_m)$$

$$b_{Q_j}(t_m) = \hat{Q}_i(t_m)\,\hat{Q}_k(t_m)$$

$$b_{Q_k}(t_m) = \hat{Q}_i(t_m)\,\hat{Q}_j(t_m)$$

$$c_{Q_i}(t_m) = c_{Q_j}(t_m) = c_{Q_k}(t_m) = 0$$

$$E\left\{P_{fC3}\right\} = \hat{Q}_i(t_m)\,\hat{Q}_j(t_m)\,\hat{Q}_k(t_m) \qquad (5.62)$$

$$\mathrm{Var}\left\{P_{fC3}\right\} = \sum_{\substack{l=j,k,i\\ n=k,i,j\\ p=i,j,k}} \hat{Q}_l(t_m)\,\hat{Q}_n(t_m)\,\mathrm{Var}\left\{\hat{Q}_p(t_m)\right\} \qquad (5.63)$$

Repairable components

First-order cut-set

Let i be the component in the cut

$$\begin{aligned}
b_{p\lambda_i} &= \hat{\mu}_i/(\hat{\lambda}_i + \hat{\mu}_i)^2\\
b_{p\mu_i} &= -\hat{\lambda}_i/(\hat{\lambda}_i + \hat{\mu}_i)^2\\
c_{p\lambda_i} &= -2\hat{\mu}_i/(\hat{\lambda}_i + \hat{\mu}_i)^3\\
c_{p\mu_i} &= 2\,\hat{\lambda}_i/(\hat{\lambda}_i + \hat{\mu}_i)^3 \qquad (5.64)\\
b_{f\lambda_i} &= \hat{\mu}_i^2/(\hat{\lambda}_i + \hat{\mu}_i)^2\\
b_{f\mu_i} &= \hat{\lambda}_i^2/(\hat{\lambda}_i + \hat{\mu}_i)^2\\
c_{f\lambda_i} &= -2\,\hat{\mu}_i^2/(\hat{\lambda}_i + \hat{\mu}_i)^3\\
c_{f\mu_i} &= -2\,\hat{\lambda}_i^2/(\hat{\lambda}_i + \hat{\mu}_i)^3
\end{aligned}$$

Second-order cut-set

Let i and j be the components in the cut-set

$$b_{p\lambda_i} = (\hat{\lambda}_j/\hat{\lambda}_j + \hat{\mu}_j) \cdot \hat{\mu}_i/(\hat{\lambda}_i + \hat{\mu}_i)^2$$

$$b_{p\mu_i} = -(\hat{\lambda}_j/\hat{\lambda}_j + \hat{\mu}_j) \cdot \hat{\lambda}_i/(\hat{\lambda}_i + \hat{\mu}_i)^2$$

$$c_{p\lambda_i} = -2\cdot(\hat{\lambda}_j/\hat{\lambda}_j + \hat{\mu}_j) \cdot \hat{\mu}_i/(\hat{\lambda}_i + \hat{\mu}_i)^3$$

$$c_{p\mu_i} = 2\cdot(\hat{\lambda}_j/\hat{\lambda}_j + \hat{\mu}_j) \cdot \hat{\lambda}_i/(\hat{\lambda}_i + \hat{\mu}_i)^3$$

$$b_{f\lambda_i} = (\hat{\lambda}_j/\hat{\lambda}_j + \hat{\mu}_j) \cdot \hat{\mu}_i/(\hat{\lambda}_i + \hat{\mu}_i)^2 \cdot (\hat{\mu}_i + \hat{\mu}_j) \qquad (5.65)$$

$$b_{f\mu_i} = (\hat{\lambda}_j/\hat{\lambda}_j + \hat{\mu}_j) \cdot \hat{\lambda}_i/(\hat{\lambda}_i + \hat{\mu}_i)^2 \cdot (\hat{\lambda}_i - \hat{\mu}_j)$$

$$c_{f\lambda_i} = -2\cdot(\hat{\lambda}_j/\hat{\lambda}_j + \hat{\mu}_j) \cdot \hat{\mu}_i/(\hat{\lambda}_i + \hat{\mu}_i)^3 \cdot (\hat{\mu}_i + \hat{\mu}_j)$$

$$c_{f\mu_i} = -2\cdot(\hat{\lambda}_j/\hat{\lambda}_j + \hat{\mu}_j) \cdot \hat{\lambda}_i/(\hat{\lambda}_i + \hat{\mu}_i)^3 \cdot (\hat{\lambda}_i - \hat{\mu}_j)$$

Similar expressions can be used for component j interchanging i by j in the above formulae.

Third-order cut-set

Let i, j and k be the components in the cut-set

$$b_{p\lambda_i} = (\hat{\lambda}_j/\hat{\lambda}_j + \hat{\mu}_j)\cdot(\hat{\lambda}_k/\hat{\lambda}_k + \hat{\mu}_k)\cdot\hat{\mu}_i/(\hat{\lambda}_i + \hat{\mu}_i)^2$$

$$b_{p\mu_i} = -(\hat{\lambda}_j/\hat{\lambda}_j + \hat{\mu}_j)\cdot(\hat{\lambda}_k/\hat{\lambda}_k + \hat{\mu}_k) \cdot \hat{\lambda}_i/(\hat{\lambda}_i + \hat{\mu}_i)^2$$

$$c_{p\lambda_i} = -\ 2\cdot(\hat{\lambda}_j/\hat{\lambda}_j + \hat{\mu}_j)\cdot(\hat{\lambda}_k/\hat{\lambda}_k + \hat{\mu}_k)\cdot\mu_i/(\hat{\lambda}_i + \hat{\mu}_i)^3$$

$$c_{p\mu_i} = 2\cdot(\hat{\lambda}_j/\hat{\lambda}_j + \hat{\mu}_j)\cdot(\hat{\lambda}_k/\hat{\lambda}_k + \hat{\mu}_k)\cdot\hat{\lambda}_i/(\hat{\lambda}_i + \hat{\mu}_i)^3$$

$$b_{f\lambda_i} = (\hat{\lambda}_j/\hat{\lambda}_j + \hat{\mu}_j)\cdot(\hat{\lambda}_k/\hat{\lambda}_k + \hat{\mu}_k)\cdot\hat{\mu}_i/(\hat{\lambda}_i + \hat{\mu}_i)^2\cdot(\hat{\mu}_i+\hat{\mu}_j+\hat{\mu}_k)$$

$$(5.66)$$

$$b_{f\mu_i} = (\hat{\lambda}_j/\hat{\lambda}_j + \hat{\mu}_j)\cdot(\hat{\lambda}_k/\hat{\lambda}_k + \hat{\mu}_k)\cdot\hat{\lambda}_i/(\hat{\lambda}_i + \hat{\mu}_i)^2\cdot(\hat{\lambda}_i-\hat{\mu}_j+\hat{\mu}_k)$$

$$c_{f\lambda_i} = -\ 2(\hat{\lambda}_j/\hat{\lambda}_j+\hat{\mu}_j)\cdot(\hat{\lambda}_k/\hat{\lambda}_k+\hat{\mu}_k)\cdot\hat{\mu}_i/(\hat{\lambda}_i+\hat{\mu}_i)^3\cdot(\hat{\mu}_i+\hat{\mu}_j+\hat{\mu}_k)$$

$$c_{f\mu_i} = -\ 2(\hat{\lambda}_j/\hat{\lambda}_j+\hat{\mu}_j)\cdot(\hat{\lambda}_k/\hat{\lambda}_k+\hat{\mu}_k)\cdot\hat{\lambda}_i/(\hat{\lambda}_i+\hat{\mu}_i)^3\cdot(\hat{\lambda}_i-\hat{\mu}_j-\hat{\mu}_k)$$

Similar expressions can be used for components j and k interchanging i by j or k respectively.

5.8.3 Confidence limits

The expected value E(x) and the variance Var(x) of a random variable x can be computed knowing the probability distribution of the random variable. However, the converse is not true, i.e., from a knowledge of E(x) and Var(x) it is not possible to reconstruct the probability distribution of x and hence quantities such as

$$\Pr\left\{ |x - E(x)| \leqslant c \right\}$$

cannot be calculated. Nonetheless it is possible to give very useful bounds to such probabilities using what is known as Chebyshev's inequality.

If x is a random variable with mean $E(x) = \gamma_x$ and standard deviation σ_x, then Chebyshev's theorem says

$$\Pr\left\{ |x - \gamma_x| < k\,\sigma_x \right\} \geqslant 1 - 1/k^2 \qquad (5.67)$$

This form of inequality is particularly indicative of how the variance measures the degree of concentration of probability near $E(x) = \gamma_x$. When deriving such an inequality no distributional assumption about x is made and it is still possible to place a bound on the probability that x is within k standard deviations of its mean.

Additional information about the distribution of the random variable x will enable us to improve on the inequality presented. However probability transformations lead to considerable difficulty and complication, each particular situation depends on the details of the transformation and the distributions involved. Shooman[18] presents a good summary for one or more models of probability distributions which transform in a simple and well-known manner when subject to addition,

reciprocation, multiplication and division. Unfortunately few models work well under certain of these transformations and fall down in one or two operations so that their use is not general. In many problems, like our reliability case, no assumptions concerning the specific distribution of the reliability indices is justified, and in such case Chebyshev's inequality can give us important information about the behaviour of the random indices. A minor nuisance is that this inequality is the wrong way round, one chooses the probability and then find the k value.

5.8.4 Tests

The main purpose of the tests performed was to investigate for the minimal cut-set approach to system reliability and the performance of thedeterministic and statistical methods the effect of :

(i) component availability,

(ii) variance of component parameters,

(iii) sample size of earlier testing.

The reliability network of Fig. 1 was used to test both approaches for the repairable and non-repairable cases. The components were assumed to be all equal and estimates and variances of the components parameters were available. If the accuracy of the estimates is given in terms of a definite tolerance, table II from [18] can be used to calculate an estimated value of the respective variance. Only the minimal cut-sets between nodes 1 and 4 were considered.

For the steady-state case the variances of the failure and repair rate of the components were computed using the minimum variance unbiased estimator for the exponential case(49). The minimum variance unbiased estimator of the reliability in the exponential case(54) was used to

analyse the performance of the approaches for the probability of failure for a given mission time. The variance of this parameter was calculated using the Gauss-Laguerre quadrature method as employed for the integral of Eqn. (5.9).

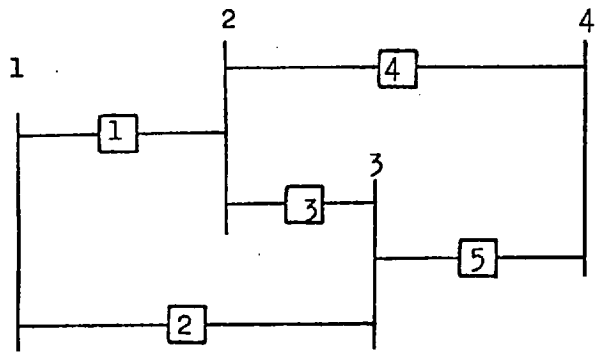


Figure 1. Sample system for tests.

TABLE II

RELATIONSHIP BETWEEN TOLERANCE AND STANDARD DEVIATION

No.	Type of component and manufacturer	Relationship
1	High-quality component in mature development stage; reputable, conservative manufacturer with much experience	$E(x) = N$ $3\sigma_x = TN$
2	Average situation somewhere between 1 and 3	$E(x) = N$ $2\sigma_x = TN$
3	Ordinary comercial component, early development stage, manufacturer little known or has poor reputation and little experience	$E(x) = N$ $\sigma_x = TN$

N = nominal value, σ_x = standard deviation, T = tolerance in per unit

The assumption of the exponential case in the steady-state was needed for the purpose to calculate the variances from the MVU estimator for this probability distribution. As Singh[63] has shown the form of the probability distribution of the up and down times do not affect the steady-state indices when the components are independent.

Test 1

The objective of this test was to evaluate the performance of the deterministic and statistical approaches for different component availabilities which were varied by selecting failure rates between 2 and 219 f/year for a repair rate of 438 repairs/year. The variances of the failure and repair rate were calculated assuming a coefficient of dispersion for both of 5%.

Tables III and IV present the results for thedeterministic approach for the probability and frequency of system failure respectively. It is apparent from these tables that the range defined by the difference between the maximum and minimum values improves as the component availability increases. It can be observed that the exact values for both indices are always inside the range but for highly unreliable components, $\lambda = 219$ f/year.

The results for the approximate statistical approach are presented in tables V to VIII. The performance of the approach compared with the single-point estimate is extremely good and has the advantage of providing information about the variance of the indices. The results of tables VII and VIII giving 95% confidence limits for both reliability indices show that the use of Chebyshev's inequality gives lower and upper limits which always enclose the best value for the high-reliability region. In the low reliability region the inequality will give only an upper limit which in the case of the probability and frequency of system failure is all what is needed.

5.8.5 Test 2

This test was performed to investigate the effect of the dispersion of component reliability data on the minimal cut-set approach to system reliability and the performance of the deterministic and statistical approaches in the steady-state case. Numerical evaluations were made for a failure rate of 4 f/year and repair rate of 438 repairs/year varying the coefficient of dispersion of both rates from 1 to 10%.

The results obtained from the deterministic approach are presented in tables IX and X for the probability and frequency of failure respectively. These tables show that the range defined by the maximum and minimum values increase for higher uncertainty in the component parameters. It seems that in case of maximum drift of these parameters from its best values, the solution of the single-point estimate of the minimal cut-set approach will be very far from the extreme points even for highly stable components.

Tables XI and XII show the results from the approximate statistical approach. These tables present the variance and coefficient of dispersion of both reliability indices for different dispersions of the component parameters. The results indicate that both variances increase as the uncertainty of the parameters increases, the incidence being higher on the probability of system failure. It is important to remark that such analysis is not possible from a single-point solution. Table XIII presents the proportional error of both reliability indices for different proportional errors of the component parameter. The results indicate that the minimal cut-set approach has very low sensitivity to errors on the data and do not amplify such errors.

Table XIV presents 95% confidence limits for different coefficients of dispersion of the component data. It can be observed that the exact

value calculated using the best value of the parameters is always inside the limits. It is also clear that the interval increases as the uncertainties of the parameters increase.

5.8.6 Test 3

The effect of the sample size of earlier testing on the performance of both approaches for the steady-state case was investigated within the high-reliability region. A failure rate of 2 f/year and a repair rate of 438 repairs/year were used for the tests.

The results from table XV show that the range provided by the deterministic approach improves as the sample size increases. The incidence of the sample size is greater in the frequency than in the probability of failure. For small samples the deterministic approach only provides maximum value.

The results in tables XVI and XVII indicate that the variance of both reliability indices is reduced as the sample size increases. These results are to be expected because the higher the sample size the smaller the variance of each component parameter. Table XVIII presents 95% confidence limits for both indices. The results show that for small samples the approach provides an upper limit which is reduced as the sample size increases. The incidence of the sample size for the lower limit is greater in the probability of system failure than in the frequency of system failure.

5.8.7 Test 4

The purpose of this test was to investigate the effect of the sample size on the best value of the reliability given by Eqn. (5.8) and its variance obtained by subtracting $\exp(-2t_m/MTTF)$ from Eqn. (5.9).

At the same time an evaluation was made of the performance of the deterministic and statistical approaches for the non-repairable case. The MTTF of the components used was 4380 h.

Tables XIX and XX show the minimum and maximum values obtained from the deterministic solution for two relations of the mission time to the component mean time to failure of 0.50 and 0.05. These two cases give the possibility to analyze the performance in both reliability regions, low and high. In both cases the "best value" obtained from the exact solution is within the range defined by the minimum and maximum values of the deterministic approach. It is clear that for a given t_m/MTTF relationship, the greater the sample size the smaller the reliability, the effect being higher for mission times greater than 50% of the MTTF. The range provided for the deterministic solution is narrowest in the high reliability region.

Tables XXI and XXII present the same results but from the statistical approach. It is also clear that increasing the mission time reduces the system reliability for a definite MTTF of the components. It is interesting to note that the dispersion of the system reliability is extremely high for the low reliability region. This result agreed with the well-known statement that the minimal cut-set approach is good in the high reliability region. It is also observed that the dispersion of the system reliability in the high reliability region is smaller than the dispersion of a single component i.e., the minimal cut-set formulae do not amplify the uncertainty in the component estimates. For both reliability regions increasing the sample size reduces the dispersion of the system reliability estimate.

Table XXIII shows that in the high reliability region the uncertainties in the components data do not affect significantly the

estimate i.e., the c-coefficients obtained in the Taylor series expansion are much smaller than the b-coefficients. It can be observed from table XXIV that the Chebyshev's inequality provides a lower limit to system reliability which for this index, system reliability, is all that is needed.

5.8.8 Test 5

The objective of the test was to investigate the effect of different component mean time to failure and its uncertainty in the system reliability and the performance of the deterministic and statistical approaches in such conditions. The tests were performed for a mission time of 100 hrs., assuming a sample size of 32.

Table XXV indicates that for different component MTTF the best value obtained from the exact solution is always within the range defined from the deterministic solution. The results of Table XXVI show that the uncertainties of component data do not affect significantly the estimate of the system reliability compared with the single-point estimate. Such estimates improve for highly reliable components, and also the variances are reduced for an increasing component mean time to failure. Table XXVII shows that the greater the component mean time to failure the smaller the dispersion of the system reliability estimate. Table XXVIII presents a 95% lower confidence limit for the system reliability which improves for an increasing mean time to failure.

5.8.9 Test 6

This test was performed to analyse the effect of component uncertainties in the system reliability for different relations between the mission time to the component mean time to failure using the results

given from the deterministic and statistical approaches. The parameters used were a component MTTF of 4380 h. and a sample size of 32.

The results of table XXIX indicate that the "best values" given by a single-point estimation from the exact solution can have more than 16% of deviation from the expected value when the variance of the component parameter estimates is taken into account for a mission time equal or greater than 50% of the component mean time to failure. The solution from the statistical approach allows us to appreciate such effects because the variance of the estimate increases under such conditions. The range defined from the deterministic approach and shown in table XXX always encloses the best value when in the high reliability region. The variance of the estimate is reduced for shorter mission times and also the dispersion of the estimate as shown in table XXXI. Table XXXII shows that Chebyshev's inequality gives important information about the system reliability for the high reliability region.

TABLE III

DETERMINISTIC SOLUTION FOR DIFFERENT COMPONENT AVAILABILITY

PROBABILITY OF FAILURE

Component repair rate = 438 repairs/year

Component dispersion coefficient = 5%

Failure rate f/year	Component availability	Single point estimate	Deterministic range	
			Minimum	Maximum
2	0.995455	0.415×10^{-4}	0.340×10^{-4}	0.507×10^{-4}
4	0.990959	0.165×10^{-3}	0.135×10^{-3}	0.202×10^{-4}
8	0.982063	0.654×10^{-3}	0.537×10^{-3}	0.799×10^{-4}
16	0.964758	0.256×10^{-2}	0.211×10^{-2}	0.313×10^{-2}
32	0.931915	0.980×10^{-2}	0.817×10^{-2}	1.20×10^{-2}
219	0.666667	0.243	0.254	0.344

TABLE IV

DETERMINISTIC SOLUTION FOR DIFFERENT COMPONENT AVAILABILITY

FREQUENCY OF FAILURE

Component repair rate = 438 repairs/year

Component dispersion coefficient = 5%

Failure rate f/year	Component availability	Single point estimate	Deterministic range	
			Minimum	Maximum
2	0.995455	0.0364	0.031	0.042
4	0.990959	0.145	0.124	0.167
8	0.982063	0.578	0.537	0.799
16	0.964758	2.28	1.95	2.63
32	0.931915	8.77	7.65	10.25
219	0.666667	202	261	323

TABLE V

APPROXIMATE STATISTICAL STEADY-STATE SOLUTION FOR

DIFFERENT COMPONENT AVAILABILITY

PROBABILITY OF FAILURE

Failure rate f/year	Component availability	Single point estimate	Statistical solution	
			Estimate	Variance
2	0.995455	0.415×10^{-4}	0.417×10^{-4}	0.846×10^{-11}
4	0.990959	0.165×10^{-3}	0.166×10^{-3}	0.132×10^{-9}
8	0.982063	0.654×10^{-3}	0.658×10^{-3}	0.200×10^{-8}
16	0.964758	0.256×10^{-2}	0.258×10^{-2}	0.288×10^{-7}
32	0.931915	0.980×10^{-2}	0.994×10^{-2}	0.376×10^{-6}
219	0.666667	0.243	0.297	0.128×10^{-3}

TABLE VI

APPROXIMATE STATISTICAL STEADY-STATE SOLUTION
FOR DIFFERENT COMPONENT AVAILABILITY

FREQUENCY OF FAILURE

Failure rate f/year	Component availability	Single point estimate	Statistical solution	
			Estimate	Variance
2	0.995455	0.0364	0.0365	0.405×10^{-5}
4	0.990959	0.145	0.146	0.630×10^{-4}
8	0.982063	0.578	0.580	0.952×10^{-3}
16	0.964758	2.28	2.30	0.136×10^{-1}
32	0.931915	8.77	8.97	0.177
219	0.666667	202	292	64.5

TABLE VII

95% CONFIDENCE LIMITS FOR THE PROBABILITY
OF SYSTEM FAILURE

Failure rate f/year	Component availability	Single point estimate	Confidence limits	
			Lower	Upper
2	0.995455	0.415×10^{-4}	0.287×10^{-4}	0.547×10^{-4}
4	0.990959	0.165×10^{-3}	0.115×10^{-3}	0.217×10^{-3}
8	0.982063	0.654×10^{-3}	0.458×10^{-3}	0.858×10^{-3}
16	0.964758	0.256×10^{-2}	0.182×10^{-2}	0.334×10^{-2}
32	0.931915	0.980×10^{-2}	0.720×10^{-2}	0.127×10^{-1}
219	0.666667	0.243	--	0.347

TABLE VIII

95% CONFIDENCE LIMITS FOR THE FREQUENCY OF SYSTEM FAILURES

Failure rate f/year	Component availability	Single point estimate	Confidence limits	
			Lower	Upper
2	0.995455	0.0364	0.0275	0.0455
4	0.990959	0.145	0.1103	0.1813
8	0.982063	0.578	0.4422	0.7182
16	0.964758	2.28	1.774	2.818
32	0.931915	8.77	7.086	10.85
219	0.666667	202	-	327.8

TABLE IX

DETERMINISTIC STEADY-STATE SOLUTION FOR DIFFERENT DISPERSION COEFFICIENTS OF COMPONENT PARAMETERS

Component parameter % coefficient of dispersion	Failures/year	
	Minimum	Maximum
1	0.159×10^{-3}	0.172×10^{-3}
2	0.153×10^{-3}	0.179×10^{-3}
3	0.146×10^{-3}	0.186×10^{-3}
4	0.141×10^{-3}	0.194×10^{-3}
5	0.135×10^{-3}	0.202×10^{-3}
6	0.130×10^{-3}	0.210×10^{-3}
7	0.125×10^{-3}	0.218×10^{-3}
8	0.120×10^{-3}	0.227×10^{-3}
9	0.115×10^{-3}	0.237×10^{-3}
10	0.111×10^{-3}	0.246×10^{-3}

TABLE X

DETERMINISTIC STEADY-STATE SOLUTION FOR DIFFERENT DISPERSION COEFFICIENTS OF COMPONENT PARAMETERS-FREQUENCY OF SYSTEM FAILURE

Component parameter % coefficient of dispersion	Encounters/year	
	Minimum	Maximum
1	0.141	0.149
2	0.137	0.154
3	0.132	0.158
4	0.128	0.163
5	0.124	0.167
6	0.119	0.172
7	0.115	0.176
8	0.111	0.180
9	0.106	0.185
10	0.102	0.189

TABLE XI

APPROXIMATE STATISTICAL STEADY-STATE SOLUTION FOR DIFFERENT DISPERSION COEFFICIENT OF COMPONENT PARAMETERS PROBABILITY OF SYSTEM FAILURE

Component failure rate = 4 f/year Component repair rate = 438

Parameter % coefficient of dispersion	Estimate	Variance	System % coefficient of dispersion
1	0.1653×10^{-3}	0.527×10^{-11}	1.39
2	0.1654×10^{-3}	0.211×10^{-10}	2.78
3	0.1656×10^{-3}	0.474×10^{-10}	4.16
4	0.1658×10^{-3}	0.843×10^{-10}	5.54
5	0.1660×10^{-3}	0.132×10^{-9}	6.91
6	0.1664×10^{-3}	0.189×10^{-9}	8.28
7	0.1669×10^{-3}	0.258×10^{-9}	9.63
8	0.1674×10^{-3}	0.337×10^{-9}	10.97
9	0.1679×10^{-3}	0.427×10^{-9}	12.31
10	0.1685×10^{-3}	0.527×10^{-9}	13.62

TABLE XII

APPROXIMATE STATISTICAL STEADY-STATE SOLUTION FOR DIFFERENT DISPERSION COEFFICIENT OF COMPONENT PARAMETERS

FREQUENCY OF SYSTEM FAILURE

Component failure rate = 4 f/year
Component repair rate = 438 repairs/year

Parameter % coefficient of dispersion	Estimate	Variance	System % coefficient of dispersion
1	0.1454	0.25×10^{-5}	1.09
2	0.1455	0.10×10^{-4}	2.18
3	0.1456	0.23×10^{-4}	3.27
4	0.1457	0.40×10^{-4}	4.36
5	0.1458	0.63×10^{-4}	5.44
6	0.1459	0.91×10^{-4}	6.52
7	0.1461	0.12×10^{-3}	7.60
8	0.1463	0.16×10^{-3}	8.68
9	0.1466	0.20×10^{-3}	9.75
10	0.1468	0.25×10^{-3}	10.81

TABLE XIII

PROPORTIONAL ERROR IN THE PROBABILITY AND FREQUENCY OF SYSTEM FAILURE FOR DIFFERENT PROPORTIONAL ERROR IN THE COMPONENT PARAMETERS

Component % proportional error	Probability of failure % proportional error	Frequency of failure % proportional error
1	0.18	0.28
2	0.24	0.34
3	0.36	0.41
4	0.48	0.48
5	0.60	0.55
6	0.84	0.62
7	1.14	0.75
8	1.43	0.89
9	1.73	1.09
10	2.08	1.23

TABLE XIV

95% CONFIDENCE LIMITS

PROBABILITY AND FREQUENCY OF SYSTEM FAILURE

Parameter % coefficient of dispersion	Probability of system failure		Frequency of failure	
	Lower	Upper	Lower	Upper
1	0.155×10^{-3}	0.176×10^{-3}	0.138	0.153
2	0.145×10^{-3}	0.185×10^{-3}	0.131	0.160
3	0.135×10^{-3}	0.196×10^{-3}	0.124	0.167
4	0.125×10^{-3}	0.207×10^{-3}	0.117	0.174
5	0.115×10^{-3}	0.217×10^{-3}	0.110	0.181
6	0.105×10^{-3}	0.228×10^{-3}	0.103	0.189
7	0.950×10^{-4}	0.239×10^{-3}	0.096	0.196
8	0.852×10^{-4}	0.250×10^{-3}	0.090	0.203
9	0.755×10^{-4}	0.260×10^{-3}	0.083	0.210
10	0.659×10^{-4}	0.271×10^{-3}	0.076	0.218

TABLE XV

DETERMINISTIC STEADY-STATE SOLUTION FOR DIFFERENT SAMPLE SIZES

Component failure rate = 2 f/year
Component repair rate = 438 repairs/year

Node 4 in Fig. 1

Sample size	Probability of failure		Frequency of failure	
	Minimum	Maximum	Minimum	Maximum
3	0.12×10^{-5}	0.14×10^{-2}	-	0.114
5	0.46×10^{-5}	0.37×10^{-3}	-	0.091
9	0.95×10^{-5}	0.18×10^{-3}	-	0.075
17	0.15×10^{-4}	0.11×10^{-3}	0.009	0.063
33	0.20×10^{-4}	0.85×10^{-4}	0.017	0.056
65	0.25×10^{-4}	0.68×10^{-4}	0.023	0.050
129	0.29×10^{-4}	0.59×10^{-4}	0.027	0.046
257	0.32×10^{-4}	0.53×10^{-4}	0.030	0.043

TABLE XVI

STATISTICAL STEADY-STATE SOLUTION FOR DIFFERENT SAMPLE SIZES

Component failure rate = 2 f/year
Component repair rate = 438 repairs/year

Sample size	Parameters % coefficient of dispersion	Probability of failure		
		Estimate	Variance	% dispersion
3	70.7	0.825×10^{-4}	0.169×10^{-8}	49.8
5	50.0	0.620×10^{-4}	0.847×10^{-9}	46.9
9	35.3	0.518×10^{-4}	0.423×10^{-9}	39.7
17	25.0	0.466×10^{-4}	0.212×10^{-9}	31.2
33	17.7	0.441×10^{-4}	0.106×10^{-9}	23.3
65	12.5	0.428×10^{-4}	0.529×10^{-10}	16.7
129	8.8	0.422×10^{-4}	0.264×10^{-10}	12.2

TABLE XVII

STATISTICAL STEADY-STATE SOLUTION FOR DIFFERENT SAMPLE SIZES

FREQUENCY OF FAILURE

Component failure rate = 2 f/year
Component repair rate = 438 repairs/year

Sample size	Parameters % coefficient of dispersion	Frequency of failures		
		Estimate	Variance	% dispersion
3	70.7	0.0544	0.810×10^{-3}	52.3
5	50.0	0.0454	0.405×10^{-3}	44.3
9	35.3	0.0409	0.203×10^{-3}	34.8
17	25.0	0.0387	0.101×10^{-3}	26.0
33	17.7	0.0376	0.506×10^{-4}	18.9
65	12.5	0.0370	0.253×10^{-4}	13.6
129	8.8	0.0367	0.127×10^{-4}	9.7

TABLE XVIII

95% CONFIDENCE LIMITS FOR DIFFERENT SAMPLE SIZES

Component failure rate = 2 f/year
Component repair rate = 438 repairs/year

Sample size	Probability of failure		Frequency of failure	
	Lower	Upper	Lower	Upper
3	-	0.267×10^{-3}	-	0.182
5	-	0.192×10^{-3}	-	0.135
9	-	0.144×10^{-3}	-	0.105
17	-	0.112×10^{-3}	-	0.0837
33	-	0.901×10^{-4}	0.0057	0.0694
65	0.103×10^{-4}	0.753×10^{-4}	0.0145	0.0595
129	0.192×10^{-4}	0.651×10^{-4}	0.0208	0.0526

TABLE XIX

DETERMINISTIC SOLUTION FOR THE NON-REPAIRABLE SYSTEM

Component MTTF = 4380.0 hrs.
Mission time = 0.50 MTTF

Sample size	System reliability		
	Best value	Minimum	Maximum
2	0.8613	0.161	1.0
4	0.7617	0.238	0.943
8	0.7143	0.323	0.845
16	0.6915	0.390	0.765
32	0.6804	0.440	0.707

TABLE XX

DETERMINISTIC SOLUTION FOR THE NON-REPAIRABLE SYSTEM

Component MTTF = 4380.0
Mission time = 0.05 MTTF

Sample size	System Reliability		
	Best value	Minimum	Maximum
2	0.9987	0.9800	1.0
4	0.9972	0.9897	0.99996
8	0.9962	0.9918	0.9989
16	0.9956	0.9928	0.9977
32	0.9953	0.9934	0.9969

TABLE XXI

STATISTICAL SOLUTION FOR THE NON-REPAIRABLE SYSTEM

Component MTTF = 4380.0 hrs
Mission time = 0.50 MTTF

Sample size	Component Reliability			System Reliability		
	Estimate	Variance	% dispersion	Estimate	Variance	% dispersion
2	0.750	0.0753	109.8	0.84375	0.10358	38.1
4	0.670	0.0300	52.5	0.71017	0.5925×10^{-1}	34.3
8	0.637	0.0132	31.6	0.63968	0.2966×10^{-1}	26.9
16	0.621	0.0062	20.7	0.60412	0.1465×10^{-1}	20.0
32	0.614	0.0030	14.1	0.58633	0.7264×10^{-1}	14.5

TABLE XXII

STATISTICAL SOLUTION FOR THE NON-REPAIRABLE SYSTEM

Component MTTF = 4380.0 hrs.
Mission time = 0.05 MTTF

Sample size	Component Reliability			System Reliability		
	Estimate	Variance	% dispersion	Estimate	Variance	% dispersion
2	0.9750000	49.7×10^{-4}	7.23	0.99872	5.16×10^{-4}	2.27
4	0.9629668	10.4×10^{-4}	3.36	0.99716	1.63×10^{-4}	1.28
8	0.9570618	3.70×10^{-4}	2.01	0.99615	6.76×10^{-5}	0.83
16	0.9541366	1.60×10^{-4}	1.33	0.99560	3.14×10^{-5}	0.56
32	0.9526808	7.51×10^{-5}	0.91	0.99530	1.50×10^{-5}	0.39

TABLE XXIII

COMPARISON OF THE BEST VALUE AND THE ESTIMATE FROM THE STATISTICAL APPROACH FOR DIFFERENT SAMPLE SIZES

Component MTTF = 4380.0 hrs.
Mission time = 0.05 MTTF

Sample size	System Reliability	
	Best value	Estimate
2	0.99872	0.99872
4	0.99716	0.99716
8	0.99617	0.99615
16	0.99562	0.99560
32	0.99533	0.99530

TABLE XXIV

95% LOWER CONFIDENCE LIMIT FOR DIFFERENT SAMPLE SIZES

Component MTTF = 4380.0 hrs.
Mission time = 0.05 MTTF

Sample size	Best value	Lower confidence limit
2	0.99872	0.8972
4	0.99716	0.9400
8	0.99617	0.9594
16	0.99562	0.9705
32	0.99533	0.9779

TABLE XXV

DETERMINISTIC SOLUTION FOR DIFFERENT MEAN TIME TO FAILURE OF COMPONENTS

Component MTTF = variable
Mission time = 100 hrs.
Sample size = 32

Component MTTF	Best value	Deterministic range	
		Minimum	Maximum
438	0.9122	0.8674	0.9356
876	0.9762	0.9658	0.9838
1095	0.9847	0.9780	0.9896
2190	0.9961	0.9945	0.9974
4380	0.9990	0.9986	0.9994
17520	0.99994	0.99991	0.99996

TABLE XXVI

STATISTICAL SOLUTION FOR DIFFERENT MEAN TIME TO FAILURE OF NON-REPAIRABLE COMPONENTS

Sample size = 32
Mission time = 100 hrs.

Component MTTF	System Reliability		
	Best value	Estimate	Variance
438	0.9122	0.90498	11.2×10^{-4}
876	0.9762	0.97570	1.67×10^{-4}
1095	0.9847	0.97570	8.80×10^{-5}
2190	0.9961	0.99609	1.17×10^{-5}
4380	0.9990	0.99902	1.50×10^{-6}
17520	0.99994	0.99994	2.39×10^{-8}

TABLE XXVII

COMPARISON OF THE DISPERSION COEFFICIENTS OF THE COMPONENT AND SYSTEM RELIABILITIES FOR DIFFERENT MEAN TIME TO FAILURE OF NON-REPAIRABLE COMPONENTS

Sample size = 32
Mission time = 100 hrs.

Component MTTF	Component % coefficient of dispersion	System % coefficient of dispersion
438	3.66	3.67
876	1.86	1.30
1095	1.46	0.95
2190	0.83	0.34
4380	0.42	0.12
17520	0.11	0.015

TABLE XXVIII

95% CONFIDENCE LIMIT FOR DIFFERENT MEAN TIME TO FAILURE OF NON-REPAIRABLE COMPONENTS

Mission time = 100 hrs.
Sample size = 32

Component MTTF	Best value	Lower limit
438	0.9122	0.75521
876	0.9762	0.91798
1095	0.9847	0.94245
2190	0.9961	0.98081
4380	0.9990	0.99355
17520	0.99994	0.99925

TABLE XXIX

EFFECTS OF COMPONENT DATA UNCERTAINTY IN THE SYSTEM RELIABILITY FOR DIFFERENT RELATION OF MISSION TIME TO MEAN TIME TO FAILURE OF NON-REPAIRABLE COMPONENTS

Component MTTF = 4380.0 hrs.
Sample size = 32

t_m/MTTF	Best value	Estimate	Variance
0.01	0.99981	0.99981	3.30×10^{-6}
0.05	0.99533	0.99531	7.50×10^{-5}
0.10	0.98167	0.98132	1.14×10^{-4}
0.20	0.93144	0.92658	7.89×10^{-4}
0.30	0.85775	0.83927	2.26×10^{-3}
0.40	0.77147	0.72413	4.48×10^{-3}
0.50	0.68043	0.58633	7.26×10^{-3}
0.60	0.59067	0.43103	1.03×10^{-2}
0.70	0.50611	0.2631	1.34×10^{-2}

TABLE XXX

DETERMINISTIC SOLUTION FOR DIFFERENT RELATION OF MISSION TIME TO MEAN TIME TO FAILURE OF NON-REPAIRABLE COMPONENTS

Component mean time to failure = 4380.0 hrs.
Sample size = 32

t_m/MTTF	Best value	Minimum	Maximum
0.01	0.99981	0.99973	0.99988
0.05	0.99533	0.99338	0.99690
0.10	0.98617	0.97368	0.98757
0.20	0.93144	0.89728	0.95048
0.30	0.85775	0.77728	0.88990
0.40	0.77147	0.62197	0.80787
0.50	0.68043	0.44001	0.70694
0.60	0.59067	0.23952	-
0.70	0.50611	0.27765	-

TABLE XXXI

DISPERSION COEFFICIENTS OF COMPONENT AND SYSTEM RELIABILITY OF DIFFERENT RELATIONS OF MISSION TIME AND MEAN TIME TO FAILURE

Component MTTF = 4380.0 hrs
Sample size = 32

t_m/MTTF	Component % coefficient of dispersion	System % coefficient of dispersion
0.01	0.18	0.036
0.05	0.91	0.39
0.10	1.57	1.06
0.20	3.60	3.03
0.30	5.38	5.66
0.40	7.14	9.25
0.50	8.89	14.54
0.60	10.63	23.58
0.70	12.36	44.03

TABLE XXXII

95% CONFIDENCE LIMITS OF SYSTEM RELIABILITY FOR DIFFERENT RELATIONS OF MISSION TIME TO MEAN TIME TO FAILURE OF NON-REPAIRABLE COMPONENTS

Component MTTF = 4380.0 hrs.
Sample size = 32

t_m/MTTF	Best value	Lower limit	Upper limit
0.01	0.99981	0.99821	1.0
0.05	0.99533	0.97786	1.0
0.10	0.98167	0.93353	1.0
0.20	0.93144	0.80100	1.0
0.30	0.85775	0.62676	1.0
0.40	0.77147	0.42466	1.0
0.50	0.68043	0.20516	1.0
0.60	0.59067	0.0	0.88559
0.70	0.50611	0.0	0.78116
1.00	0.30079	0.0	0.37026

5.9 Approximate confidence limits using Monte Carlo simulation

In the analytical method, the technique of combination of distribution functions is used to derive the sampling distribution of the system reliability indices from the sampling distributions of the component reliability estimates; confidence limits are then obtained from this sampling distribution.

For the Monte Carlo method, the numerical sampling distributions of component reliability estimates are generated. Sample values selected from these distributions are substituted into the mathematical relationship between system and components. This simulation is repeated until the numerical sampling distribution of the system reliability indices are generated. Confidence limits are then computed on this numerical distribution.

5.9.1 Non-repairable systems

The approach described here is similar to that of [60], except that in this case the system is represented by its minimal cut-sets and [60] use the series and parallel reliability formulae to obtain the system reliability.

The probability of failure Q at time t is usually expressed as

$$Q(t) = \int_0^t f(t;\theta)\ dt$$

where $f(t;\theta)$ is the failure density model with parameter θ. If the parameter of the model is unknown and must be estimated from component test data, then the probability of failure must be expressed as an estimate $\hat{Q}(t)$. Such a model parameter has been extensively estimated by the method of maximum likelihood. The results of such estimations summarized in [60] are presented here for the exponential case, where t_{in} are the i-th order statistics from a sample of size n.

If the exponential density is written as

$$f(t;\theta) = 1/\theta \exp(-t/\theta) \qquad \theta > 0,\ t \geqslant 0$$
$$= 0 \qquad \text{elsewhere}$$

Then the maximum likelihood estimator $\hat{\theta}$, for the mean time between failures θ, from a complete sample or censored after r failures is given by

$$\hat{\theta}_{r,n} = \left\{ \sum_{i=1}^{r} t_{in} + (n-r)t_{rn} \right\}/r \qquad (5.68)$$

where the numerator of Eqn. (5.68) is the total accumulated operating time T of the components under test. Further $\hat{\theta}_{r,n}$ is unbiased and $2r\hat{\theta}_{r,n}/\theta$ is distributed as chi-square with 2r degrees of freedom[46]. For a two-sided confidence level $(1 - \alpha)$ we can write confidence limits

making use of the $\alpha/2$ and $1-\alpha/2$ percentage points of the chi-square distribution

$$\frac{2\,r\,\hat{\theta}}{\chi^2_{\alpha/2,2r}} \leqslant \theta \leqslant \frac{2\,r\,\hat{\theta}}{\chi^2_{1-\alpha/2,2r}} \tag{5.69}$$

Since the parameter θ of the exponential failure model is unknown and must be estimated from component test data, then the probability of failure must be expressed as an estimate $\hat{Q}(t)$. By the invariant property of ML estimator of parameters[64], if a random value is taken from the distribution of the estimator for the parameter θ and this is substituted into the expression for $Q(t_m)$ in the exponential case, one obtains a sample probability of failure given by

$$\hat{Q}(t_m) = 1 - \exp(-\,t_m\,\chi^2_{2r} / 2\,r\,\hat{\theta}) \tag{5.70}$$

The Monte Carlo computer technique for the reliability of a non-repairable system represented by its minimal cut-sets involves use of a digital computer to obtain a large number of sample probability of failure values for each component from which one obtains a numerical sampling distribution for the system reliability $R_s(t_m)$ through the mathematical expression

$$R_s(t_m) = 1 - \sum_{i=1}^{NMC} Pr\{\bar{C}_i\} = 1 - \sum_{i=1}^{NMC} \prod_{j\varepsilon i} Q_j(t_m) \tag{5.71}$$

If m(Monte Carlo sample size) values are obtained for $R_s(t_m)$ by the previous synthetic sampling and arranged in order of increasing magnitude, each value will represent a 1/m increase in the step cumulative distribution function. If m is taken as large as, say 1000 then the 99.9% confidence limits for $Q_s(t_m)$ will be the largest and smallest

values of the ordered numerical distribution. In general, the two-sided lower confidence limit is

$$R_{SL}(t_m) = R_S(i) \quad i = m \cdot \alpha/2 \tag{5.72}$$

and the upper confidence limit is

$$R_{SU}(t_m) = R_S(j) \quad j = m \cdot (1-\alpha/2) \tag{5.73}$$

where m is the Monte Carlo sample size and α is the level of significance. When it is required that the system reliability does not go below a minimum value we use the procedure of one-sided confidence limit and Eqn. (5.72) corresponds to a lower one-sided confidence limit at $100(1-\alpha/2)$ confidence level.

5.9.2 Algorithm and an illustrative system

The main steps for the Monte Carlo simulation procedure can be summarised as follows:

Step 1 : Generate the minimal cut-set matrix of the system.

Step 2 : Generate component sample probability of failure values for the given mission time and sample size, using a random number generator from a chi-square distribution. Component sample values must not be ordered.

Step 3 : Simulate minimal cut-set sample values and sample system reliability values.

Step 4 : Sort the sample system reliability values in increasing order of magnitude.

Step 5 : From the ordered values of system reliability obtain system reliability limits at any specified confidence level.

To illustrate the Monte Carlo simulation technique, we shall use the reliability network shown in Fig. 1. The problem is of course a hypothetical one, but it allowed us to compare the method of moments approach with the exact analytical result and now allows us to compare the approximate lower confidence limit from the Chebyshev's inequality with the present Monte Carlo solution. Components one through five were assumed to be equals, to follow the exponential failure density and tested until all components failed.

Using the previous Monte Carlo technique, one thousand samples of $Q(t_m)$ from Eqn. (5.70) were obtained for the components. These component-samples combined through the minimal cut-set matrix generated one thousand sample values for each minimal cut-set which were used to simulate the sample reliability values using Eqn. (5.71). The system reliability values were ordered to construct an approximate step distribution. Table XXXIII presents the results for 1000 experiments with a component mean-time-to-failure of 4380 hours, a relation of mission time to MTTF of 0.05 and a sample size of 32. Column 1 is the confidence level that the system reliability will fall between the given upper and lower limits of columns 2 and 3, respectively. Column 4 represents the one-sided confidence level that the system reliability is less than or equal to the values of column 2 or equal to or greater than the values of column 3.

The program was also run for 100, 500 and 1000 simulations with the same data as before except that a sample size of 16 was used. Comparison of the 1000-experiments of Tables XXXIII and XXXIV indicate that the higher the sample size the narrower the interval at any confidence level. This effect is obtained for any number of simulations used as shown in Table XXXV. This table also shows that the same effect is obtained when for the same sample size the number of trials is increased.

Table XXXVI indicates that the interval given by the Monte Carlo approach improves in the high reliability region. This effect is even more clear from Table XXXVII where the limits were obtained for different relations of the mission time to the component mean-time-to-failure.

5.9.3 Comparison with the moment approach

Table XXXVIII shows that one-hundred cycles of the simulation process (Monte Carlo method) for the sample system of Fig. 1 resulted in data samples for the system reliability having mean and variance very close to those computed using the equations of the moment approach. Hence, 100 cycles of the simulation process was deemed adequate.

Table XXXIX shows that the lower limit provided by the Chebyshev's inequality is always more pessimistic than the one from the Monte Carlo technique and obviously the former improves for more stable components, i.e., components with smaller variance. One of the major drawbacks of the Monte Carlo method is the large number of trials required to give a representative numerical distribution, and the time of the present computer approach is increased as the sample size and the number of trials is increased. Table XXXIX presents the computer times (seconds of CPU) for a selected number of sample sizes between 2 and 32 for 100, 500 and 1000 experiments. It must be emphasized that the computer time of the Chebyshev's approach is unaffected by the component sample size and is a one-experiment solution. The average computer time for the Chebyshev's inequality approach is 2.4 seconds of CPU in a CDC-6400 digital computer.

TABLE XXXIII

CONFIDENCE LIMITS FROM THE MONTE CARLO TECHNIQUE FOR NON-REPAIRABLE SYSTEMS

MTTF = 4380 hrs. Sample size = 32 t_m/MTTF = 0.05
Monte Carlo size = 1000 experiments

Two-sided confidence level 1	Upper limit 2	Lower limit 3	One-sided confidence level 4
99.0	0.996895	0.992634	99.5
98.0	0.996822	0.992818	99.0
97.0	0.996711	0.992902	98.5
96.0	0.996521	0.993022	98.0
95.0	0.996472	0.993165	97.5
90.0	0.996283	0.993558	95.0
80.0	0.996048	0.993938	90.0
70.0	0.995869	0.994129	85.0
60.0	0.995750	0.994294	80.0
50.0	0.995643	0.994486	75.0

TABLE XXXIV

CONFIDENCE LIMITS FROM THE MONTE CARLO TECHNIQUE FOR NON-REPAIRABLE SYSTEMS WITH DIFFERENT NUMBER OF TRIALS

MTTF = 4380 hrs. Sample size = 16 t_m/MTTF = 0.05

Two-sided confidence level	100-experiments		500-experiments		1000-experiments	
	Upper limit	Lower limit	Upper limit	Lower limit	Upper limit	Lower limit
99.0	0.997479	0.991266	0.997325	0.991118	0.997600	0.991115
98.0	0.997474	0.991274	0.997216	0.991349	0.997399	0.991406
97.0	0.997161	0.992248	0.997197	0.991847	0.997269	0.991976
96.0	0.997106	0.992488	0.997144	0.992111	0.997118	0.992109
95.0	0.996947	0.992710	0.997032	0.992312	0.997094	0.992339
90.0	0.996457	0.993058	0.996817	0.992684	0.996826	0.992941
80.0	0.996232	0.994099	0.996543	0.993462	0.996514	0.993486
70.0	0.995891	0.994674	0.996363	0.993880	0.996294	0.993818
60.0	0.995599	0.994924	0.996199	0.994182	0.996082	0.994072
50.0	0.995308	0.995304	0.996073	0.994370	0.995903	0.994274

TABLE XXXV

90% CONFIDENCE LIMITS FOR DIFFERENT SAMPLE SIZES

MTTF = 4380 Hrs. t_m/MTTF = 0.05

Sample size	100-experiments		500-experiments		1000-experiments	
	Upper limit	Lower limit	Upper limit	Lower limit	Upper limit	Lower limit
2	0.999337	0.987198	0.999120	0.988493	0.999082	0.988200
4	0.998375	0.989928	0.998179	0.991117	0.998299	0.990204
8	0.997548	0.991739	0.997584	0.992072	0.997431	0.991914
16	0.996631	0.992710	0.996817	0.992684	0.996826	0.992941
32	0.996602	0.993456	0.996339	0.993550	0.996283	0.993558

TABLE XXXVI

90% CONFIDENCE LIMITS FOR DIFFERENT MEAN TIME TO FAILURE

1000-experiments Sample size = 32 t_m = 100 hrs

Component MTTF	1000-experiments	
	Upper limit	Lower limit
438	0.924528	0.874458
876	0.980748	0.967168
1095	0.987608	0.978845
2190	0.996902	0.994619
4380	0.999228	0.998650

TABLE XXXVII

90% CONFIDENCE LIMITS FOR DIFFERENT RELATION OF MISSION TIME TO COMPONENT MEAN TIME TO FAILURE

1000-experiments Sample size = 32 MTTF = 4380 hrs.

t_m/MTTF	Upper limit	Lower limit
0.01	0.999852	0.999741
0.05	0.996283	0.993558
0.10	0.985170	0.974729
0.20	0.941678	0.902517
0.30	0.872867	0.790141
0.40	0.780800	0.644682
0.50	0.670542	0.474584

TABLE XXXVIII

ESTIMATE AND VARIANCE OBTAINED ANALYTICALLY AND BY MONTE CARLO SIMULATION FOR DIFFERENT NUMBERS OF TRIALS

Component MTTF = 4380 hrs. t_m/MTTF = 0.5

Sample size = 32

Analytical solution

Estimate = 0.58633
Variance = 0.72643×10^{-2}

Monte Carlo size	Estimate	% diff.	Variance	% diff.
20	0.56197	4.3	0.57660×10^{-2}	26.0
30	0.56734	3.3	0.59543×10^{-2}	22.0
50	0.57673	1.7	0.60500×10^{-2}	20.1
100	0.58653	0.03	0.71288×10^{-2}	1.9

TABLE XXXIX

COMPARISON OF 90% LOWER CONFIDENCE LIMIT FOR

DIFFERENT SAMPLE SIZES

Monte Carlo size = 100 experiments
MTTF = 4380 hrs. t_m/MTTF = 0.05

Sample size	Chebyshev's inequality	Monte Carlo technique	% diff.
2	0.92690	0.98720	6.1
4	0.95675	0.98993	3.4
8	0.97015	0.99174	2.2
16	0.97787	0.99271	1.5
32	0.99945	0.99346	0.6

TABLE XL

COMPUTER TIME (SECONDS OF CPU) FOR DIFFERENT
NUMBER OF TRIALS AND SAMPLE SIZES

Sample size	Number of experiments		
	100	500	1000
2	1.50	2.20	4.50
4	1.60	2.30	4.90
8	1.80	3.50	5.60
16	2.10	4.90	8.00
32	2.30	6.30	12.10

5.9.4 Repairable systems

This section presents a new Monte Carlo technique for the probability and frequency of system failure which combines minimal cut-set representation and Monte Carlo simulation. The individual components have been assumed to be statistically independent of each other, and repair of failed components is allowed. The method is a synthetic one, in that the steady-state probability that components are failed and the rate of departure from the failed state for each components are generated and then used to determine numerical distributions for the probability and frequency of system failure.

The probability that component i is failed in a steady state situation is given by

$$P_{d_i} = \frac{\lambda_i}{\lambda_i + \mu_i}$$

where λ_i is the component failure rate

μ_i is the component repair rate.

The one-to-one correspondence between P_{d_i} and the ratio of its repair and failure rate μ_i/λ_i is obvious and will be used to define F-distributed variables from which P_d-sample values can be generated.

The usual sample estimate of P_{d_i} is

$$\hat{P}_{d_i} = \frac{1}{1 + \hat{\mu}_i/\hat{\lambda}_i} \tag{5.74}$$

Let us assume that t_1 (time-to-failure) and t_2 (time-to-repair) are statistically independent random variables with exponential probability density functions

$$f_1(t_1) = e^{-\lambda t_1} \tag{5.75}$$

$$f_2(t_2) = e^{-\mu t_2} \tag{5.76}$$

then $\hat{\lambda}$, the sample estimate of λ, is obtained from

$$\hat{\lambda} = \frac{n_1}{\sum_{i=1}^{n_1} t_{1_i}} \tag{5.77}$$

where t_{1_i} = time between the (i-1)st and the i-th failures

n_1 = number of failures

and $\hat{\mu}$, the sample estimate of μ, is obtained from

$$\mu = \frac{n_2}{\sum_{j=1}^{n_2} t_{2_j}} \tag{5.78}$$

where t_{2_j} = time-to-repair associated with the j-th failure

n_2 = number of repair actions initiated.

Consider a random sample of n_1 times-to-failure and n_2 times-to-repair drawn from the populations described by Eqns. (5.75) and (5.76) with sample means $\hat{\lambda}$ and $\hat{\mu}$ calculated from Eqns. (5.77) and (5.78) respectively. It is well known that $2n_1\lambda/\hat{\lambda}$ and $2n_2\mu/\hat{\mu}$ are chi-square distributed variables with $2n_1$ and $2n_2$ degrees of freedom respectively. Since they are independent due to the independence of the random variables t_1 and t_2, it is possible to define a new variable

$$z_1 = \frac{\dfrac{2\,n_2\mu/\hat{\mu}}{2\,n_2}}{\dfrac{2\,n_1\,\lambda/\hat{\lambda}}{2\,n_1}} = \frac{\hat{\lambda}\,\mu}{\hat{\mu}\,\lambda} \tag{5.79}$$

which is F-distributed with $2n_2$, $2n_1$ degrees of freedom. Similar sample estimators for t_1 and t_2 of those given by Eqns. (5.77) and (5.78) can be formulated for the gamma and Weibull probability density function[60]. Therefore a similar z-variable can be defined also for such distributions.

The variable z_1 can be used to obtain a lower confidence limit for P_{d_i} as follows:

$$\Pr\left\{ \frac{\mu\,\hat{\lambda}}{\lambda\,\hat{\mu}} \leqslant F_{1-\alpha}\ (2n_2,2n_1) \right\} = 1 - \alpha$$

$$\Pr\left\{ \frac{\mu}{\lambda} \leqslant \frac{\hat{\mu}}{\hat{\lambda}}\ F_{1-\alpha}\ (2n_2,2n_1) \right\} = 1 - \alpha$$

$$\Pr\left\{ 1 + \frac{\mu}{\lambda} \leqslant 1 + \frac{\hat{\mu}}{\hat{\lambda}}\ F_{1-\alpha}\ (2n_2,2n_1) \right\} = 1 - \alpha$$

$$\Pr\left\{ \frac{1}{1 + \frac{\hat{\mu}}{\hat{\lambda}}\ F_{1-\alpha}\ (2n_2,2n_1)} \leqslant \frac{1}{1 + \mu/\lambda} \right\} = 1 - \alpha \qquad (5.80)$$

In most practical cases $n_1=n_2$ and Eqn. (5.80) becomes

$$\Pr\left\{ \frac{1}{1 + \frac{\hat{\mu}}{\hat{\lambda}}\ F_{1-\alpha}\ (2n,2n)} \leqslant P_d \right\} = 1 - \alpha \qquad (5.81)$$

and the $(1 - \alpha)$ lower confidence limit is found from

$$P_{d_L} = \frac{\hat{\lambda}}{\hat{\lambda} + \hat{\mu}\ F_{1-\alpha}(2n,2n)} \qquad (5.82)$$

Equation (5.83) is used to generate sample steady-state probability of failure values for each component

$$P_{d_i} = \frac{\hat{\lambda}_i}{\hat{\lambda}_i + \hat{\mu}_i\ F_{2n_i,2n_i}} \qquad (5.83)$$

where $F_{2n_i,2n_i}$ represents random values drawn from an F-distribution with $(2n_i,2n_i)$ degrees of freedom.

As mentioned before $2n_2\ \mu/\hat{\mu}$ is chi-square distributed with $2n_2$ degrees of freedom. Hence, a $(1 - \alpha)$ lower confidence limit for μ can be obtained as follows :

$$\Pr\left\{ \frac{2n_2\mu}{\hat{\mu}} \leqslant \chi^2_{1-\alpha,2n_2} \right\} = 1 - \alpha$$

$$\Pr\left\{ \mu \leqslant \frac{\hat{\mu}}{2n_2}\ \chi^2_{1-\alpha,2n_2} \right\} = 1 - \alpha$$

and the $(1 - \alpha)$ lower confidence limit for μ with $n_2 = n$ is found from

$$\mu_L = \frac{\hat{\mu}}{2n}\ \chi^2_{1-\alpha,2n} \tag{5.84}$$

Equation (5.85) is used to generate sample values of the rate of departure from the failed state for each component, thus

$$\mu_i = \frac{\hat{\mu}_i}{2n_i}\ \chi^2_{2n_i} \tag{5.85}$$

where $\chi^2_{2n_i}$ represents random values drawn from a chi-square distribution with $2n_i$ degrees of freedom.

The Monte Carlo technique involves use of a digital computer to obtain a large number of component sample values of the steady-state probability of being in the down state P_{d_i} and for the rate of departure from the failed state μ_i. These values are combined through

Eqns. (5.86) and (5.87) to obtain sample values for P_f and f_f respectively.

$$P_f = \sum_i \prod_{j \varepsilon i} P_{d_j} \quad ; \quad i = 1,NMC \tag{5.86}$$

$$f_f = \sum_i \prod_{j \varepsilon i} P_{d_j} \cdot \sum_{j \varepsilon i} \mu_j \tag{5.87}$$

Use of the same ordering procedure mentioned for the non-repairable system, allows us to form numerical distributions from which approximate confidence limits for both reliability indices at any confidence level can be obtained.

5.9.5 Algorithm and an illustrative system

The main steps of the Monte Carlo technique can be summarised as follows:

Step 1 : Generate the minimal cut-set and cut-node incidence matrices of the system.

Step 2 : Generate sample values for the steady-state probability of being in the down state and for the rate of departure from the failed state for each component using random number generators from an F- and Chi-square distribution respectively. Sample values must not be ordered.

Step 3 : Simulate sample minimal cut-set values using the component sample values from step 2 and the minimal cut-set matrix of step 1.

Step 4 : Simulate sample values for the probability and frequency of failure of the i-th node of the system using the information from the cut-node incidence matrix.

Step 5 : Sort the probability and frequency of i-node failure arrays in increasing order of magnitude.

Step 6 : From the ordered values of step 5 obtain confidence limits for the probability and frequency of i-node failure at any specified confidence level. If reliability indices are required for all or for special nodes of the system repeat 4, 5 and 6 for such nodes.

For illustrative purposes we use the reliability network of Fig. 1 to compute approximate confidence limits for the probability and frequency of failure of node 4. Components one through five were assumed to be equal, to follow the failure and repair exponential densities and were tested until all components failed.

One thousand samples of P_d and μ from Eqns. (5.83) and (5.85) respectively were generated for the components. These values combined through the minimal cut-set and cut-node incidence matrices simulated 1000-samples of the probability and frequency of node 4 failure. The node samples were ordered to construct an approximate step distribution. Tables XLI and XLII present the results for 1000 experiments using a component failure rate of 2 f/year and a mean-time-to-repair of 20 hours for a component sample size of 10. Columns 1 to 4 in both tables have the same interpretation as given in the non-repairable case.

Results were also obtained for 80, 500 and 1000 experiments with the same data as before but the sample size was assumed to be 16.

Tables XLIII and XLIV show the results for the probability and frequency of failure respectively. Comparison of the results of the 1000 experiments for sample sizes of 10 and 16 show that the larger the sample size the narrower the reliability intervals at any confidence level. It is observed that both the sample size and the number of trials, increase the computer time.

5.9.6 Comparison with the Moment approach

Table XLV shows that eighty cycles of the simulation process for the sample system of Fig. 1 resulted in data samples for the probability and frequency of system failure having mean and variance very close to those computed using the equations of the analytical approach. Hence, 80 cycles of the simulation process was deemed adequate.

Tables XLVI and XLVII compare the two-sided 80 and 90% confidence limits, for different sample sizes, obtained from the Monte Carlo with the same results provided by the Chebyshev's inequality for the probability and frequency of failure respectively. The upper bound provided by the Chebyshev's inequality is more conservative than the limit from the Monte Carlo simulation. They approach each other when the sample size increases.

Table XLVIII presents the computer time for the Monte Carlo approach for different sample sizes and 500 experiments. It is observed that the larger the sample size i.e., the more stable is the component data, the longer the computer time for the Monte Carlo technique. Obviously the time will be even longer still if we increase the number of trials. The Chebyshev's approach, although more pessimistic, has the advantage of being a one-experiment solution and its time solution is independent of the component sample size. The average computer time of the Chebyshev's approximation for the same cases of table XLVIII was 0.625 s. As the variance of the reliability indices decrease for highly stable components (see tables XVI and XVII) the bounds from the Chebyshev's inequality will improve as the component sample size increase.

TABLE XLI

APPROXIMATE CONFIDENCE LIMITS FROM THE MONTE CARLO TECHNIQUE FOR THE PROBABILITY OF FAILURE

Component failure rate = 2 f/year
Component repair rate = 438 repairs/year
Component sample size = 10
Number of trials = 1000

Two-sided confidence level 1	Upper limit 2	Lower limit 3	One-sided confidence level 4
99.0	0.11354×10^{-3}	0.18715×10^{-4}	99.5
98.0	0.99245×10^{-4}	0.21985×10^{-4}	99.0
97.0	0.89982×10^{-4}	0.24545×10^{-4}	98.5
96.0	0.85320×10^{-4}	0.26095×10^{-4}	98.0
95.0	0.78378×10^{-4}	0.28334×10^{-4}	97.5
90.0	0.74842×10^{-4}	0.29876×10^{-4}	95.0
80.0	0.68322×10^{-4}	0.32120×10^{-4}	90.0
70.0	0.63162×10^{-4}	0.34686×10^{-4}	85.0
60.0	0.59447×10^{-4}	0.37234×10^{-4}	80.0
50.0	0.55660×10^{-4}	0.39377×10^{-4}	75.0

Computer time for tables XLI and XLII = 14.9 s. of CPU

TABLE XLII

APPROXIMATE CONFIDENCE LIMITS FROM THE MONTE CARLO TECHNIQUE FOR THE FREQUENCY OF FAILURE

Component failure rate = 2 f/year
Component repair rate = 438 repairs/year
Number of trials = 1000
Component sample size = 10

Two-sided confidence level 1	Upper limit 2	Lower limit 3	One-sided confidence level 4
99.0	0.106118	0.015220	99.5
98.0	0.091146	0.018642	99.0
97.0	0.081142	0.020209	98.5
96.0	0.073153	0.022182	98.0
95.0	0.068939	0.024022	97.5
90.0	0.066956	0.025227	95.0
80.0	0.060108	0.027326	90.0
70.0	0.055070	0.029681	85.0
60.0	0.051658	0.032263	80.0
50.0	0.048486	0.034276	75.0

TABLE XLIII

PROBABILITY OF FAILURE

APPROXIMATE CONFIDENCE LIMITS FROM THE MONTE CARLO TECHNIQUE FOR DIFFERENT NUMBER OF SIMULATIONS

λ_i = 2 f/year μ_i = 438 repairs/year sample size = 16

2-sided confid. level	80 - experiments		500 - experiments		1000-experiments	
	Upper limit	Lower limit	Upper limit	Lower limit	Upper limit	Lower limit
99.0	0.816×10^{-4}	0.233×10^{-4}	0.844×10^{-4}	0.221×10^{-4}	0.860×10^{-4}	0.225×10^{-4}
98.0	0.700×10^{-4}	0.251×10^{-4}	0.766×10^{-4}	0.253×10^{-4}	0.777×10^{-4}	0.256×10^{-4}
97.0	0.659×10^{-4}	0.264×10^{-4}	0.711×10^{-4}	0.274×10^{-4}	0.722×10^{-4}	0.276×10^{-4}
96.0	0.639×10^{-4}	0.278×10^{-4}	0.682×10^{-4}	0.296×10^{-4}	0.677×10^{-4}	0.295×10^{-4}
95.0	0.614×10^{-4}	0.286×10^{-4}	0.658×10^{-4}	0.311×10^{-4}	0.642×10^{-4}	0.311×10^{-4}
90.0	0.602×10^{-4}	0.300×10^{-4}	0.636×10^{-4}	0.323×10^{-4}	0.619×10^{-4}	0.322×10^{-4}
80.0	0.559×10^{-4}	0.319×10^{-4}	0.605×10^{-4}	0.336×10^{-4}	0.577×10^{-4}	0.346×10^{-4}
70.0	0.520×10^{-4}	0.332×10^{-4}	0.567×10^{-4}	0.353×10^{-4}	0.544×10^{-4}	0.362×10^{-4}
60.0	0.500×10^{-4}	0.362×10^{-4}	0.534×10^{-4}	0.365×10^{-4}	0.514×10^{-4}	0.379×10^{-4}
50.0	0.477×10^{-4}	0.382×10^{-4}	0.508×10^{-4}	0.394×10^{-4}	0.492×10^{-4}	0.396×10^{-4}
Time*	3.10		8.60		19.40	

* Computer time in seconds of CPU for tables XLIII and XLIV

TABLE XLIV

FREQUENCY OF FAILURE

APPROXIMATE CONFIDENCE LIMITS FROM THE MONTE CARLO TECHNIQUE
FOR DIFFERENT NUMBER SIMULATIONS

λ_i = 2 f/year μ_i = 438 repairs/year sample size = 16

2-sided confid. level	80 - experiments		500 - experiments		1000 - experiments	
	Upper limit	Lower limit	Upper limit	Lower limit	Upper limit	Lower limit
99.0	0.068789	0.018970	0.075004	0.018019	0.078305	0.019258
98.0	0.059203	0.021769	0.068794	0.021400	0.068699	0.021614
97.0	0.057627	0.023003	0.064587	0.022943	0.064372	0.023395
96.0	0.057186	0.024773	0.061666	0.024746	0.060134	0.024654
95.0	0.053620	0.025260	0.059069	0.026049	0.057546	0.026579
90.0	0.051733	0.026061	0.057234	0.027037	0.055374	0.027761
80.0	0.047365	0.027598	0.054259	0.028512	0.051075	0.030033
70.0	0.045595	0.029213	0.050662	0.030060	0.048230	0.031573
60.0	0.043263	0.031013	0.047824	0.031571	0.046354	0.033388
50.0	0.039641	0.032914	0.045105	0.034078	0.043638	0.034634

TABLE XLV

COMPARISON OF ESTIMATE AND VARIANCE OBTAINED ANALYTICALLY AND BY MONTE CARLO FOR DIFFERENT NUMBER OF TRIALS

λ = 2 f/year

μ = 438 repairs/year

Sample size = 32

Analytical solution : $P_f = 0.44158 \times 10^{-4}$

$Var(P_f) = 0.10917 \times 10^{-9}$

$f_f = 0.03760$

$Var(f_f) = 0.52265 \times 10^{-4}$

Monte Carlo size	Probability of failure				Frequency of failure			
	Estimate	% Diff.	Variance	% Diff.	Estimate	% Diff.	Variance	% Diff.
20	0.40790×10^{-4}	8.3	0.13461×10^{-9}	18.9	0.03990	5.8	0.94401×10^{-4}	44.6
40	0.46482×10^{-4}	5.0	0.12722×10^{-9}	14.2	0.03591	4.7	0.80533×10^{-4}	35.1
50	0.43105×10^{-4}	2.4	0.10225×10^{-9}	6.8	0.03703	1.5	0.70594×10^{-4}	26.0
80	0.44241×10^{-4}	0.2	0.11325×10^{-9}	5.5	0.03777	0.5	0.66276×10^{-4}	16.6

TABLE XLVI

COMPARISON OF CONFIDENCE LIMITS OBTAINED FROM THE CHEBYSHEV'S INEQUALITY AND BY MONTE CARLO SIMULATION FOR THE SAMPLE SYSTEM IN FIGURE 1

PROBABILITY OF FAILURE

λ = 2 f/year μ = 438 repairs/year

Sample size	Confidence level	Lower limit			Upper limit		
		Analytical Chebyshev's App.	Monte Carlo	% diff.	Analytical Chebyshev's App.	Monte Carlo	% diff.
2	80	0.0	$.24406 \times 10^{-4}$	-	$.25367 \times 10^{-3}$	$.15766 \times 10^{-3}$	60.9
	90	0.0	$.19387 \times 10^{-4}$	-	$.30755 \times 10^{-3}$	$.17049 \times 10^{-3}$	80.4
4	80	0.0	$.28339 \times 10^{-4}$	-	1.43972×10^{-4}	$.92113 \times 10^{-4}$	36.0
	90	0.0	$.26578 \times 10^{-4}$	-	1.75081×10^{-3}	$.10396 \times 10^{-3}$	68.4
8	80	0.04069×10^{-4}	$.30804 \times 10^{-4}$	86.8	1.02401×10^{-4}	$.69031 \times 10^{-4}$	48.3
	90	0.0	$.28057 \times 10^{-4}$	-	1.22766×10^{-4}	$.77275 \times 10^{-4}$	58.9
16	80	0.13396×10^{-4}	$.31873 \times 10^{-4}$	58.0	$.80568 \times 10^{-4}$	$.55947 \times 10^{-4}$	44.0
	90	0.0	$.30019 \times 10^{-4}$	-	$.94480 \times 10^{-4}$	$.60237 \times 10^{-4}$	56.8
32	80	0.20795×10^{-4}	$.35286 \times 10^{-4}$	41.1	$.67521 \times 10^{-4}$	$.52114 \times 10^{-4}$	29.6
	90	0.11117×10^{-4}	$.33220 \times 10^{-4}$	66.5	$.77199 \times 10^{-4}$	$.54512 \times 10^{-4}$	41.6
64	80	0.26424×10^{-4}	$.36788 \times 10^{-4}$	28.2	$.59202 \times 10^{-4}$	$.48297 \times 10^{-4}$	22.6
	90	0.19636×10^{-4}	$.36098 \times 10^{-4}$	45.6	$.65980 \times 10^{-4}$	$.49113 \times 10^{-4}$	34.4
128	80	0.30613×10^{-4}	$.37017 \times 10^{-4}$	17.3	$.53699 \times 10^{-4}$	$.45039 \times 10^{-4}$	19.2
	90	0.25832×10^{-4}	$.35538 \times 10^{-4}$	27.3	$.58480 \times 10^{-4}$	$.46817 \times 10^{-4}$	24.9

TABLE XLVII

COMPARISON OF CONFIDENCE LIMITS OBTAINED FROM THE CHEBYSHEV'S INEQUALITY AND BY MONTE CARLO SIMULATION FOR THE SAMPLE SYSTEM IN FIGURE 1

FREQUENCY OF FAILURE

$\lambda \doteq 2$ f/year $\mu = 438$ repairs/year

Sample Size	Confidence level	Lower limit			Upper limit		
		Analytical Chebyshev's ineq.	Monte Carlo	% diff.	Analytical Chebyshev's ineq.	Monte Carlo	% diff.
2	80	0.0	.0170057	-	.162316	.102333	58.6
	90	0.0	.0138928	-	.199597	.187130	6.7
4	80	0.0	.0242289	-	.1003648	.0857474	17.0
	90	0.0	.0215687	-	.1218895	.0942943	29.3
8	80	.0075509	.0220612	65.8	.0755891	.0635672	18.9
	90	.0065403	.0202890	67.8	.0896803	.0742623	20.8
16	80	.0156010	.0275988	43.5	.0620790	.0473646	31.1
	90	.0059751	.0260608	77.1	.0717049	.0517327	38.6
32	80	.0214345	.0297346	27.9	.0537655	.0474789	13.2
	90	.0147385	.0282148	47.8	.0604615	.0488776	23.7
64	80	.024151	.0326170	26.0	.049869	.0418941	19.0
	90	.0209732	.0314058	33.2	.0530468	.0434758	22.0
128	80	.0287431	.0322662	10.9	.0447169	.0397986	12.4
	90	.0254349	.0315736	19.4	.0480251	.0407258	17.9

TABLE XLVIII

COMPUTER TIME OF 500-EXPERIMENT MONTE CARLO TECHNIQUE FOR THE PROBABILITY AND FREQUENCY OF NODE FAILURE

Sample size	Computer Time (seconds of CPU)
2	4.10
4	4.50
8	5.98
16	8.60
32	14.40
64	25.90
128	49.30

5.10 Distributions of the reliability indices

Determination of the exact distribution of the probability and frequency of system failure does not seem practicable. Firstly, the probability transformations involved in the minimal cut-set formulae with the exponential failure and repair models do not result in a tractable final distribution. Secondly, different distributional forms for the system components would seem to prevent general solutions for the required distributions.

The approach adopted here was to use the data produced by Monte Carlo simulation (section 5.9) for the sample network in Fig. 1 to discover the distributions of the probability and frequency of system failure. All components of the sample network were assumed to follow the exponential failure and repair models. The normal distribution was tested by statistical goodness-of-fit tests and by probability plotting. In both cases, repairable and nonrepairable, the normal distribution was found to provide a good fit to the data produced by the Monte Carlo simulation for the sample system when the component data had been obtained from high sample sizes i.e. when the data is more stable. Figures 2 and 3 illustrate the fit of the normal distribution to the data produced by Monte Carlo simulation for the sample system in the nonrepairable and repairable case respectively. Tables XLIX and L show the results of 4 selected tests of the Chi-square test for different sample sizes in the nonrepairable and repairable cases respectively.

Table LI to LIII compare limits for two-sided 80 and 90 per cent confidence intervals on the reliability indices when repair is or not permitted respectively. The agreement between lower and upper confidence limits obtained by the normal approximation and by Monte Carlo simulation

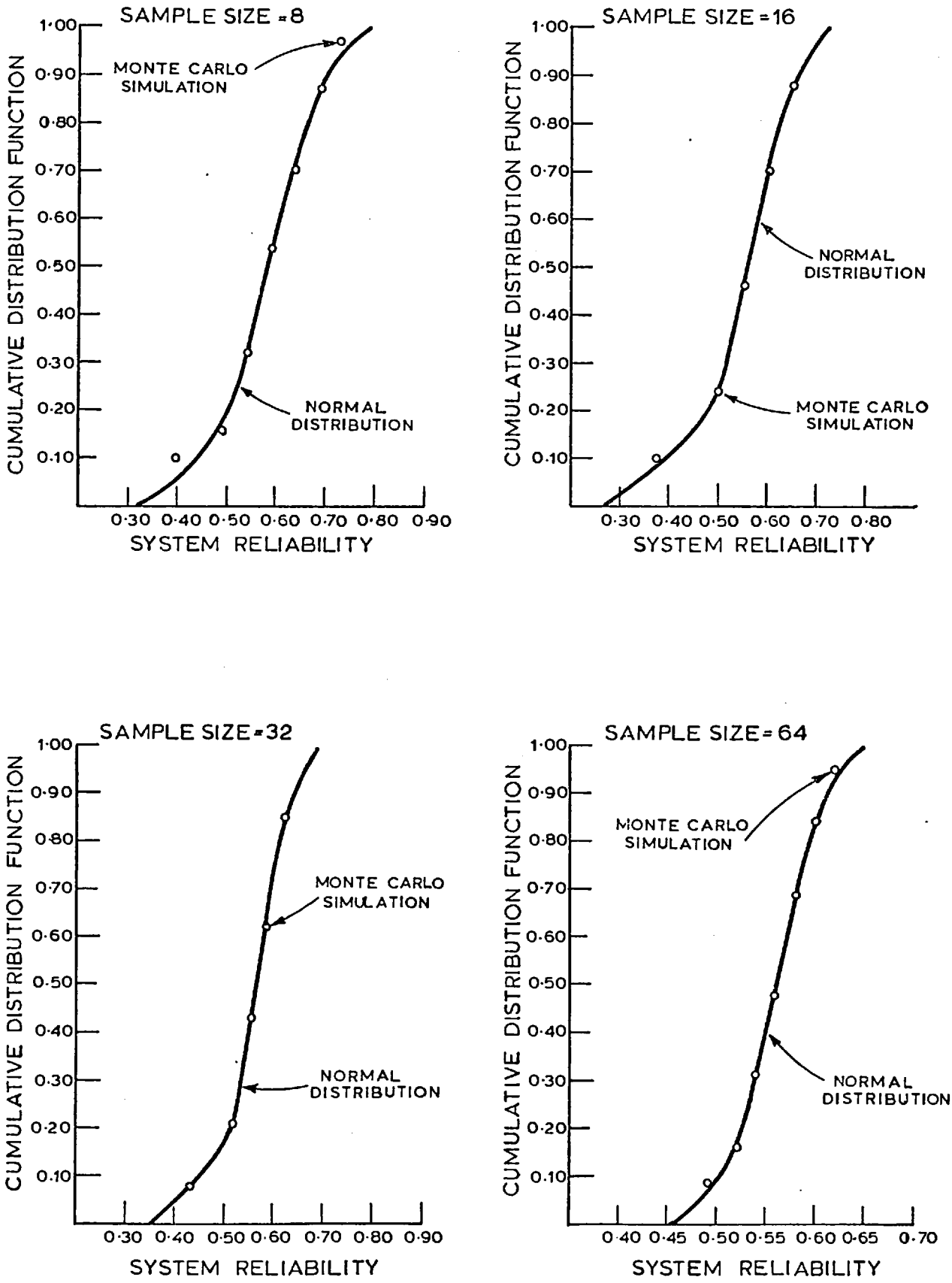


Fig. 2. Probability plotting of the reliability of the system in Fig. 1

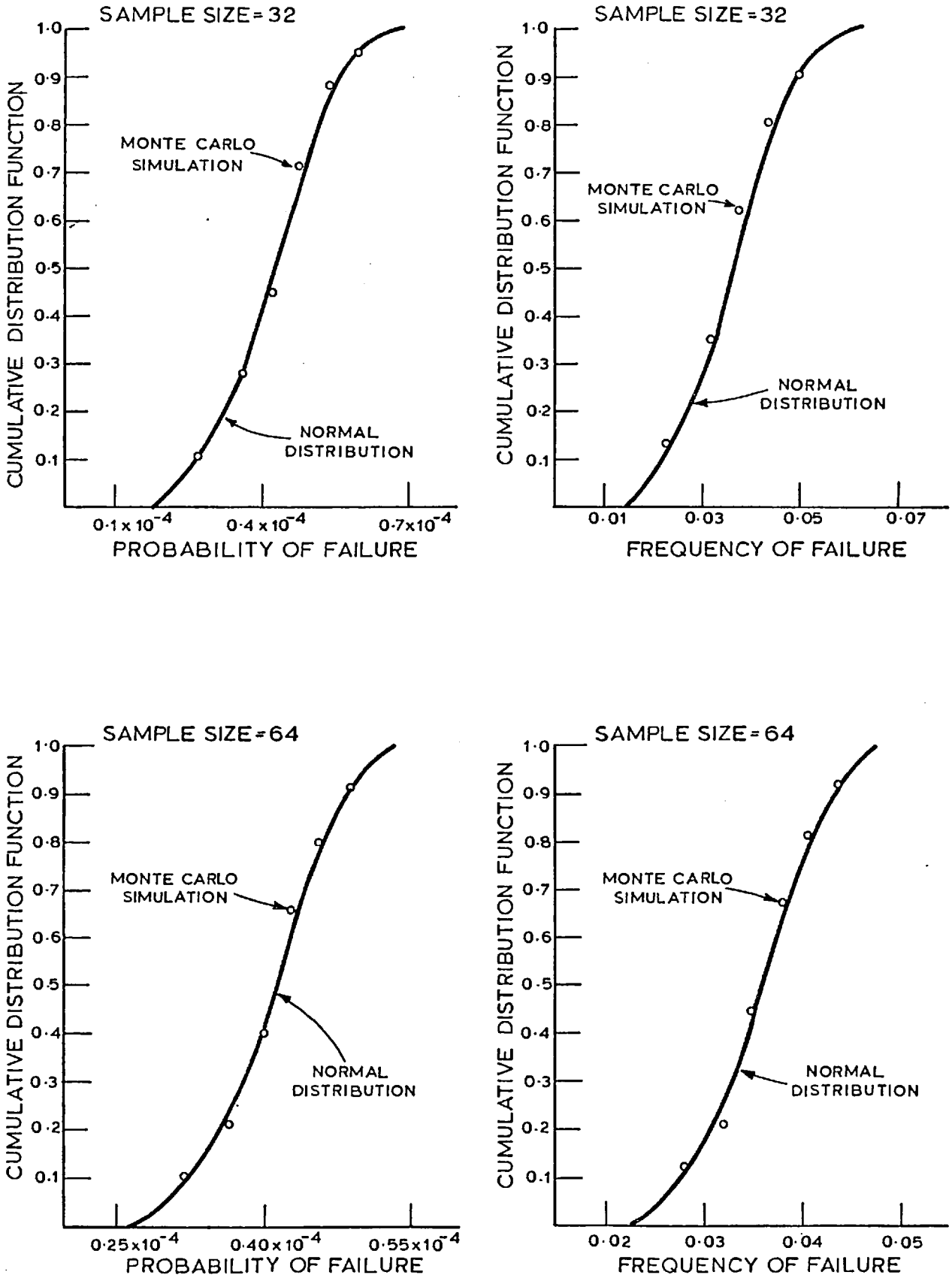


Fig. 3. Probability plotting of the reliability indices for the system in Fig. 1

TABLE XLIX

GOODNESS-OF-FIT TEST FOR NORMAL DISTRIBUTION
CASE OF NONREPAIRABLE COMPONENTS

Component MTTF = 4380 hrs.

t_m/MTTF = 0.5

No. of trials = 100

Sample size	Test	Degrees of freedom	Theoretical $\chi^2_{0.90}$	Computed χ^2
2	1	3	6.25	22.22
	2	2	4.61	11.87
	3	3	6.25	14.71
	4	4	7.78	20.85
4	1	4	7.78	11.43
	2	4	7.78	10.48
	3	4	7.78	15.32
	4	4	7.78	9.83
8	1	5	9.24	8.94
	2	4	7.78	4.46
	3	4	7.78	3.97
	4	5	9.24	4.23
16	1	5	9.24	8.13
	2	3	6.25	3.34
	3	4	7.78	4.68
	4	4	7.78	7.06
32	1	4	7.78	5.89
	2	4	7.78	2.12
	3	3	6.25	2.42
	4	4	7.78	2.37
64	1	3	6.25	2.56
	2	4	7.78	3.02
	3	3	6.25	2.50
	4	5	9.24	2.22

TABLE L

GOODNESS-OF-FIT TEST FOR NORMAL DISTRIBUTION
CASE OF REPAIRABLE COMPONENTS

λ = 2 f/year
μ = 438 repairs/year
No. of trials = 80

Sample size	Test	Degrees of freedom	Theoretical $\chi^2_{0.90}$	Computed χ^2
PROBABILITY OF FAILURE				
8	1	3	6.25	13.52
	2	4	7.78	19.09
	3	2	4.61	14.76
	4	4	7.78	23.15
16	1	4	7.78	7.81
	2	5	9.24	10.83
	3	4	7.78	16.05
	4	4	7.78	21.19
64	1	4	7.78	4.35
	2	4	7.78	3.92
	3	4	7.78	6.80
	4	4	7.78	2.67
FREQUENCY OF FAILURE				
8	1	2	4.61	29.80
	2	4	7.78	32.08
	3	3	6.25	19.44
	4	4	7.78	16.62
16	1	4	7.78	9.22
	2	5	9.24	9.71
	3	4	7.78	10.83
	4	4	7.78	16.18
64	1	5	9.24	7.77
	2	3	6.25	6.07
	3	4	7.78	4.22
	4	4	7.78	3.66

TABLE LI

COMPARISON OF CONFIDENCE LIMITS OBTAINED FROM THE NORMAL APPROXIMATION AND BY MONTE CARLO SIMULATION

Component MTTT = 4380 hrs.

No. of trials = 100

Sample size	Confidence level	Lower limit			Upper limit		
		Normal approximation	Monte Carlo	% Diff.	Normal approximation	Monte Carlo	% Diff.
8	80	0.41921	0.48536	13.6	0.86015	0.74109	16.1
	90	0.35635	0.40294	11.6	0.92301	0.76766	20.2
16	80	0.44927	0.50163	10.4	0.75915	0.68755	10.4
	90	0.40508	0.46206	12.3	0.80334	0.69772	15.1
32	80	0.47723	0.51588	7.5	0.69543	0.65505	6.2
	90	0.44613	0.48427	7.9	0.72653	0.67297	8.0

LII

COMPARISON OF CONFIDENCE LIMITS OF THE PROBABILITY OF FAILURE OBTAINED FROM THE NORMAL APPROXIMATION AND BY MONTE CARLO SIMULATION

λ = 2 f/year

μ = 438 repairs/year

No. of trials = 80

Sample size	Confidence level	Lower limit			Upper limit		
		Normal approximation	Monte Carlo	% Diff.	Normal approximation	Monte Carlo	% Diff.
32	80	$.30784 \times 10^{-4}$	$.35286 \times 10^{-4}$	12.8	$.57532 \times 10^{-4}$	$.52114 \times 10^{-4}$	10.4
	90	$.26970 \times 10^{-4}$	$.33220 \times 10^{-4}$	18.8	$.61346 \times 10^{-4}$	$.54512 \times 10^{-4}$	12.5
64	80	$.33432 \times 10^{-4}$	$.36788 \times 10^{-4}$	9.1	$.52194 \times 10^{-4}$	$.48297 \times 10^{-4}$	8.1
	90	$.30756 \times 10^{-4}$	$.36098 \times 10^{-4}$	14.8	$.54870 \times 10^{-4}$	$.49113 \times 10^{-4}$	11.7
128	80	$.35549 \times 10^{-4}$	$.37017 \times 10^{-4}$	4.0	$.48763 \times 10^{-4}$	$.45039 \times 10^{-4}$	8.3
	90	$.33664 \times 10^{-4}$	$.35538 \times 10^{-4}$	5.3	$.50648 \times 10^{-4}$	$.46817 \times 10^{-4}$	8.2

TABLE LIII

COMPARISON OF CONFIDENCE LIMITS OF THE FREQUENCY OF FAILURE OBTAINED FROM THE NORMAL APPROXIMATION AND BY MONTE CARLO SIMULATION

λ = 2 f/year

μ = 438 repairs/year

No. of trials = 80

Sample size	Confidence level	Lower limit			Upper limit		
		Normal approximation	Monte Carlo	% Diff.	Normal approximation	Monte Carlo	% Diff.
32	80	0.0283463	0.0297346	4.7	0.0468537	0.0474789	1.3
	90	0.0257075	0.0282148	8.9	0.0494925	0.0488776	1.3
64	80	0.0305187	0.0322662	5.4	0.0435013	0.0397986	9.3
	90	0.0286677	0.0315736	9.2	0.0453523	0.0407258	11.4
128	80	0.0296683	0.0326170	9.9	0.0435012	0.0418941	3.8
	90	0.0308543	0.0314058	1.8	0.0426057	0.0434758	2.0

is seen to be almost equal for the nonrepairable condition in table LI. The same comparison for the repairable condition in tables LII and LIII indicates that the agreement is better for the frequency of failure than for the probability of failure. In both cases for all tests made the analytical results (normal approximation) are more optimistic (lower limit) or pessimistic (upper limit) than those obtained from the simulation process. For small samples sizes (8 and 16) the upper limit of the analytical method for the nonrepairable case would appear to give differences slightly greater than the lower confidence limit. For high sample size both have almost equal deviations from the simulation results. A similar tendency was observed for the repairable condition but for both cases the agreement improves as the sample size increases.

5.11 Conclusion

This chapter has presented two different approaches for taking into account the effect of uncertain component failure and repair rates on the probability of failure of nonrepairable and repairable systems and on the frequency of failure of repairable systems. This is accomplished by computing approximate confidence limits for these reliability indices. The interval estimates define upper and lower bounds such that these random variables lie between these limits with a specified probability.

The analytical approach for the computation of confidence limits is based on the low order moments of the variables and uses Chebyshev's inequality to bound the probabilities that these random variables lie within a certain interval. The formulae for the estimate and variance of minimal cut-sets up to the third order have been presented.

The Monte Carlo simulation approach presented for the repairable case uses the one-to-one correspondence between the steady state probability of an element being down and the ratio of its repair and failure rates to define F-distributed variables from which sample values can be generated. These sample values for each component are combined through the minimal cut-set relationship between system reliability and components to generate system samples. A simulation process is made to generate the numerical distribution from which confidence limits can be computed.

The results obtained have proved that the minimal cut-set approach do not amplify the uncertainty in the component data. The tests performed show that the performance of both approaches improve as the component data is more stable and bounds from the Chebyshev's inequality are more conservative than those obtained from the Monte Carlo simulation approach. A case study has indicated that the reliability indices from the minimal cut-set approach for systems with highly stable components seem to obey a normal distribution to a reasonable approximation when the components follow the exponential failure and repair models.

CHAPTER VI

SENSITIVITY ANALYSIS AND RELIABILITY CALCULATIONS IN DIGITAL PROTECTION SYSTEMS

6.1 Introduction

The object of sensitivity analysis of system reliability indices is to improve the reliability of the system through knowledge of the component parameters that have a critical effect. Indeed, without performing such a study it is quite difficult to determine the effect of fractional changes in component parameters on fractional changes of the system reliability indices.

The aim of the present chapter is to present a sensitivity analysis approach based on the minimal cut-set approximation which combines data on the variation of the component reliability parameters with information of the sensitivity coefficients of the system reliability indices. It also presents a practical application of the methods given in preceding chapters by examining a proposed digital hardware configuration[7] for integrated protection of one corner of a mesh type high voltage substation used typically on the 400 kV system of the C.E.G.B. General alternative configurations are investigated and a sensitivity analysis of the reliability indices is done for a particular application.

6.2 Sensitivity coefficients

The sensitivity coefficient of function u with respect to parameter x, designated by S_x^u, is defined as the limit of the fractional change in u, also called proportional error, divided by the fractional change in x as Δx and Δu become small, i.e.

$$S_x^u = \lim_{\substack{\Delta x \to 0 \\ \Delta u \to 0}} \frac{\Delta u/u}{\Delta x/x} = \frac{du/u}{dx/x} = \frac{x}{u} \frac{du}{dx} \tag{6.1}$$

Solving Eqn. (6.1) for the fractional change in u yields

$$\frac{du}{u} = S_x^u \cdot \frac{dx}{x} \tag{6.2}$$

If u is a function of several variables, we can write the total differential of u as

$$du = \frac{\partial u}{\partial x_1} dx_1 + \frac{\partial u}{\partial x_2} dx_2 + \dots + \frac{\partial u}{\partial x_n} dx_n$$

Dividing by u yields

$$\frac{du}{u} = \frac{\partial u}{\partial x_1} \frac{dx_1}{u} + \frac{\partial u}{\partial x_2} \frac{dx_2}{u} + \dots + \frac{\partial u}{\partial x_n} \frac{dx_n}{u} \tag{6.3}$$

If we replace du/dx with $\partial u/\partial x$ in Eqn. (6.1) because u is a function of several variables and substitute into Eqn. (6.3), we obtain

$$\frac{du}{u} = S_{x_1}^u \frac{dx_1}{x_1} + S_{x_2}^u \frac{dx_2}{x_2} + \dots + S_{x_n}^u \frac{dx_n}{x_n} \tag{6.4}$$

This equation is extremely useful since each sensitivity coefficient appears as a weighting factor which tells us the effect of fractional changes in x_i on the fractional change in u. Such an equation enables the designer to evaluate his design and choice of component in terms of parameter variations, so that he will be able to identify which component improvements will result in the best system improvement.

6.3 Application to system reliability indices

A general procedure for a sensitivity analysis of the system reliability indices found from the minimal cut-set approach can be summarised as follows :

1. Define the reliability network of the system.
2. Find its minimal cut-set and cut-node incidence matrices.
3. Develop an equation for the system reliability indices.
4. Obtain the first partial derivatives of the index equations with respect to each independent component reliability measure or parameter.
5. Solve the index equation for the estimated values of each component reliability measure or parameter.
6. Solve the partial derivatives for the estimated values.
7. From the values obtained in 5) and 6), determine the sensitivity coefficient matrices for the system reliability indices.

The first-partial derivatives of the indices equations with respect to each independent component parameter were used in chapter V when the variance equations of such indices were formulated. For each minimal cut-set these derivatives were identified as b-coefficients with the parameter being indicated as a subindex of the b's, e.g., $b_{p\lambda_i}$ was the first-partial derivative of the steady-state probability of failure with respect to the failure rate of the i-th component. The total sensitivity coefficient of any one of the indices with respect to each component parameter or reliability measure will be the summation of the respective b-coefficients from all the minimal cut-sets which include the particular component.

For the steady-state probability of failure of a system with NMC minimal cut-sets, the sensitivity coefficients for node k are given by

$$S^{n_k}_{P_f,\lambda_j} = \frac{\lambda_j}{P_f} \sum_{i=1}^{NMC} b^{i,n_k}_{p\lambda_j} \qquad (6.5)$$

$$S^{n_k}_{P_f,\mu_j} = \frac{\mu_j}{P_f} \sum_{i=1}^{NMC} b^{i,n_k}_{p\mu_j} \tag{6.6}$$

where $b^{i,n_k}_{p\lambda_j}$ means the first-partial derivative of the steady-state probability of cut i with respect to the failure rate of component j, being summated if and only if cut i is a minimal cut for node k.

From Eqns. (6.5) and (6.6) it can be shown that

$$S^{n_k}_{P_f,\lambda_j} = - S^{n_k}_{P_f,\mu_j} \tag{6.7}$$

implying that the fractional changes of the failure and repair rates of the j-component have the same weight on the fractional changes of the steady-state probability of failure of node k but that the same fractional change of each parameter produces the opposite effect on the node index.

For the steady-state frequency of failure the sensitivity coefficients for node k are

$$S^{n_k}_{f_f,\lambda_j} = \frac{\lambda_j}{f_f} \sum_{i=1}^{NMC} b^{i,n_k}_{f\lambda_j} \tag{6.8}$$

$$S^{n_k}_{f_f,\mu_j} = \frac{\mu_j}{f_f} \sum_{i=1}^{NMC} b^{i,n_k}_{f\mu_j} \tag{6.9}$$

Use of the minimal cut-set and cut-node incidence matrices allow us to compute efficiently sensitivity coefficients of all nodes in the system with respect to each component parameter.

The sensitivity coefficients for the probability of system failure $Q_s(t_m)$ in the interval from 0 to t_m are given by

$$S^{Q_s}_{Q_j} = \frac{Q_j(t_m)}{Q_s(t_m)} \sum_{i=1}^{NMC} b^{i}_{Q_j(t_m)} \tag{6.10}$$

6.4 Illustrative example

Tables I and II show examples of the sensitivity coefficient matrices for the nodes of Fig. 1 in chapter V, where all elements have been assumed to be equal with a failure rate of 2 f/year, a mean-time-to-repair of 20 hours and a sample size of 257. The information in both tables shows the extent to which the node reliability index is sensitive to changes in each component parameter. For example, Table I indicates that the probability of failure of node 2 is greatly reduced with a reduction of the failure rates of components 1 and 2 but is almost unaffected even for large reductions of the failure rates of all the other components. Node 3 shows the same effects as node 2 although component 2 has more influence than component 1, being the reverse for node 2. Node 4 is equally affected by all component changes except for component 3, which has a negligible effect on its probability of failure. For all nodes, no sizable improvement of the reliability indices is obtained from an improvement of component 3. Similar conclusions are obtained from the upper-half of Table II concerning the effects of component failure changes on node frequency of failure changes. The lower half of the same table shows the opposite effect on the frequency of failure from changes in the component repair rate. The example is a trivial one but illustrates the help that the designer can obtain from the sensitivity coefficients.

Tables III and IV present the effects of different component availability on the sensitivity coefficient of node 4 for its probability and frequency of failure. For all components except component 3, the higher the availability, the greater the node sensitivity to component parameter changes.

Tables V and VI present the sensitivity coefficients of node 4 for different sample sizes of a component failure rate estimate of 2

f/year and a repair rate estimate of 438 repairs/year. Both tables indicate that for all components but component 3 the more stable are the components i.e., the lower their variances, the greater the sensitivity coefficient for both reliability indices. This tendency is the same as saying that the estimates of the probability and frequency of failure are influenced by the variances of the component parameters.

6.5 Computer-operated protection reliability

As power systems grow in complexity, protective relays play a more and more important role in the detection and removal of faulted equipment. Traditionally the protection functions in the power system substation has been performed by analog and wired-logic circuitry. Continuing rapid advances in digital computer technology have prompted a re-evaluation of protective devices and techniques with the result that digital processors are beginning to be used for the protection and control of high voltage systems.

Quantitative determinations of reliability allow comparison of different alternatives of protection system configurations to evaluate the relative reliability of different schemes. For analog relays only a few groups of papers[66-70] were found related to the subject of protective relay reliability. There is now considerable literature describing the application of digital computers to protection tasks in substations. Most papers deal with proposals for the application of digital computers to transmission line protection, busbar protection and transformer protection. The use of digital computers for protection purposes is economically more attractive if it can be applied to a number of protection and control functions simultaneously. A group of papers[67],[71-75] have considered the concepts of integration of control and protection

TABLE I

NODE PROBABILITY OF FAILURE SENSITIVITY MATRIX

λ = 2 f/year μ = 438 repairs /year

Sample size = 257

Node / Component	Failure rate sensitivities*		
	2	3	4
1	0.987808	0.978909	0.493912
2	0.978909	0.987808	0.493912
3	0.008899	0.008899	0.004470
4	0.004450	0.004450	0.493912
5	0.004450	0.004450	0.493912

* Repair rate sensitivities are equal with opposite sign

TABLE II

NODE FREQUENCY OF FAILURE SENSITIVITY MATRIX

λ = 2 f/year μ = 438 repairs /year

Sample size = 257

Node / Component	Failure rate sensitivities		
	2	3	4
1	0.991617	0.978277	0.495821
2	0.978277	0.991617	0.495821
3	0.013340	0.013340	0.006715
4	0.006670	0.006670	0.495821
5	0.006670	0.006670	0.495821
Node / Component	**Repair rate sensitivities**		
	2	3	4
1	-0.495778	-0.486905	-0.247341
2	-0.486905	-0.495778	-0.247341
3	-0.008873	-0.008873	-0.004467
4	-0.004437	-0.004437	-0.247341
5	-0.004437	-0.004437	-0.247341

TABLE III

PROBABILITY OF FAILURE SENSITIVITY COEFFICIENTS
FOR DIFFERENT COMPONENT FAILURE RATES
μ = 438 repairs/year Sample size = 257

Failure rate	Component availability	Failure rate sensitivity*	
		Component 1,2,4,5	Component 3
2	0.995455	0.493912	0.004470
4	0.990959	0.491720	0.008820
8	0.982063	0.487394	0.017177
16	0.964758	0.478966	0.032610
32	0.931915	0.462953	0.059022
219	0.666667	0.332684	0.166342

* Repair rate sensitivities are equal with opposite sign

TABLE IV

FREQUENCY OF FAILURE SENSITIVITY COEFFICIENTS FOR DIFFERENT
COMPONENT AVAILABILITY
μ = 438 repairs/year Sample size = 257

Failure rate	Component availability	Failure rate		Repair rate	
		Component 1,2,4,5	Component 3	Component 1,2,4,5	Component 3
2	0.995455	0.495821	0.006715	-0.247341	-0.004467
4	0.990959	0.493608	0.013222	-0.245662	-0.008774
8	0.982063	0.489242	0.025637	-0.242328	-0.016935
16	0.964758	0.480738	0.048275	-0.235758	-0.031595
32	0.931915	0.464591	0.086102	-0.223024	-0.055304
219	0.666667	0.333526	0.222351	-0.111175	-0.111175

TABLE V

PROBABILITY OF FAILURE SENSITIVITY COEFFICIENTS
FOR DIFFERENCE SAMPLE SIZES

λ = 2 f/year μ = 438 repairs/year

Sample size	Failure rate*	
	Components 1,2,4,5	Components 3
2	0.250286	0.002265
4	0.333080	0.003014
8	0.399089	0.003612
16	0.442984	0.004009
32	0.468763	0.004242
64	0.482811	0.004369
128	0.490156	0.004434

* Repair rate sensitivities are equal with opposite sign

TABLE VI

FREQUENCY OF FAILURE SENSITIVITY COEFFICIENTS
FOR DIFFERENT SAMPLE SIZES

λ = 2 f/year μ = 438 repairs/year

Sample size	Failure rate		Repair rate	
	Components 1,2,4,5	Components 3	Components 1,2,4,5	Components 3
2	0.333593	0.004518	-0.166413	-0.003005
4	0.399457	0.005410	-0.199270	-0.003599
8	0.443210	0.006003	-0.221096	-0.003993
16	0.468890	0.006351	-0.233906	-0.004224
32	0.482878	0.006540	-0.240884	-0.004350
64	0.490190	0.006639	-0.244532	-0.004416
128	0.493930	0.006690	-0.246397	-0.004450

in a computer-based substation. Most of the pertinent papers present the possible benefits in terms of speed, flexibility and cost that could potentially be offered by digital computers in these applications. In general, some form of redundancy is implicit in achieving high reliability but no explicit and systematic procedure has been used for the reliability evaluation of different schemes.

In general any protection system has to satisfy two requirements: (i) it must operate in the presence of a fault and (ii) it must be selective i.e., it must not produce false operation. Existing relaying systems are dedicated devices which usually do not operate unless a fault occurs. Since faults are relatively rare, the system spends almost its entire life in a passive state. This is all right, of course, because the system needs to operate only when a fault occurs. However, it is not known if the system will operate correctly until the next fault occurs and the system is called upon to operate. Thus the probability of failure depends upon the frequency of faults occurring within the protected zone i.e., is a function of the number of operations the relay performs and is not a function of real time. Proposed computer based protection systems or substation control systems should be continually in an active state and have a probability of failure which is independent of the frequency of fault occurrence on the protected item. This allows convenient prediction (via self-testing) of successful operation should a fault occur.

A failure in a computer system can occur either as a hardware failure or as a software failure. Whereas the former can be determined from the hardware reliability figures, software reliability is extremely difficult to predict. In this context modularity of programs and of self-checking procedures have been suggested to make software related failure arbitrarily small. The reliability measures provided by the techniques presented in this thesis are based on the steady state values

of the probability and frequency of system failure. For a computer system which is intended to operate for 20 years or so, it is evident that all hardware failures must be repaired so steady state indices can be used to measure the hardware performance. Other quantities of interest such as availability, mean down time, mean up time, etc., are easily evaluated from probability and frequency of failure indices. The sensitivity coefficients described in section (6.2) are useful when deciding where the improvements must be located. The ability of computer-based protection schemes, among others, to anticipate power system transients and swings, to change protection characteristics and to provide flexible compensation for mutual coupling effects would effectively increase the security of the protection system against false operation.

Only one of the transmission line techniques(77-78) is known to have been implemented in a practical field trial in the protection of a 230 kV line. Therefore lack of service experience prevents a meaningful assessment to be made of the reliability of any form of computer-based protection and control scheme. It remains to be demonstrated that such computer schemes ensure a level of reliability equal to that achieved in conventional schemes. Nevertheless any system with self-checking facilities which allows prediction of its own operation, detects faults and provides alternative solutions, modifies the reliability concept used for conventional schemes where the faults are usually detected by failure of the equipment to operate.

6.6 A computer-based protection scheme

In order to study different techniques for digital protection of high voltage substations a model has been designed for one corner of a mesh substation. The details of the proposed hardware configuration including the model details were published in [7].

Fig. 1 shows the proposed hardware configuration for one mesh corner with the following blocks :

(i) Data acquisition interface (DAI) : Digital signals (switch positions) and analog signals (current, voltage) will be fed through an appropriate interface from the power system (substation equipment).

(ii) Data validation processor (DVP) : A small digital processor will control data collection in digital form, perform appropriate validation checks upon it and control data output to an intermediate store.

(iii) Data intermediate storage (DIS) : At the end of each data acquisition cycle the resulting words of validated data will be transferred to the protection processor by an interface which operates in a first-in, first-out (FIFO) mode.

(iv) Control and protection processor (CPP) : The data block from the DIS is transferred to the data base (DB) which is a portion of the CPP core store. Using the contents of the DB the CPP will perform the computations necessary for control and protection functions and provide suitable control strategy to the substation equipment.

6.7 Alternative configurations - Reliability evaluation

The general alternative configurations presented in Fig. 2 were investigated from a reliability point of view. Specifically excluded are current and voltage transformers and circuit breakers.

If all the component-blocks are assumed to be identical, with λ and μ as the failure and repair rate respectively, table VII lists the probability and frequency of failure for each system configuration.

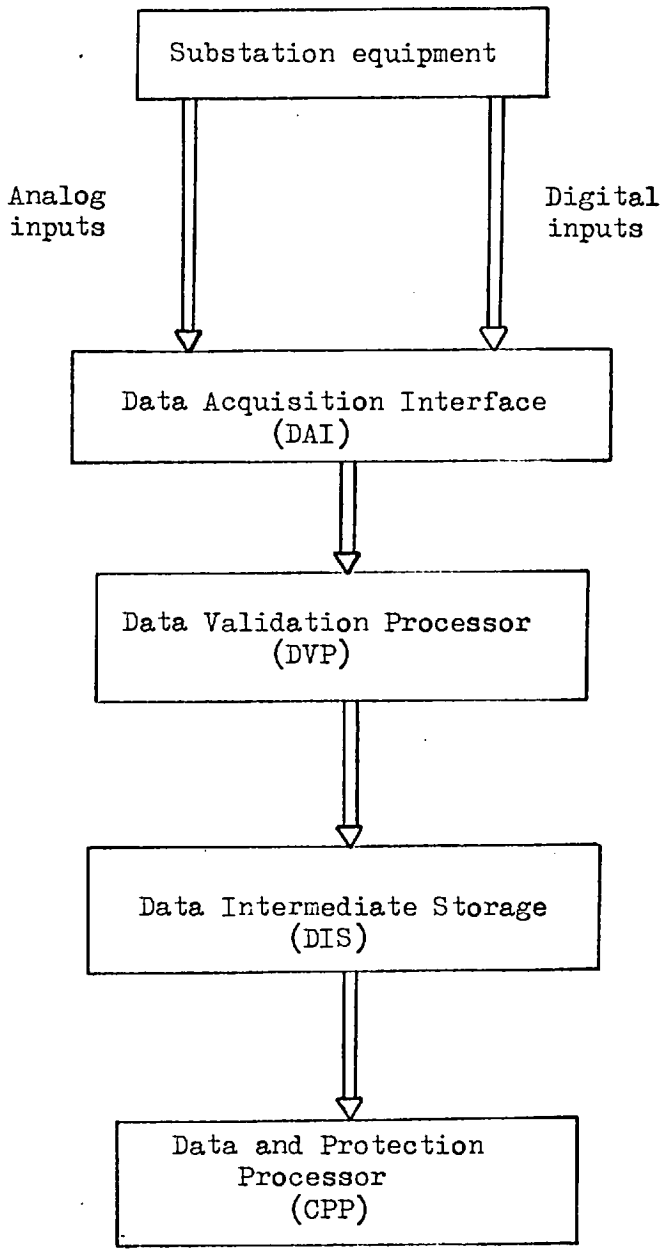


Figure 1. A hardware configuration for a digital protection system

P_d corresponds to the steady-state probability that a component is failed and is given by $\lambda/\lambda+\mu$.

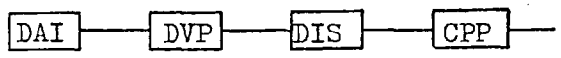
The five different availability expressions $(1 - P_f)$ derived from table VII are compared in Figure 3. Note that the duplex system with parallel operation of the processors in charge of the data validation and protection functions will offer the best solution. The two-out-of-three system is better than the single processor-single interface as long as $0.916 < 1 - P_d \leqslant 1.0$ and becomes worse than the single system for $0 < 1 - P_d \leqslant 0.916$, where $(1 - P_d)$ is the component-block availability. Table VIII presents a numerical comparison of the reliability indices of the five configurations.

Tables IX and X present the probability and frequency of failure of the duplex system for selected values of the component failure and repair rate respectively. Obviously both reliability indices increase as the component availability decrease. An increase of 20 times the MTTR of the components will have greater consequences on the system reliability than an increase of 20 times of its failure rate. Hence, the duplex system with parallel operation of the processors will have greater availability if themean-time-to-repair of each component is kept as low as possible i.e., the system must have a self-checking and fault-locating facility. This facility should continuously monitor the ability of the hardware to function correctly and in the event of a failure alarm, enables personnel to quickly locate the trouble.

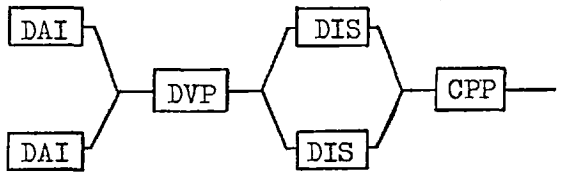
6.8 Reliability analysis in a digital protection system

The lack of proper data for each single component prohibits a particular reliability evaluation, so an experimental sensitivity analysis was conducted on the system shown in Fig. 4 to obtain the

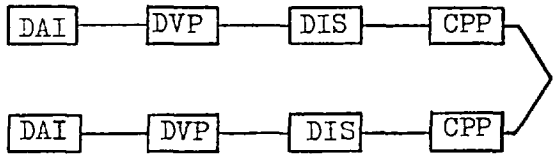
(1) Single processor - single interface



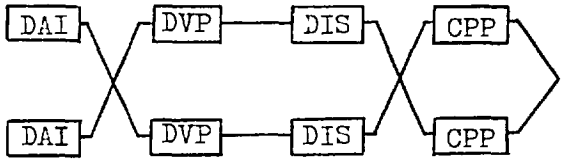
(2) Single processor - duplicate interface



(3) Duplex system-independent operation



(4) Duplex system - parallel operation of processors



(5) Two-out-of-three configuration

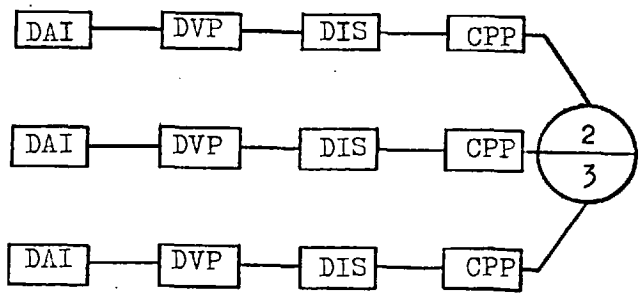


Figure 2. Alternative system configurations

TABLE VII

PROBABILITY AND FREQUENCY OF FAILURE FOR DIFFERENT SYSTEM CONFIGURATIONS

System configuration	Probability of failure	Frequency of failure
1	$4\ P_d$	$4\ P_d\ \mu$
2	$2P_d + 2P^2_d$	$(2P_d + 4P_d^2)\ \mu$
3	$16\ P_d^2$	$32\ P_d^2\ \mu$
4	$6\ P_d^2$	$12\ P_d^2\ \mu$
5	$48\ P_d^2$	$96\ P_d^2\ \mu$

TABLE VIII

COMPARISON OF RELIABILITY INDICES FOR DIFFERENT SYSTEM CONFIGURATIONS

λ = 2 f/year μ = 876 repairs/year

System configuration	Probability of failure	Frequency of failure	Up-time p/unit	Down-time (hrs)
1	0.911×10^{-2}	7.98	0.124	10.00
2	0.457×10^{-2}	3.99	0.25	10.03
3	0.830×10^{-4}	0.145	6.9	5.0
4	0.311×10^{-4}	0.055	18.2	4.95
5	0.249×10^{-3}	0.436	2.2	5.0

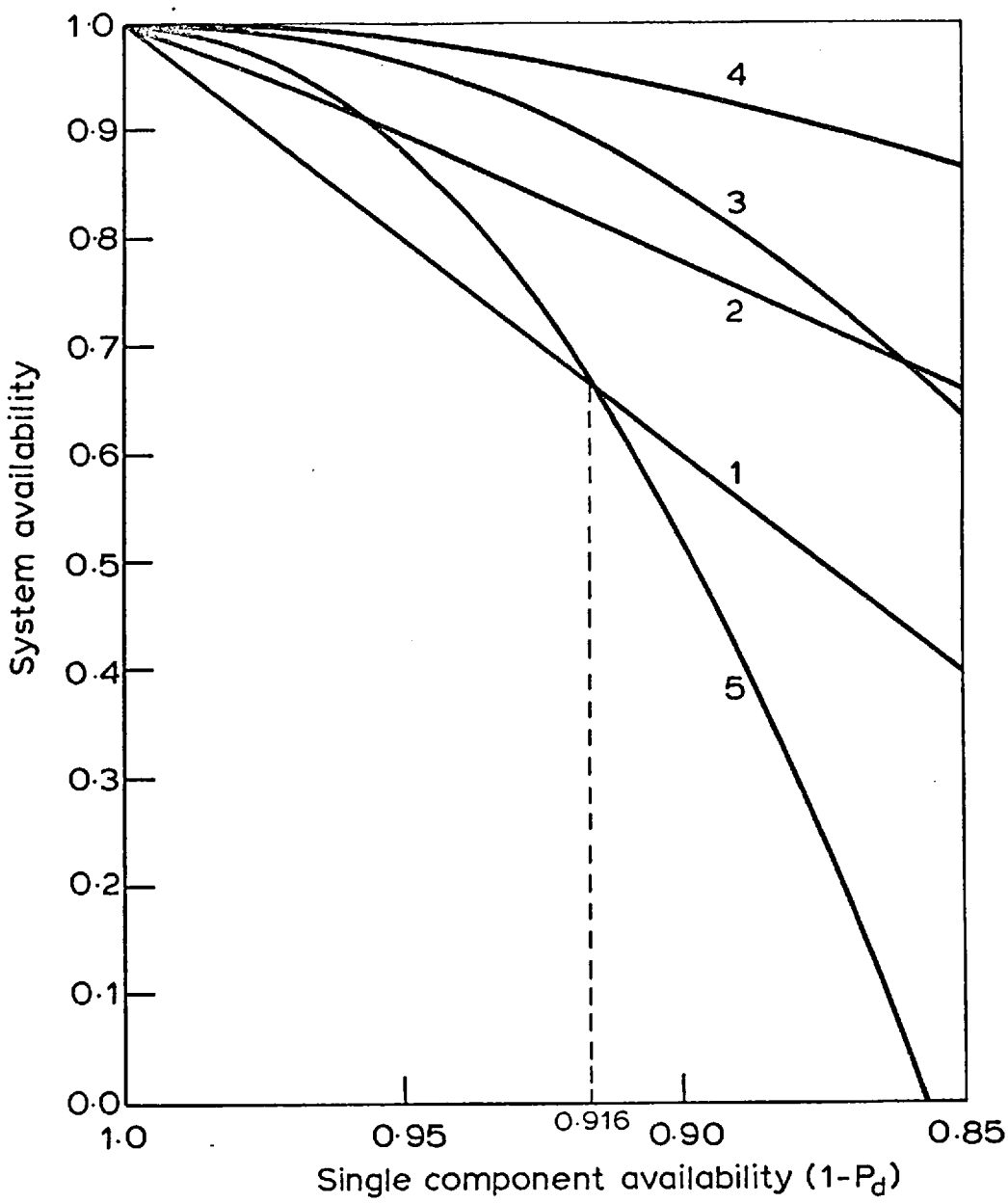


Fig. 3 : Availability comparison of five system configurations.

TABLE IX

RELIABILITY INDICES OF THE DUPLEX SYSTEM FOR DIFFERENT FAILURE RATES

Component MTTR = 5 hrs

Failure rate f/year	Component availability	Probability of failure	Frequency of failure
0.1	0.999943	0.194×10^{-7}	0.680×10^{-4}
0.2	0.999886	0.779×10^{-7}	0.273×10^{-3}
0.3	0.999829	0.175×10^{-6}	0.613×10^{-3}
0.4	0.999772	0.313×10^{-6}	0.109×10^{-2}
0.5	0.999715	0.487×10^{-6}	0.171×10^{-2}
0.6	0.999658	0.702×10^{-6}	0.246×10^{-2}
0.7	0.999601	0.955×10^{-6}	0.335×10^{-2}
0.8	0.999544	0.125×10^{-5}	0.438×10^{-2}
0.9	0.999487	0.158×10^{-5}	0.554×10^{-2}
1.0	0.999430	0.195×10^{-5}	0.683×10^{-2}
1.5	0.999145	0.439×10^{-5}	0.154×10^{-1}
2.0	0.998860	0.780×10^{-5}	0.273×10^{-1}

TABLE X

RELIABILITY INDICES OF THE DUPLEX SYSTEM FOR DIFFERENT REPAIR TIMES

Component failure rate = 2 f/year

MTTR (hrs)	Component availability	Probability of failure	Frequency of failure
1	0.999772	0.313×10^{-6}	0.00546
2	0.999544	0.125×10^{-5}	0.01093
4	0.999086	0.501×10^{-5}	0.02195
8	0.998177	0.199×10^{-4}	0.04367
10	0.988713	0.764×10^{-3}	1.34
20	0.977679	0.299×10^{-2}	2.62

variation of the probability and frequency of system failure from a wide range of the reliability component parameters, taking into account different grades of data dispersion.

Regardless of which system configuration is chosen for a particular system, it will be quite important to determine :

(i) how the estimate and variance of the reliability indices would be affected by the failure and repair rate of the components and for the dispersion of the component data,

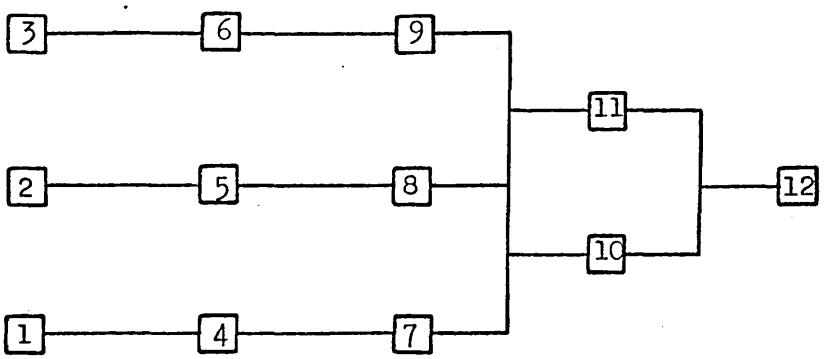
(ii) what will be the approximate confidence limits of the probability and frequency of failure and how the uncertainty of the data will affect these confidence limits

(iii) what component improvement will result in the maximum improvement of the system reliability indices.

A sensitivity analysis of this nature will indicate the accuracy of the estimates and will show which components have a critical effect. Indeed, without performing such a study it is quite difficult to determine how data uncertainty will affect the estimate and variance of the reliability indices and how much confidence should be placed in the estimates.

The present sensitivity analysis was also done to see how the reliability indices were affected when components of lower availabilities such as current transformers, are included in the reliability evaluation of the scheme.

The system configuration selected is a simple representation of the digital equivalent of an overcurrent scheme for a distribution feeder. The three-phase currents are fed into the system by different channels and a programmed device (element 10) acts as main protection. Simultaneously a back-up facility (element 11) incorporated in the system to prepare a data base through which the decision logic (element 12)



1 - 3 Current transformers

4 - 6 Sample-hold circuits

7 - 9 Analog/digital converters

10 Programmed device

11 Back-up processor

12 Decision logic

Figure 4. A simplified digital overcurrent protection scheme

provides an additional check on the correct operation of the main protection. The scheme in Fig. 4 is a simple application of the dedicated and integrated concepts introduced in [76] and is used to illustrate the reliability analysis that can be done with the methods presented in this thesis. The minimal cut-set algorithm of section (4.7) was used to find the 29 minimal cut-sets between nodes 1 and 10 of Fig. 4. The system considered has twelve components which are identified as digital (components 4 to 12) and non-digital (components 1 to 3) in the next sections.

6.8.1 Failure rate

The failure rate of each type of component was given selected values between 0.1 to 2 f/year for the digital components and between 0.5 and 10 f/year for the non-digital elements. In both cases the relation between minimum and maximum values assigned was 20.

The sensivitivy analysis assumed an average failure rate of 4 f/year for the non-digital components. This is an average calculated from [8] . The MTTF of digital components is approximately 8000 hours i.e., an average failure rate of 1 f/year.

The estimates and variances for both reliability indices for different failure rates of the digital components are listed in table XI. It is observed that an increase of 20 times the failure rate of the digital components would cause an increase of 2 and 20 times of the probability and frequency of failure estimates respectively. A similar analysis for the failure rate of the non-digital components gave almost constant values for the estimate and variance of both reliability indices. These values were equal to the ones presented in table XI for a failure rate of 1 f/year. The higher effect of the digital components, despite

its smaller failure rate, is due to the greater influence of the first-order cut-set produced by component 12. Similar results were found when the failure rate of both types of components were varied simultaneously.

6.8.2 Repair rate

In a test similar to that for failure rate, the mean time to repair of the components were given selected values between 1 and 20 hours. The average values are the same as used in the previous test.

The estimates and variances are presented in table XII. In this test the repair rate of both digital and non-digital components were varied simultaneously. It is observed that an increase of 20 times the repair time causes an increase in the probability of failure but has no influence in the frequency of failure. This index presents almost a constant value of approximately 1 encounters/year with a standard deviation of 0.07 encounters/year.

Tables XI and XII indicate that a variation of the mean time to repair of the components has greater effect on the probability of failure than a similar variation of its failure rate, the effect being reversed on the frequency of failure.

6.8.3 Coefficient of dispersion

This parameter refers to the quotient of the standard deviation and the estimate of the reliability indices. The test was performed to see what influence different grades of data uncertainty has on the dispersion of the reliability indices. The variable parameter was the sample size from which the component data has been obtained. Selected values between 4 and 32 were used for this parameter.

Table XIII shows the coefficient of dispersion of the estimates

of both reliability indices for different sample sizes of both components. For both indices, the more stable the component data the smaller the dispersion of the estimates, the effect being greater for the frequency of failure.

6.8.4 Confidence limit

An upper confidence limit was found for both reliability indices using the technique presented in section (5.8). The sample size of the components was varied between 4 and 32 for both types of components.

Table XIV presents a 90% upper confidence limit for the probability and frequency of failure for different sample size of the digital and non-digital components. Both results indicate that the greater the sample size the smaller the upper limit defined by Chebyshev's inequality.

6.8.5 Sensitivity coefficients

The results presented in tables XI to XIV give estimate, variance and an upper confidence limit for both reliability indices but they do not give clear information about the effect of an improvement in component rates on system reliability indices.

Table XV lists the sensitivity coefficients of each reliability index with respect to each component parameter of the system. These coefficients were calculated using the technique presented in section 6.3. It is observed in table XV that both reliability indices are increased if the failure rate of any one of the components is increased, the effect being greater from component 12. The probability of failure is reduced by increasing the repair rate i.e., reducing the mean-time-to-repair of any one of the components, especially 10 to 12. The frequency

of system failure is reduced by increasing the repair rate of components 1 to 11 but is increased if component 12 increases its repair rate.

6.9 Conclusion

From the different general configurations considered here as alternatives to the single processor-single interface proposed in [7] for the hardware of an integrated digital protection of one corner mesh substation, the duplex system with both processors operating in parallel offers the best solution based on a reliability criterion. A self-checking and fault-locating facility must be incorporated for maximum availability of the system.

A complete reliability and sensitivity analysis can be done for any proposed configuration using the techniques presented in this thesis. Such analysis should examine the incidence of data uncertainty in the estimates and variances of the indices and find how much confidence should be placed in these estimates. Similarly, it will identify what component improvements result in the best system reliability improvement.

TABLE XI

ESTIMATES AND VARIANCES OF THE SYSTEM RELIABILITY FOR DIFFERENT FAILURE RATES OF THE DIGITAL COMPONENTS

λ_{nd} = 4 f/year Sample size = 16

$r_d = r_{nd}$ = 5 hrs

Failure rate f/year	Component availability	Probability of failure		Frequency of failure	
		Estimate	Variance	Estimate	Variance
0.1	0.999943	0.609×10^{-4}	0.434×10^{-9}	0.10009	0.667×10^{-3}
0.2	0.999886	0.122×10^{-3}	0.174×10^{-8}	0.20012	0.267×10^{-2}
0.3	0.999829	0.183×10^{-3}	0.391×10^{-8}	0.30016	0.600×10^{-2}
0.4	0.999772	0.244×10^{-3}	0.694×10^{-8}	0.40021	0.107×10^{-1}
0.5	0.999715	0.304×10^{-3}	0.109×10^{-7}	0.50028	0.167×10^{-1}
0.6	0.999658	0.365×10^{-3}	0.156×10^{-7}	0.60036	0.240×10^{-1}
0.7	0.999601	0.426×10^{-3}	0.213×10^{-7}	0.70045	0.326×10^{-1}
0.8	0.999544	0.487×10^{-3}	0.278×10^{-7}	0.80056	0.426×10^{-1}
0.9	0.999487	0.548×10^{-3}	0.351×10^{-7}	0.90068	0.539×10^{-1}
1.0	0.999430	0.609×10^{-3}	0.433×10^{-7}	1.00081	0.665×10^{-1}
1.5	0.999145	0.913×10^{-3}	0.974×10^{-7}	1.50166	0.150
2.0	0.998860	0.122×10^{-3}	0.173×10^{-6}	2.00284	0.265

TABLE XII

ESTIMATE AND VARIANCE OF THE SYSTEM RELIABILITY FOR DIFFERENT MEAN TIME TO REPAIR OF THE COMPONENTS

λ_{nd} = 4 f/year r_{nd} = 5 hrs

λ_d = 1 f/year sample size = 16

Component MTTR (hrs)	Probability of failure		Frequency of failure	
	Estimate	Variance	Estimate	Variance
1	0.122×10^{-3}	0.174×10^{-8}	1.00024	0.666×10^{-1}
2	0.244×10^{-3}	0.694×10^{-8}	1.00038	0.666×10^{-1}
5	0.609×10^{-3}	0.433×10^{-7}	1.00081	0.665×10^{-1}
10	0.122×10^{-2}	0.173×10^{-6}	1.00156	0.664×10^{-1}
20	0.244×10^{-2}	0.689×10^{-6}	1.00321	0.661×10^{-1}

TABLE XIII

COEFFICIENT OF DISPERSION OF SYSTEM RELIABILITY INDICES FOR DIFFERENT SAMPLE SIZES OF THE COMPONENT DATA

λ_d = 4 f/year $r_{nd} = r_d$ = 5 hrs

λ_{nd} = 1 f/year

Sample size	Probability of failure % c.d.	Frequency of failure % c.d.
4	61.2	57.7
8	46.7	37.7
16	34.2	25.8
32	24.6	17.9

TABLE XIV

90% UPPER CONFIDENCE LIMIT OF RELIABILITY INDICES FOR DIFFERENT SAMPLE SIZES

λ_{nd} = 4 f/year λ_d = 1 f/year

$r_{nd} = r_d$ = 5 hrs

Sample size	Probability of failure	Frequency of failure
4	0.284×10^{-2}	3.580
8	0.202×10^{-2}	2.689
16	0.154×10^{-2}	2.154
32	0.124×10^{-2}	1.803

TABLE XV

SENSITIVITY COEFFICIENTS FOR THE PROBABILITY AND FREQUENCY OF FAILURE FOR THE SYSTEM IN FIG. 4

λ_{nd} = 4 f/year λ_d = 1 f/year

$r_{nd} = r_d$ = 5 hrs sample size = 16

Component	Probability of failure		Frequency of failure	
	Failure rate	Repair rate	Failure rate	Repair rate
1-3	0.000044	-0.000044	0.000140	-0.000093
4-9	0.000011	-0.000011	0.000035	-0.000023
10-11	0.000534	-0.000534	0.001139	-0.000569
12	0.936424	-0.936424	0.998054	0.000570

CHAPTER VII

CONCLUSIONS AND SCOPE FOR FUTURE RESEARCH

7.1 Use of data validation techniques in digital relaying systems

Power system protection with on-line digital computers requires a maximum availability of the data used for protection decisions. Protection statistics have shown that the second highest cause of fault-clearance equipment can be attributed to the presence of bad data. Data validation techniques possess the ability to detect and identify missing data and large unexpected errors in data points. These properties can easily be achieved by checking the measurements themselves, their differences against chosen bounds and cross-checks between analogue (current or voltage) and digital (switch position) values. Once errors have been detected and identified, a simple estimation routine based on any type of available redundancy can be used to improve the reliability of the relaying data base.

7.1.1 Zero data detection

Loss of data is a critical factor on the reliability of the relaying data base. Zero data detection by software requires consideration of the conversion tolerance of the analog-digital converters. The time taken by such checks will consume most of the time interval available between successive samples. Zero data detection by hardware is shown to be essential in the practical implementation of a relaying data validation algorithm.

7.1.2 Doubtful data identification

The use of a logic 1 as zero and error flags incorporated in the data-word structure was found to be extremely useful in the validation

procedure to optimise the time spent in data identification. Such a facility adds one and two extra bits to the word structure of switch and analogue data respectively. These flag informations ease the development of a systematic estimation procedure for the best use of the redundancy available in the measurement set.

7.1.3 Application to one-corner of a mesh substation

The critical factor in the design of a relaying data validation algorithm is the short time available between samples. If high sampling rates e.g. 32 samples per cycle, are required in a scheme similar to the computer-substation model described in chapter II, no valuable checks can be implemented and missing data may deteriorate the performance of the protection scheme. However the data validation algorithm and the estimation procedure proposed in the same chapter can be successfully applied in such a scheme using a maximum sampling rate of 16 samples/cycle. Other problems such as core storage is not critical. Use of assembly language was found to be essential for the practical implementation of any relaying data validation algorithm.

Special care must be taken in the design of the interface structure (sample-hold circuits, A/D converters and multiplexers) to minimize the input and conversion time, maximizing the time available for validation. Packaging group of switch positions in one data-word also contributes to this objective.

Off-line and on-line test results indicate that the check on two complementary bits representing each switch position provide a reliable method for the detection of any doubtful switch indication. The use of a sample-switch routine proved to allow correction of such doubtful indications. On-line results confirmed the reliable performance of the

zero data detection by hardware. Test results also showed that the use of a residual value and a residual counter work satisfactorily in the detection of missing data. The tolerance limit to be used requires special consideration of any unbalance present either in the power system or in each data acquisition link.

7.2 Reliability evaluation of complex system

The reliability evaluation of complex systems with unreliable branches and nodes is only feasible using the tie-set or cut-set approaches described in chapter III. For systems with an average node degree (elements incident on the node) greater than four and/or components of high reliability the cut-set approach is more efficient than the tie-set technique. Bounds obtained using the inclusion-exclusion principle have smaller errors than those obtained from the disjoint or independence approximation for both tie-set and cut-set methods.

7.3 Minimal cut-set enumeration

The enumeration of the minimal cut-sets of all nodes in a separable, nonseparable or acyclic directed graph can be solved with one application of the appropriate algorithm described in chapter IV. The combination of the novel concept of the cut-node incidence matrix and sparsity techniques produces very efficient algorithms (fast and small storage requirements) for complex systems.

7.4 Node failure problem in power systems

Unreliable nodes were shown to have considerable influence on the reliability indices of each load point in an electric power system, even for highly reliable nodes. The branch-node cut-set algorithm

described in chapter IV is computationally efficient and combined with the minimal cut-set formulae developed by Billinton and Grover[79] provides a sequential method for performing complete reliability calculations in distribution and transmission systems.

7.5 Probability and frequency of system failure

The new statistical formulation of the probability and frequency of system failure provides extra information (variances and confidence limits) which enables the system designer to evaluate the goodness of his design. The variance equations developed in chapter V can be an aid to the designer in determining the causes of excessive reliability index variance since he can study these equations and determine which $(\partial u/\partial x_i)^2 \quad \sigma_{x_i}^{\ 2}$ terms are contributing most heavily to index variance. He can decide whether to reduce the large terms by using more stable components (reduction of $\sigma_{x_i}^{2}$) or by modifying the system configuration in such a way that the sensitivity of the reliability index to a particular component parameter is reduced (reduction of $\partial u/\partial x_i$). Similarly the sensitivity coefficients presented in chapter VI provide useful information about where to locate redundancy for the greatest increase of the system reliability.

The moment approach and the use of Chebyshev's inequality presented in Chapter V always give more conservative results than the Monte Carlo technique but the agreement improves for highly stable components. The advantage of the former method is in a one-experiment constant-time solution whilst the computation time of the simulation procedure increases with the component sample size and the number of trials.

7.6 Reliability considerations in the design of digital protection systems

The double nature of protection functions requires the evaluation of two performance indices, the probability of failure to operate in the

presence of a fault and the security of the system against false operation. Moreover a proper evaluation also involves the determination of the frequency of such incidents. At the moment it appears that the only meaningful approach for the reliability assessment of any computer-based protection scheme is on the average value of failure and repair rates available for different types of digital components. In this sense the methods presented in this thesis represent a systematic approach for the comparative evaluation of the reliability of different alternatives. The same techniques are applicable if failure and repair rates are available for each type of incident and the analysis is based on the steady state values of the indices. However it remains to be demonstrated that the solution adopted for the hardware and software structure of a computer-based protection scheme will have a level of reliability and security better or worse than that achieved in conventional schemes.

7.7 Future work

1. Data validation

Research is required on the feasibility of a data validation structure in a digital micro-processor based relay.

2. Minimal cut-set enumeration

An algorithmic procedure is required for the enumeration of the minimal cut-sets of complex systems with an r-out-of-n configuration.

3. Variance and confidence limits

Further investigations into the Moment method approach is needed for the case of component data obtained from small sample sizes.

4. Sensitivity coefficients

Research on the extension of the method is needed so that when the system reliability indices are at a certain level the overall system cost is minimized.

5. Digital protection reliability

Further investigation is required on the determination of the resultant reliability of a computer-based protection system taking into account both the reliability of the operation in the presence of a fault and the reliability of the system against false operations.

7.8 Original contributions

i. the development of a data validation algorithm for the detection and identification of gross errors (missing or erroneous data) and the on-line testing of its performance using a laboratory computer-based substation model.

ii. design of an estimation procedure to provide replacements for loss data based on the results of the data validation algorithm and using either the inherent redundancy of electrical networks or redundant measurements.

iii. development and testing of algorithms for the enumeration of the set of minimal cuts of separable, nonseparable and acyclic directed graphs based on graph theory concepts.

iv. inclusion of the node failure problem in the reliability evaluation of distribution and transmission systems using a new tested branch-node minimal cut-set algorithm.

v. development and testing of two new approaches for the variance and confidence limits of the probability and frequency of failure of repairable systems using a Moment method or Monte Carlo simulation.

vi. formulation of the sensitivity of the probability and frequency of system failure to component failure and repair rates by means of sensitivity coefficients to be used in the redundancy location problem.

APPENDIX 1

1.0 Minimal path examples of table II - Chapter III : All sets of paths have been randomly generated.

Network	Minimal paths	
	Binary representation	Base 10
1	0 1 0 1	5
	0 1 1 0	6
	1 0 1 0	10
2	0 0 1 0 1	5
	0 1 0 1 0	10
	1 0 0 1 0	18
3	1 1 0 1 0 0	52
	1 1 0 0 1 0	50
	1 0 1 0 1 0	42
	1 0 1 0 0 1	41
4	1 0 1 0 0 0 1	81
	1 0 0 1 1 0 1	77
	1 0 0 1 0 1 0	74
	0 1 1 1 0 0 1	57
	0 1 0 0 1 0 1	37
	0 1 0 0 0 1 0	34
5	0 0 1 0 1 0 0	20
	1 0 1 0 1 0 1	85
	0 0 1 1 0 0 1	25
	0 1 0 0 0 1 1	35
	1 1 0 1 0 1 0	106
	0 0 0 1 1 1 0	14
	1 1 1 0 0 0 1	113
	0 0 1 0 0 1 1	19
6	1 0 0 1 0 0 0	72
	1 0 1 0 1 0 0	84
	1 0 1 0 0 1 0	82
	1 0 1 0 0 0 1	81
	0 1 1 1 0 0 0	56
	0 1 0 0 1 0 0	36
	0 1 0 0 0 1 0	34
	0 1 0 0 0 0 1	33

2.0 Edge listings of graphs in Table IV - Chapter IV

Graphs 1 and 4 have been taken from [32] and [30] respectively. Graphs 5, 6 and 7 are complete graphs. Graph 9 corresponds to the AEP 14 busbar test-system. All other graphs have been randomly generated.

Graph 1

E	SB	EB
1	1	3
2	3	4
3	4	7
4	7	8
5	1	2
6	2	5
7	5	6
8	6	8
9	3	5
10	5	7

Graph 2

E	SB	EB
1	1	2
2	1	2
3	1	3
4	2	4
5	2	4
6	3	4
7	4	5
8	4	5
9	5	7
10	4	7
11	4	6
12	4	8
13	6	8
14	6	8
15	7	9
16	7	9
17	8	10
18	8	11
19	9	12
20	9	13
21	10	13
22	11	14
23	12	15
24	13	15
25	14	15
26	15	16
27	16	20
28	15	17
29	17	20
30	15	18
31	18	20
32	15	19
33	19	20

Graph 3

E	SB	EB
1	1	2
2	1	2
3	1	3
4	2	4
5	2	4
6	3	4
7	4	5
8	4	5
9	5	7
10	4	7
11	4	6
12	4	8
13	6	8
14	6	8
15	7	9
16	7	9
17	8	10
18	8	11
19	9	12
20	9	13
21	10	13
22	11	14
23	12	15
24	13	15
25	14	15

Graph 4

E	SB	EB
1	1	2
2	1	2
3	1	3
4	2	4
5	2	4
6	3	4
7	4	5
8	4	5
9	5	6
10	4	6
11	6	8
12	6	8
13	6	8
14	4	7
15	7	8
16	7	8

Graph 5

E	SB	EB
1	1	2
2	1	3
3	1	4
4	1	5
5	1	6
6	1	7
7	2	3
8	2	4
9	2	5
10	2	6
11	2	7
12	3	4
13	3	5
14	3	6
15	3	7
16	4	5
17	4	6
18	4	7
19	5	6
20	5	7
21	6	7

Graph 6

E	SB	EB
1	1	2
2	1	3
3	1	4
4	1	5
5	1	6
6	1	7
3	2	3
8	2	4
9	2	5
10	2	6
11	2	7
12	3	4
13	3	5
14	3	6
15	3	7
16	4	5
17	4	6
18	4	7
19	5	6
20	5	7
21	6	7
22	1	8
23	2	8
24	3	8
25	4	8
26	5	8
27	6	8
28	7	8

Graph 7

E	SB	EB
1	1	2
2	1	3
3	1	4
4	1	5
5	1	6
6	1	7
7	2	3
8	2	4
9	2	5
10	2	6
11	2	7
12	3	4
13	3	5
14	3	6
15	3	7
16	4	5
17	4	6
18	4	7
19	5	6
20	5	7
21	6	7
22	1	8
23	2	8
24	3	8
25	4	8
26	5	8
27	6	8
28	7	8
29	1	9
30	2	9
31	3	9
32	4	9
33	5	9
34	6	9
35	7	9
36	8	9

Graph 8

E	SB	EB
1	1	2
2	1	2
3	1	3
4	2	4
5	2	4
6	3	4
7	4	5
8	4	5
9	5	7
10	4	7
11	4	6
12	4	8
13	6	8
14	6	8
15	7	9
16	7	9
17	8	10
18	8	11
19	9	12
20	9	13
21	10	13
22	11	14
23	12	15
24	13	15
25	14	15
26	15	16
27	16	20
28	15	17
29	17	20
30	15	18
31	18	20
32	15	19
33	19	20
34	20	21
35	20	22
36	20	22
37	20	24
38	20	24
39	20	25
40	21	28
41	22	23
42	23	28
43	24	28
44	25	26
45	25	27
46	26	28
47	27	28

Graph 9

E	SB	EB
1	1	2
2	1	3
3	2	3
4	2	6
5	3	4
6	3	5
7	3	6
8	4	5
9	5	6
10	5	8
11	5	9
12	6	7
13	7	11
14	7	12
15	7	13
16	8	9
17	9	10
18	9	14
19	10	11
20	12	13
21	13	14

3.0 Edge listing of graphs in table VIII and XVI - Chapter IV

Graphs 10 and 11 have been randomly generated. Graph 12 corresponds to the West-Venezuelan 115 kV system.

Graph 10

E	SB	EB
1	1	10
2	1	5
3	10	7
4	10	2
5	5	2
6	2	7
7	-2	8
8	-2	8
9	-7	9
10	-7	3
11	-8	6
12	-9	4

Graph 11

E	SB	EB
1	1	2
2	1	3
3	2	4
4	2	10
5	3	10
6	10	4
7	10	9
8	10	9
9	-4	5
10	4	7
11	9	7
12	-5	6
13	-7	8
14	-8	11

Graph 12

E	SB	EB
1	1	2
2	1	4
3	1	5
4	1	7
5	1	13
6	2	3
7	3	4
8	4	5
9	5	6
10	6	7
11	2	8
12	8	9
13	-9	10
14	9	11
15	-11	12
16	11	13
17	-2	14
18	-14	15

APPENDIX 2

This appendix includes the detailed discussion and proof of each algorithm presented in chapter IV. First two general theorems are introduced whose proofs can be found in the reference quoted in each case. Then it follows the proofs specially developed by the author for the new minimal cut-set generation algorithms presented in sections 4.5, 4.6 and 4.7 of chapter IV.

A2.1 General theorems

Theorem 1[32] : The removal of the components in a minimal cut-set between nodes s and t separates the graph into exactly two connected subgraphs G_s and G_t where G_s contains node s and G_t contains node t.

Theorem 2[42] : The ring sum of any two cut-sets in a connected graph is either a third cut-set or an edge disjoint union of cut-sets.

A2.2 Discussion and proof of the algorithm for non-separable graphs of section 4.5

This section justifies the steps of the algorithm and proves that it generates the complete set of minimal cuts between the reference node and all other nodes of a nonseparable graph.

The procedure begins generating the basic minimal cut-set with the set of edges incident to the reference node. Each time a new node is scanned the algorithm automatically generates a minimal cut-set with the set of edges incident to the node being scanned. Steps 1 and 4 of the algorithm are justified by :

Theorem 3 : In a nonseparable graph G the set of edges incident on each node of G is a minimal cut-set.

Proof : To prove this lemma one should note that the removal of all edges incident to node i breaks all paths from the reference node r to node i. Then this removal separates the graph into at least two connected subgraphs, one contains node i and the other contains node r. Now we prove, by contradiction, that the removal of the edges incident to node i separates G into exactly two connected subgraphs G_1 and G_2 where G_1 contains only node i and G_2 contains all other nodes. Assume that there are more than two connected subgraphs. Then there must be a connected subgraphs G_3 that includes neither i nor r. As G is connected there must be at least one path in G between r and each node in G_3. Since the only edges removed are the ones incident to node i, if G_3 is disconnected from G_2 all paths from G_2 to G_3 must contain one of the removed edges. Then the original graph is separable because all paths from the reference node to any node in G_3 pass through node i i.e., graph G has node connectivity one. But this is contrary to the original assumption that G is nonseparable. Then the set of edges incident to node i separates the graph into exactly two connected subgraphs G_1 and G_2. Hence by Theorem 1 the Theorem is proved.

Corollary 1 : The minimal cut-set formed by the set of edges incident to node i is a minimal cut-set only for node i.

Corollary 2 : The minimal cut-set formed by the set of edges incident to the reference node is a minimal cut-set for all other nodes of the graph.

Corollary 1 and 2 justify the construction of the cut-node vectors generated by steps 4 and 1 of the algorithm respectively.

So far we have generated only a set of minimal cuts constituted by the proper minimal cuts of all nodes of the graph including the reference node. Step 5 suggests additional cut-sets are obtained from the ring sum operations between two or more of these cut-sets. Theorem 2 indicates that the ring sum of any two cut-sets in a graph is either a third cut-set or an edge-disjoint union of cut-sets. However we need to prove that the set of proper cuts is a fundamental set of minimal cuts from where all other cut-sets of a graph can be obtained.

<u>Theorem 4</u> : The set of minimal cuts corresponding to the set of $(n - 1)$ proper minimal cuts of a graph of n nodes forms a fundamental set for the generation of the complete set of minimal cuts between the reference node and all other nodes of the graph. This set includes the proper cut of the reference node.

Proof: By Theorem 1 the problem of generating the set of minimal cuts between the reference node r and all other nodes of a graph is the same problem that generates all the partitions of the set N of nodes into the subsets Y and $\overline{Y}$. The set Y must define a connected subgraph that includes node r. The set $\overline{Y}$ must define a connected subgraph that includes at least one of the other nodes of the graph. There are 2^n ways to partition N into two mutually exclusive subsets Y and $\overline{Y}$ in a connected graph. Since r is restricted to be a member of Y the number of partitions is reduced to 2^{n-1}. Every such partition of a complete graph is a minimal cut-set. Then we shall prove that the set of $(n - 1)$ proper cut-sets is a basis for the complete set of minimal cuts of a graph by showing that every partition of the set N in which node r and at least one of the other nodes are separated can be obtained from the set of subgraphs representing $(n - 1)$ proper cut-sets of the graph.

By Corollary 1 of Theorem 3 each proper cut-set j is a subgraph g_j which can be represented by an n-tuple(an ordered array of n elements)

$$X_j = (x_1, x_2, \ldots, x_n)$$

such that

$$x_i = 1 \quad \text{if node i is in } g_j \text{ and}$$
$$x_i = 0 \quad \text{if node i is not in } g_j$$

Then the set of subgraphs representing the proper cut-sets of a connected graph of n nodes are

$$X_1 = 1\,0\,0\,.\overset{n}{.}\,.\,0$$
$$X_2 = 0\,1\,0\,.\overset{n}{.}\,.\,0$$
$$\cdots\cdots\cdots\cdots$$
$$X_n = 0\,0\,0\,.\overset{n}{.}\,.\,1$$

Consider an arbitrary partition N_1 of the set N such that

$$N_1 = 0_1\, 1_2 \ldots 0_j\, 1_k\, 0_\ell\, 1_m\, 0_n$$

where nodes 2, k and m form the subset Y of N and all other nodes form subset $\overline{Y}$. The n-tuple representing N_1 can be seen to correspond to the ring sum of the n-tuples X_2, X_k and X_m. Then any of the 2^{n-1} partitions which define subsets Y and $\overline{Y}$ can be obtained by the ring sum of the n-tuples representing the proper cut-sets of the nodes in subset Y. Besides the n-tuple X_n define a partition equivalent to the one given by the ring sum of all other (n - 1) n-tuples therefore only (n - 1) proper cut-sets are necessary to generate the complete set of 2^{n-1} partitions. Hence the theorem is proved and justifies step 2 of the algorithm.

Obviously only for a complete graph is it necessary to enumerate the 2^{n-1} partitions because each one corresponds to a minimal cut-set. Then an efficient algorithm will not enumerate 2^{n-1} cases unless, of course, there are 2^{n-1} minimal cuts.

From the definitions of ring sum operation of two graphs and edge-disjoint events the ring sum of two cut-sets with no edges in common is always an edge-disjoint union of cut-sets. This property can be used to discard all the ring sum operations which positively produce an edge-disjoint union of cut sets. This reduction is achieved if only the ring sums between two cuts that have at least one edge in common are considered during the cut-set generation procedure. It has been provided in step 3 of the algorithm by the index $CI_k(i)$ associated with the ith cut-set when scanning node k.

However when scanning node k the ring sum of the proper cut-set with any of the cut-sets with non-zero index may also give, by Theorem 2, an edge-disjoint union of cut-sets. The algorithm must provide the steps to identify these edge-disjoint events.

<u>Lemma 1</u> : The possible edge-disjoint union of cut-sets generated by the algorithmic procedure when scanning node k is always contained in one of the minimal cut-sets generated during the same scan.

Proof: Note that the search of the nodes is made starting by the reference node r and once the basic minimal cut-set is generated we move to an adjacent node v. At v we form the proper cut-set of v and make the ring sums with all previous cut-sets which have at least one edge in common. Once we do all ring sums we move then to one of the unscanned nodes closest to r. This process is continued till (n - 1) nodes in the graph are considered. This method of searching is the one responsible that only

(n - 1) nodes need to be scanned because the proper cut-set of the nth node has been automatically generated from the ring sum of the (n - 1) proper cuts. At the same time this way of searching guarantees that each time a ring sum of the kth proper cut is made with a cut-set of non-zero index, the resulting set of edges (cut-set or edge-disjoint union of cut-sets) have at least one edge in common with the kth proper cut-set which is not contained in any of the minimal cut-sets generated when scanning the previous (k - 1) nodes. Then an edge-disjoint union of cut-sets generated from the ring sums of the kth proper cut contain one or more edges in common with this cut-set which are not present in any of the cut-sets generated before the scan of node k. Since by definition any edge-disjoint union of cut-sets can be decomposed into two mutually exclusive cut-sets, the one which includes the edges in common with the kth proper cut must correspond to one of the minimal cut-sets generated during the scan of node k. Hence the lemma.

This check is provided by step 5 of the algorithm and reduces the number of comparisons to a minimum, increasing the efficiency (by reducing the computation time) of the algorithm.

So far, only the incidence of the basic minimal cut-set and the proper cut-sets on the nodes of the graph have been defined. By definition the cut-node vector associated with each cut-set tells us which nodes are affected for that cut i.e., it contains the nodes that we have associated with subset $\overline{Y}$. The cut-node vector for all other cut-sets is generated as follows:

<u>Lemma 2</u> : A cut-set generated from the ring sum of the basic minimal cut-set and the jth proper cut-set is not a cut-set for node j.

Proof: From Corollary 2 of Theorem 3 the subsets Y and $\overline{Y}$ from the basic cut-set are $Y_1 = 1$ and $\overline{Y}$ contain all other nodes. Now, the basic

cut-set and the jth proper cut-set are combined (ring sum) only if they have at least one edge in common. Since the ring sum operation discard all the edges in common the resulting set of edges cannot be anymore a minimal cut-set for node j because the edge(s) discarded provide a path(s) from the reference node to node j. For the new set of edges $Y = 1,j$ and $\overline{Y}$ all other nodes.

Lemma 3 : A cut-set generated from the ring sum of two proper cut-sets is a cut-set for both nodes.

Proof: From Corollary 1 of Theorem 3 the jth and kth proper cut-sets are cuts only for nodes j and k respectively i.e., $\overline{Y}_j = j$ and $\overline{Y}_k = k$. In the procedure they are combined only if they have at least one edge in common. Since each edge has only 2 terminals, any edge in common between the two cut-sets must have j and k as terminal nodes. Then the set of edges from their ring sum still breaks all paths from the reference node to nodes j and k. For the new set of edges $\overline{Y} = j$ and k and Y contain all other nodes.

Therefore the algorithmic procedure presented in this section does represent an efficient and systematic way to generate the complete set of minimal cuts between the reference node and all other nodes of a nonseparable graph.

A2.3 Discussion and proof of the algorithm for separable graphs of Section 4.6

The algorithm for separable graphs with some of its subgraphs with tree configuration results from slight modifications introduced to the algorithm presented for nonseparable graphs. However some of the Theorems and Lemmas presented in section (A2.2) are not directly applicable

for separable graphs. This section will justify the steps of the algorithm and prove that it does generate the complete set of minimal cuts between the reference node and all other nodes of separable graphs as defined in section (4.6.1).

As before the process starts generating the basic minimal cut-set with the set of edges incident to the reference node. Since the systems analysed by the algorithm are considered to be connected, the reference node as defined in this thesis cannot be an articulation point. Then the removal of the edges incident on the reference node r must separate the graph into two connected subgraphs, one contains r and the second all other nodes of the system. Otherwise node r is an articulation point. Hence by Theorem 1 the set of edges incident to node r is a minimal cut-set and affects all other nodes of the graph. Hence step 1 of the algorithm is justified. All other proper cut-sets enumerated in step4 of the process are justified below.

<u>Theorem 5</u> : In a separable graph the complete set of edges on an articulation point are not a minimal cut-set.

Proof: To prove the Theorem let A be the set of all edges incident on the articulation point v of graph G. If A is a minimal cut-set then by Theorem 1 of Appendix 2 the removal of A from G must separate the graph into exactly two connected subgraphs G_1 and G_2. Since A contains all edges incident on v, if subgraph G_1 contains only node v then G_2 must contain all other nodes. Now, by definition a connected graph is said to be separable if there exists a subgraph g in G such that $\overline{g}$ (the complement of g in G) and g have only one node in common. Then all paths between any node in g and a node in $\overline{g}$ must pass through

this common node. Such common node is what is called an articulation point. Since G_1 contains only node v, if v is an articulation point the removal of A must define a third subgraph G_3 such that all paths from G_2 to G_3 pass through v. Then by Theorem 1, the set A is not a minimal cut and the Theorem is proved.

From Theorem 5 it can be concluded that only a subset of the edges incident to an articulation point will be a minimal cut-set between that point and the reference node.

Let G_1 and G_2 the two subgraphs of G defined by an articulation point v. If subgraph G_1 contains the reference node then we have assumed that G_2 has a tree configuration where the set of edges with v as terminal will form the root of the tree. Then all cut-sets between the tree nodes (internals or pendants) and the reference node will also be cut-sets between the articulation point and the tree nodes. Since there is one and only one path between every pair of nodes in a tree, there is only one path (see Theorem 6) between the articulation point and each one of the other nodes of the tree. This one-to-one correspondence suggests to solve the cut-set enumeration problem by representing each subgraph G_2 by a directed subgraph whose edges are oriented moving away from the articulation point. With this convention the subset of edges in the tree connected to the articulation point will be oriented incident out of the articulation point. Then only the set of undirected edges incident on the articulation point will form a cut-set for that point. Similarly only the set of edges incident into a tree node will separate the node from the articulation point dividing the subgraph into two connected subgraphs. Then this set of edges is a minimal cut-set for the node tree. This explains the generation of the proper cut-sets by step 4 of the algorithm. Moreover all cut-sets

of this tree will be disjoint events respective of any other cut-set of the graph, then there is no need to make the ring sums with the previous cut-sets as indicated in step 4. This disjoint property also justifies step 2. However if one restricts that the last unscanned node will never be a tree node, only NN - 1 nodes need to be scanned by the procedure.

The subgraph which includes all nodes between the reference point and each articulation point is a nonseparable graph therefore for all these nodes the algorithms steps were justified in section (A2.2). The only difference is when constructing the cut-node vectors of the cut-sets between the reference node r and an articulation point w. In this case, as mentioned before, each cut-set will separate the graph into two connected subgraphs. One subgraph will include the reference node and the other will always include w and all nodes whose paths from r pass through w. Then the cut-node vector generation follows from Lemmas 2 and 3 if we note that the presence or absence of a path between r and w implies also the connection or disconnection respectively of all tree nodes connected to the articulation point.

<u>Theorem 6</u> : In a tree every node with degree greater than one is an articulation point.

Proof: A node with a degree greater than one will have at least two incident edges. Then these two incident edges will have one common node and two non-adjacent nodes. A tree by definition is a connected graph without circuits. The union of two different paths between two nodes will always form a circuit. Then there is one and only one path between every pair of nodes in a tree. Therefore the removal of a node tree with degree greater than one will disconnect every pair of non-

adjacent nodes of the edges incident to that node and therefore will disconnect the tree. Hence the theorem is proved.

By Theorem 6 the construction of the cut-node incidence vector for each internal node of a tree is justified as mentioned for the case of an articulation point. Clearly the proper cut-set of a pendant node will affect only that node.

The fundamental set of minimal cuts used by the algorithm is the group of proper cut-sets of the nodes of the graph. Then by Theorem 4 the set of (n - 1) proper cut-sets in an n-nodes graph does generate the complete set of minimal cuts. As mentioned before this fundamental set must include all proper cut-sets of any tree node in the graph.

A2.4 Discussion and proof of the algorithm for directed acyclic graphs of section 4.7

The method is based on a combinational approach to the problem of enumerating minimal cut-sets rather than a purely mathematical one. The requirement for a combination of edges to be identified as an s-t directed cut-set is that this set of edges must break all directed paths between s and t. However the procedure instead of enumerating all directed paths and after solving the problem by a combinational approach[30], it searches directly for all the edges combinations that break the s-t paths as it moves from node to node.

It is evident that if we must satisfy that each combination of edges break all s-t paths, one needs to start the combination procedure with one set that holds that property. Since all directed paths between an origin node s and a terminal node t must include at least one edge incident out of s and one edge incident into t, the set of edges incident out of s is an s-t minimal cut-set. Step 1 of the algorithm constructs this cut-set and uses it as the basis to generate all other

combinations as follows. Once this cut-set is formed the procedure moves to an unscanned node w adjacent to the origin. This is achieved by step 3 that uses an index cut to search for adjacent nodes. Once a node w we scan all edges incident on w. Each cut-set already in the list will form a new cut-set replacing the edges incident into w that are contained in the cut by the set of edges going out of w. To be sure that all nodes nearest than w to the origin have been scanned all the edges incident into w must be already scanned by the algorithm and therefore contained in at least one of the cut-sets previously generated. This also ensures that all possible directed paths passing through w have been implicitly constructed by the algorithm so any combination with the edges leaving w will break all s-t paths. Once all combinations have been made with the edges incident out of w, we move to an adjacent node v. At v we scan again all edges connected to v and make new combinations with the edges going out of v. This process is continued until all edges incident out of any node in the graph are scanned. Since node t has only edges leading into t the procedure stops when (n - 1) nodes of a graph of n nodes have been searched (step 2).

Each new combination of edges is a cut-set because starting from a cut-set all that it does is to replace edges for those that generate the directed paths that pass through the node under scan. Unfortunately no all cuts generated by the procedure need to be minimal therefore if a new combination of edges is contained or contains one of the previous cut-sets, only the combination with minimum number of elements is kept as a cut-set. Note that each new combination contains all the edges incident out of the node under scan and none of these edges can be present in any one of the cuts present in the list before the actual scan is started. Otherwise it is incident out from two different nodes.

This reduces the comparisons at minimum because each new combination of edges needs to be compared only with the cut-sets generated when scanning the same node (step 4).

REFERENCES

1. HECHT, H. and GRONBERG, G. "The use of on-line minicomputers in Swedish municipal power distribution systems", Symposium on implementation of real-time control by digital computer, paper M, Imperial College, London, September 1973.

2. WIMMEL, K.H. and ZOLLENKOPH, K. "Experience in the use of on-line digital computer for switching a 380/110 kV system", Symposium on implementation of real-time control by digital computer, paper N, Imperial College, London, September 1973.

3. COUCH, G.H. and MORRISON, I.F. "Data validation and topology determination for power system monitoring and control", IEEE Conference paper No. C74 361-2, presented at the Summer Power Meeting, July 1974.

4. BREWER, G.L. and TAM, C.C. "The computational aspects of the CEGB new national control centre", 3rd PSCC Proceedings, paper STY-5, June 1969.

5. PULSFORD, H.E., LAST, F.H., and LOWE, B. "On-line facilities in the national control centre of the Central Electricity Generating Board", CIGRE paper No. 32-12, 1972.

6. MURRAY, B.E. and DROMEY, G. "Practical design considerations affecting the use of digital computers in high voltage substations", 4th IFAC/IFIP International conference on digital computer applications to process control, Part II pp. 292-303, March 1974.

7. HORNE, E. and CORY, B.J. "Digital processors for substation switching and control", IEE Conference on developments in power system protection, Conference publication No. 125, March 1975.

8. EDGLEY, R.K. "Back-up protection and overall protection of power system networks" Electra-CIGRE, No. 34, May 1974.

9. WARRINGTON, A.R. Van C. "Protective relays: Their theory and practice", Volume 1, Chapman and Hall Ltd., London 1968.

10. HARDER, E.L. and CUNNINGHAM, J.C. "Relay and circuit breaker application". Electrical transmission and distribution reference book, Chapter 11, Westinghouse Electric Corporation, 1950.

11. SEYMOUR, C.R. "Statistics of CEGB protection performance 1968-73", IEE Conference on developments in power system protection", Conference publication No. 125, March 1975.

12. GONZALEZ, G.E. "Relaying data validation". Digital protection of power systems - Post experience course, Imperial College, London, March 1975.

13. MINIESY, S.M. "Integrated digital protection and control of high voltage substations", Power system report No. 94, Imperial College, London September 1973.

14. HORNE, E. "Interface reference manual", Unpublished report, Imperial College, London, November 1974.

15. MOORE, E.F. and SHANNON, C.E. "Reliable circuits using less reliable relays", Journal of the Franklin Institute, Vol. 262, 1956, pp. 191-208 and pp. 281-297.

16. SINGH, C. and BILLINTON, R. "A new method to determine the failure frequency of a complex system", IEEE Trans. on Reliability, Vol. R-23 No. 4, October 1974.

17. BAZOVSKY, I. "Reliability theory and practice", Prentice-Hall Space Technology series, New Jersey, 1961.

18. SHOOMAN, M.L. "Probabilistic reliability : An engineering approach", McGraw-Hill, New York, 1968.

19. DUBES, R. "Two algorithms for computing reliability" IEEE Trans. on Reliability, December 1963.

20. WING, O. and DEMETRIOUS, P. "Analysis of probabilistic networks", IEEE Trans. on Commun. Technol., Vol. COM-12, September 1964.

21. MISRA, K.B. "An algorithm for the reliability evaluation of redundant networks", IEEE Trans. on Reliability, Vol. R-19, No. 4, November 1970.

22. MOSKOWITZ, F. "The analysis of redundancy networks", AIEE Trans. Commun. Electron. Vol. 77, November 1958.

23. HENLEY, F.J. and WILLIAMS, R.A. "Graph theory in modern engineering", Academic Press, New York, 1973.

24. LEE, C.Y. "Analysis of switching networks", Bell Syst. Techn. J., November 1955.

25. BROWN, D.B. "A computerized algorithm for determining the reliability of redundant configurations" IEEE Trans. on Reliability, Vol. R-20, March 1971.

26. FRATTA, L. and MONTANARI, U.G. "A Boolean algebra method for computing the terminal reliability in a communication network" IEEE Trans. on Circuit theory, Vol. CT-20, May 1973.

27. LIN, P.M., LEON, B.J. and HUANG, T.C. "A new algorithm for symbolic system reliability analysis" IEEE Trans. on Reliability, Vol. R-25, No. 1, April 1976.

28. PIERCE, A.R. "Bibliography on algorithms for shortest path, shortest spanning tree, and related routing problems 1956-1974", Networks, Vol. 5, No. 2, April 1975.

29. OKADA, S. "Algebraic determination of loop and paths in graphs", the fourth international conference on operations research - Pre-prints of proceedings, International Federation of Operational Research Society, Boston 1966.

30. NELSON, A.C. Jr., BATTS, J.R. and BEADLES, R.L. "A computer program for approximating system reliability" IEEE Trans. on Reliability, Vol. R-19, No. 2, May 1970.

31. BATTS, J.R. "Computer program for approximating system reliability - Part II", IEEE Trans. on Reliability, Vol. R-20, No. 2, May 1971.

32. JENSEN, P.A. and BELLMORE, M. "An algorithm to determine the reliability of a complex system", IEEE Trans on Reliability, Vol. R-18, No. 4, November 1969.

33. BELLMORE, M. and JENSEN, P.A. "An implicit enumeration scheme for proper cut generation", Technometrics, Vol. 12, No. 4, November 1970.

34. HANSLER, E., McAULIFFE, G.K. and WILKOV, R.S. "Exact calculation of computer network reliability" Networks, Vol. 4, 1974.

35. BARLOW, R.E. and PROSCHAN, F. "Mathematical theory of reliability", John Wiley and Sons Inc., New York, 1965.

36. BARLOW, R.E. and PROSCHAN, F. "Statistical theory of reliability and life testing: Probability models", Holt, Rinehart and Winston Inc., New York, 1975.

37. FRANK, H. and FRISCH, I.T. "Communication, transmission and transportation networks", Addison-Wesley, Reading, Mass., 1971.

38. WILLIAMS, G. "The design of survivable communication networks", IEEE Trans. Commun. Syst., Vol. CS-11, June 1963.

39. MESSINGER, M. and SHOOMAN, M.L. "Reliability approximations for complex structures", Proc. 1967 Ann. Symp. Reliability IEEE, New York.

40. ESARY, J.D. and PROSCHAN, F. "Coherent structures of non-identical components", Technometrics, Vol. 5, No. 2, May, 1963.

41. FELLER, W. "An introduction to probability theory and its applications", Vol. I, 3rd Ed., John Wiley and Sons, New York, 1968.

42. DEO, N. "Graph theory with applications to engineering and computer science", Prentice-Hall Inc., Englewood Cliffs, N.J., 1974.

43. GONZALEZ-URDANETA, G.E. "Reliability of redundant configurations", Power System Report No. 97, Imperial College, London, May 1975.

44. CHRISTIAANSE, W.R. "A technique for the analysis of repairable redundant systems", IEEE Trans. on Reliability, Vol. R-19, No. 2, May 1970.

45. ELMAGHRABY, S.E. "Activity networks and their generalizations", Lecture notes in operations research and mathematical systems, No. 29, Springer-Verlag, 1970.

46. EPSTEIN, B. and SOBEL, M. "Life testing", American Statistical Association Journal, pp. 486-501, September 1953.

47. BORG, P.W. ter "Field reporting system for reliability analysis on telecommunication equipment", Proceedings of the NATO conference on reliability testing and reliability evaluation, paper III-A, September 1972.

48. DRENICK, R.F. "The failure law of complex equipment", J. Soc. Ind. and Appl. Math., Vol. 8, pp. 680-690, 1960.

49. EL-SAYYAD, G.M. "Estimation of the parameter of an exponential distribution", Journal of the Royal Statistical Society, Series B, Vol. 29, pp. 525-532, 1967.

50. GNEDENKO, B.V., BELYAYEV, Yu.K. and SOLOVYEV, A.D. "Mathematical methods of reliability theory", Academic Press, 1969.

51. NAGELE, G.G. and SELLINSCHEGG, D.W. "Test cost minimization for reliability assessment", Proc. 1972 Annual Symposium on Reliability.

52. LLOYD, D.K. and LIPOW, M. "Reliability, management, methods and mathematics", Englewood Cliffs, N.J., Prentice Hall, 1962.

53. SATHE, Y.S. and VARDE, S.D. "On minimum variance unbiased estimation of reliability", The Annals of Mathematical Statistics, Vol. 4D, pp. 710-714, April 1969.

54. PUGH, E.L. "The best estimate of reliability in the exponential case", Operation Research, Vol. 11, pp. 57-61, 1963.

55. KOPAL, Z. "Numerical Analysis, Chapman and Hall, Ltd., 1961.

56. MANN, N.R. "A survey and comparison of methods for determining confidence bounds on system reliability from subsystem data", Proceedings of the NATO conference on reliability testing and reliability evaluation, paper I-B, September 1972.

57. ROSENBLATT, J.R. "Confidence limits for the reliability of complex systems", Statistical Theory of Reliability (Ed. M. Zelen), The University of Wisconsin Press, Madison, Wisconsin, 1963.

58. MURCHLAND, J.D. and WEBER, G.G. "A moment method for the calculation of a confidence interval for the failure probability of a system", Proc. 1972 Annual Reliability and Maintainability Symposium.

59. BURNETT, T.L. and WALES, B.A. "System reliability confidence limits", Proc. 7th National Symp. on Reliability and Quality Control, pp. 118-128, 1961.

60. LEVY, L.L. and MOORE, A.H. "A Monte Carlo technique for obtaining system reliability confidence limits from component test data", IEEE Trans. on Reliability, Vol. R-16, No. 2, pp. 69-72, September 1967.

61. KAMAT, S.J. and RILEY, M.W. "Determination of reliability using event-based Monte Carlo simulation", Trans. on Reliability, Vol. R-24, No. 1, pp. 73-75, April 1975.

62. MEYER, P.L. "Introductory probability and statistical applications", Addison-Wesley Publishing Co. Inc., 1965.

63. SINGH, C. "Reliability modelling and evaluation in electric power systems", Ph.D. thesis, University of Saskatchewan, Saskatoon, Canada, August 1972.

64. LARSON, H.J. "Introduction to probability theory and statistical inference", Wiley series in Probability and Mathematical Statistics, 1969.

65. GONZALEZ-URDANETA, G.E. and CORY, B.J. "A minimal cut-set algorithm for the reliability evaluation of transmission and distribution systems". 3rd Annual Symposium on Uncertainty in Risk and Reliability, University of Bradford, England, 1976.

66. SMIRNOV, E.P. "An approach to the evaluation of the reliability of relay protection gear", Electric Technology USSR, Vol. 3, No. 9-10, 1965.

67. ZUL', N.M. and KULIEV, F.A. "On the dependability of relay control gear in electrical networks", Electric Technology USSR, Vol. 3, No. 9-10, 1965.

68. FABRIKANT, V.L. "Applying reliability theory to appraisal of the performance of relay protection gear", Electric Technology USSR, Vol. 3, No. 9-10, 1965.

69. GRIMES, J.D. "On determining the reliability of protective relay systems", IEEE Trans. on Reliability, Vol. R-19, No. 3, August 1970.

70. NAYLOR, J.H. "A mathematical model for the prediction of protection reliability" IEE Conference - Developments in Power System Protection, London, March 1975.

71. MURRAY, B. and HUGHES, G. "An approach to on-line substation control", Fourth Power Systems Computation Conference, Grenoble, September 1972, Paper 3.2/1.

72. ARNOTT, I.A., PITMAN, J.W., BATES, I.P., MILLER, I.H. and SCOTT, A.J.M. "Development of system control facilities for the Victorian transmission network", CIGRE SC31-32 combined meeting, Australia, March 1973, paper D3.

73. CONNELLY, K., SCOTT, A.J.M., CAMPAIN, A.J. and MORRISON, I.F. "Integrated protection and control in substations", CIGRE Paper 34-07, August 1974.

74. DROMEY, G. and RASMUSSEN, J.P. "Integrated control concepts in power systems", CIGRE Paper 34-08, August 1974.

75. LUCKETT, R.G., MUNDAY, P.J. and MURRAY, B.E. "A substation based computer for control and protection", IEE Conference-Developments in power system protection, London March 1975.

76. CORY, B.J., DROMEY, G. and MURRAY, B.E. "Digital systems for protection", CIGRE Paper No. 34-08, August 1976.

77. GILCREST, G.B., ROCKEFELLER, G.D. and UDREN, E,A. "High speed distance relaying using a digital computer - Part I : System description" IEEE Trans. on Power App. and Systems, Vol. PAS-91, No. 3, 1972.

78. GILCREST, G.B., ROCKEFELLER, G.D. and UDREN, E.A. "High speed distance relaying using a digital computer - Part II : Test results" IEEE Trans. on Power App. and Systems, Vol. PAS-91, No. 3, 1972.

79. BILLINTON, R. and GROVER, M.S. "Reliability evaluation in distribution and transmission systems", Proc. IEE, Vol. 122, No. 5, May 1975.